



**HAL**  
open science

# Advanced data validation methods for wastewater sensors using Artificial Intelligence

Imane Zidaoui

► **To cite this version:**

Imane Zidaoui. Advanced data validation methods for wastewater sensors using Artificial Intelligence. Fluids mechanics [physics.class-ph]. Université de Strasbourg, 2024. English. NNT: 2024STRAD006 . tel-04639689

**HAL Id: tel-04639689**

**<https://theses.hal.science/tel-04639689v1>**

Submitted on 9 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*ÉCOLE DOCTORALE MATHÉMATIQUES, SCIENCES DE  
L'INFORMATION ET DE L'INGÉNIEUR*

Laboratoire des sciences de l'ingénieur, de l'informatique et de  
l'imagerie

(ICube) – UMR 7357

**THÈSE** présentée par :

**Imane ZIDAOU**

soutenue le : 06 mai 2024

pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité : Hydraulique urbaine

**Advanced Data Validation Methods for  
Wastewater Sensors Using Artificial  
Intelligence**

**THÈSE dirigée par:**

**Pr. VAZQUEZ José**

Professeur, ENGEES, Université de Strasbourg

**RAPPORTEURS:**

**M. RODRIGUEZ Fabrice**

**Pr. COUTURIER Raphaël**

Chercheur IDTPE, Université Gustave Eiffel

Professeur, Université de Franche-Comté

**AUTRES MEMBRES DU JURY:**

**Pr. FORESTIER Germain**

**Pr. WEMMERT Cédric**

**M. JOANNIS Claude**

**Mme. ISEL Sandra**

**M. WERTEL Jonathan**

Professeur, Université de Haute-Alsace

Professeur, Université de Strasbourg

Consultant indépendant, CJ Conseil

Chef d'agence, 3D EAU Strasbourg

Directeur général, 3D EAU



# Acknowledgments

“If it doesn’t challenge you, it won’t change you” – Fred DeVito

These words perfectly sum up the essence of my PhD journey. This adventure has not been without its difficulties, but each challenge I have encountered has enabled me to mature, both academically and personally. Before sharing with you the fruit of these three years of research work devoted to the use of artificial intelligence in the service of data validation, I would like to express my gratitude to all those who have contributed, directly or indirectly, to the achievement of this thesis.

First of all, I would like to express my sincere thanks to **3D EAU** for funding this thesis, and in particular to **Jonathan Wertel**, CEO of 3D EAU, for his confidence in allowing me to carry out this innovative research within his teams.

I would also like to express my gratitude to all the members of the jury who agreed to assess my research work. My warmest thanks go to Professors **Jean-Luc Bertrand Krajewski** and **Raphaël Couturier** for taking the time to read and review this paper. I am also thankful to Professor **Germain Forestier** for agreeing to evaluate this work.

Without the invaluable help of **José Vazquez**, this thesis would never have seen the light. He patiently guided me through my first steps in urban hydraulics as a first-year student at ENGEES and introduced me to the world of research. His support and confidence greatly influenced my choices of specialty and thesis. I am infinitely grateful to him for his guidance throughout these years.

I would also like to express my gratitude to **Matthieu Dufresne** and **Sandra Isel** for their warm welcome to the Strasbourg team. Their kindness, attentive supervision and sound advice have greatly contributed to the quality of this research work. My special thanks go to **Claude Joannis** for his support in validating the data and his enlightened advice, which enriched my thinking and sharpened my critical mind. His high-quality feedback contributed greatly to the completion of this manuscript. My thanks also go to **Cédric Wemmert** for his guidance in learning artificial intelligence and programming. His availability and constant support have been invaluable throughout this work. I'm grateful to have had the opportunity to work with such a remarkable team, and I won't forget the constructive and stimulating exchanges we had during our monthly meetings.

Special thanks to **Saint Malo Agglomeration** and **SPGE of Wallonia** for their invaluable contribution in making available measurement data from their wastewater systems.

A PhD also relies heavily on moral support. So, I'd like to express my gratitude to those who have been there for me every day. **Gabriel Guibu Pereira**, thank you for our discussions over a cup of coffee. I look forward to tackling many R&D challenges together at 3D EAU. A big thank you also to **Thibaud Maire** for his constant support, good humor and attentive listening. Your new passion for AI is stimulating, and I can't wait to develop lots of things together. Finally, I can't forget **Angel Manjarres**, with whom I shared part of this adventure, our theses being practically synchronized. Thank you for our long conversations on the docks and for your unfailing support along the way.

A sincere thank you also to **my friends**, both in Morocco and in France. Your support has been of paramount importance to me. I know I'm probably forgetting many people, but I want to assure you of my gratitude for your presence and for your contribution to the development of this work.

أخيراً و ليس أخراً، أنا مدينة بالامتنان لعائلتي التي كانت مرساتي في هذه العاصفة. **نادية وعبد الخالق**، دعمكما غير المشروط وتضحياتكما وإيمانكما الثابت بي طوال الوقت منحنتي القوة للمثابرة. إلى إخوتي، **مها وبدر الدين و حسنة** أشكركم على ووقوفكم الدائم إلى جانبي. إلى توأم روحي، **أيوب**، أشكرك على كونك مشجعي وعلى تنكيري دائماً بأنني لست وحدي في هذه الرحلة. وأخيراً أود أن أهدي هذا العمل **لذكرى أجدادي**. لن أنسى أبداً جدي الذي ظل يناديني بالدكتورة إيمان، وكنت أشرح له دائماً أنني أدرس الهندسة وليس الطب. وها أنا اليوم أحمل هذا اللقب، وكأن الأمر كان مقدراً. محبتكم لم تفارقني أبداً، وأعلم أنكم ترعونني أينما كنتم

# *Abstract*

Data reliability in wastewater system management is crucial because of the direct implications on operations. However, current approaches to data validation are often costly and/or lack objectivity. This thesis explores advances in artificial intelligence to establish robust validation. The establishment of a human validation pool shows that the average F1 score between experts remains at 0.81, highlighting the inevitable human bias. The models tested, namely Matrix Profile, ResNet and the Autoencoder, show promising results, with an F1 score of 0.96 for the latter, indicating an ability to effectively detect abnormal sequences in the time series. Matrix Profile excels in non-supervised, ideal for low failure sites, while ResNet is useful in more problematic contexts, which can justify a manual validation phase a priori. These findings open up prospects for improved management of wastewater networks, based on data made more reliable thanks to AI.

**Keywords:** Wastewater networks, Artificial intelligence, Sensors, Time series, Validation, Anomalies, Matrix Profile, ResNet, Autoencoder

# Résumé

La fiabilité des données dans la gestion des réseaux d'eaux usées est cruciale en raison des implications directes sur les opérations. Cependant, les approches actuelles de validation des données sont souvent coûteuses et/ou manquent d'objectivité. Cette thèse explore les avancées en intelligence artificielle pour instaurer une validation robuste. La mise en place d'un pôle de validation humaine montre que le F1 score moyen entre experts reste à 0.81, soulignant l'inévitable biais humain. Les modèles testés, à savoir Matrix Profile, ResNet et l'Autoencodeur, présentent des résultats prometteurs, avec un F1 score de 0.96 pour ce dernier, indiquant une capacité à détecter efficacement les séquences anormales dans les séries temporelles. Matrix Profile excelle en non-supervisé, idéal pour des sites à faible défaillance, tandis que ResNet se montre utile dans des contextes plus problématiques, pouvant justifier une phase de validation manuelle à priori. Ces conclusions ouvrent des perspectives pour une gestion améliorée des réseaux d'eaux usées, basée sur des données fiabilisées grâce à l'IA.

**Mots-clés :** Assainissement, Intelligence artificielle, Capteurs, Séries temporelles, Validation, Anomalies, Matrix Profile, ResNet, Autoencodeur

# Table of Contents

<b>Acknowledgments</b> .....	<b>iii</b>
<b>Abstract</b> .....	<b>v</b>
<b>Résumé</b> .....	<b>vi</b>
<b>Table of Contents</b> .....	<b>vii</b>
<b>Table of Figures</b> .....	<b>xiii</b>
<b>Table of Tables</b> .....	<b>xxii</b>
<b>Table of Equations</b> .....	<b>xxiv</b>
<b>Acronyms and abbreviations</b> .....	<b>xxv</b>
<b>General Introduction</b> .....	<b>1</b>
Context and issues .....	1
Objectives and thesis outline .....	9
Context of the thesis.....	11
Scientific contributions .....	12
<b>Introduction of Part I : Literature Review</b> .....	<b>14</b>
<b>Chapter 1. Wastewater data in urban networks</b> .....	<b>15</b>
1.1 Measurement network monitoring .....	16
1.1.1 Data acquisition.....	17
1.1.2 Data management (supervision).....	18
1.2 Special features of data in wastewater networks.....	19
1.3 Defects in wastewater systems .....	21
1.3.1. Defining invalid data .....	21
1.3.2. Categorizing anomalies .....	23
1.4 Focus on turbidity data.....	24
1.5 Synthesis of Chapter 1 .....	28
<b>Chapter 2. Exploring data validation pathways: State-of-art approaches</b> .....	<b>29</b>
2.1 Data quality checkup: Pre-validation .....	29
2.2 Validation phase .....	31
2.2.1. Manual Approach .....	32
2.2.2. Statistical Tools .....	34
2.2.3. Hydraulic modelling .....	37
2.3. Synthesis of Chapter 2.....	41
<b>Chapter 3. Artificial Intelligence - Enhanced Data Validation Framework</b> .....	<b>43</b>
3.1 AI Vocabulary & History: A Primer.....	43
3.1.1. A timeline of artificial intelligence advancements.....	43



3.1.2.	Learning approaches.....	44
3.1.3.	Traditional AI, ML, DL.....	46
3.1.4.	From perceptron to neural networks.....	47
3.2	Key Considerations Before Implementing AI: Questions to Ask.....	52
3.2.1.	What do we want to do?.....	53
3.2.2.	What do we have as input?.....	54
3.2.3.	What are we looking to produce as output?.....	55
3.2.4.	Major Problem Complexities.....	56
3.3	Anomaly detection using AI in urban hydrology.....	57
3.3.1.	Anomaly Detection through Classification Approaches.....	58
3.3.2.	Exploring Anomaly Detection in Unsupervised Mode.....	63
3.3.3.	Detecting Anomalies Using Prediction Models.....	68
3.4	AI-Powered Anomaly Detection: A Broader Outlook.....	71
3.5	Synthesis of Chapter 3.....	74
<b>Synthesis of Part I : Literature Review.....</b>		<b>75</b>
<b>Introduction of Part II : Material and Methods.....</b>		<b>78</b>
<b>Chapter 4. AI's Backbone: Introducing our model evaluation database.....</b>		<b>79</b>
4.1.	Introduction and background.....	79
4.2.	Sensors' network and data availability.....	80
4.2.1.	Data collection process.....	80
4.2.2.	An overview of sensors in place.....	81
4.3.	Data Exploration: Analyzing the Database.....	82
4.3.1.	Data acquisition.....	82
4.3.2.	Understanding data statistics.....	84
4.3.3.	Understanding data dynamics.....	87
4.4.	Data validation.....	88
4.4.1.	Manual data validation processus.....	88
4.4.2.	Anomalies qualification.....	92
4.4.3.	Mitigating subjectivity by organizing a validation pool.....	94
4.5.	Synthesis of Chapter 4.....	95
<b>Chapter 5. Benchmarking Models for Data Validation and Anomaly Detection.....</b>		<b>97</b>
5.1	Does our data justify the use of AI approaches?.....	97
5.2	Benchmark of models and tests.....	98
5.3	Matrix Profile.....	102
5.3.1	Introduction and Background.....	102
5.3.2	Definitions and Notation.....	104
5.3.3	Principle and algorithms.....	108

5.3.4	Anomaly detection using Matrix Profile.....	109
5.3.5	Conclusion .....	112
5.4	ResNet.....	114
5.4.1	Introduction and background .....	114
5.4.2	Definitions and notation .....	115
5.4.3	Model architecture.....	119
5.4.4	Anomaly detection using ResNet.....	120
5.4.5	Conclusion .....	127
5.5	Autoencoder .....	129
5.5.1	Introduction and background .....	129
5.5.2	Definitions and notation .....	130
5.5.3	Model architecture.....	131
5.5.4	Anomaly detection using AE.....	133
5.5.5	Conclusion .....	139
5.6	Synthesis of Chapter 5.....	141
<b>Chapter 6. Beyond data and models: Performance Metrics and Hardware-Software Configuration.....</b>		<b>143</b>
6.1	Model's performance metrics .....	143
6.1.1	Confusion Matrix .....	143
6.1.2	Matthew's Correlation Coefficient.....	145
6.1.3	Characteristic curves.....	148
6.2	Annotator agreement metrics.....	150
6.2.1	Cohen's kappa coefficient.....	151
6.2.2	Pairwise F1 score .....	152
6.2.3	Smyth's coefficient.....	153
6.2.4	Dendrogram.....	154
6.3	Behind the Scenes of AI Models: Hardware & Software.....	155
6.3.1	Programming language.....	155
6.3.2	Environment set-up.....	155
6.3.3	Hardware .....	156
6.4	Synthesis of Chapter 6.....	157
<b>Synthesis of Part II : Material and Methods .....</b>		<b>159</b>
<b>Introduction of Part III : Results and Discussion .....</b>		<b>162</b>
<b>Chapter 7. Annotator Agreement.....</b>		<b>163</b>
7.1.	Identifying "Outlier" Experts with Pairwise F1 Score.....	164
7.2.	Clustering Experts using Dendrogram.....	167
7.3.	Assessing Beyond Chance Agreement with Cohen's Kappa.....	169

7.4.	Smyth's Coefficient Analysis for Evaluating global annotator agreement.....	174
7.5.	Synthesis of Chapter 7.....	178
<b>Chapter 8.</b>	<b>Matrix Profile Evaluation.....</b>	<b>181</b>
8.1.	Sensitivity to input data.....	181
8.1.1.	Preprocessing.....	181
8.1.2.	Input data.....	186
8.2.	Hyperparameters tuning.....	187
8.3.	How can we improve the results ?.....	196
8.3.1.	Combining the results using raw data.....	196
8.3.2.	Ensemble model.....	197
8.3.3.	Pre-validation.....	200
8.4.	Generalization to other sites.....	201
8.5.	Multivariable anomaly detection.....	204
8.5.1.	Bivariate matrix profile.....	205
8.5.2.	Multivariate matrix profile.....	209
8.5.3.	How can we improve the results ?.....	210
8.6.	Synthesis of Chapter 8.....	214
<b>Chapter 9.</b>	<b>ResNet Evaluation.....</b>	<b>217</b>
9.1.	Sensitivity to input data.....	217
9.1.1.	Preprocessing.....	218
9.1.2.	Data Enhancement.....	218
9.1.3.	Input data.....	220
9.2.	Hyperparameters tuning.....	226
9.2.1.	Sensitivity to the input window size.....	227
9.2.2.	From probabilities to sequence label.....	229
9.3.	How can we improve the results ?.....	232
9.3.1.	Implementing pre-validation approaches.....	232
9.3.2.	Multiclass classification.....	236
9.3.3.	Predicting the anomaly rate per sequence.....	238
9.4.	Generalization to other sites.....	242
9.4.1.	Direct evaluation on other sites.....	242
9.4.2.	Training the best model using data from different sites.....	243
9.4.3.	Tuning a specific model for Roosevelt.....	246
9.5.	Multivariable anomaly detection.....	247
9.6.	Synthesis of Chapter 9.....	249
<b>Chapter 10.</b>	<b>Autoencoder evaluation.....</b>	<b>251</b>
10.1.	Sensitivity to input data.....	251

10.1.1.	Preprocessing .....	252
10.1.2.	Input data .....	254
10.1.3.	Size of the input database .....	255
10.2.	Hyperparameters tuning.....	257
10.2.1.	Sensitivity to the feature size .....	258
10.2.2.	Testing different architectures .....	262
10.2.3.	Window size .....	269
10.3.	How can we improve the model's performance ? .....	272
10.3.1.	Increasing database size.....	272
10.3.2.	Adjusting classification rules.....	273
10.3.3.	Ensemble model using the best architectures .....	280
10.3.4.	Implementing pre-validation approaches .....	283
10.4.	Generalization to other sites.....	286
10.4.1.	Direct evaluation of the best models.....	286
10.4.2.	Training Model B using data from different sites .....	287
10.4.3.	Tuning a specific model for each site.....	288
10.5.	Multivariable approach for anomaly detection .....	290
10.6.	Synthesis of Chapter 10 .....	292
<b>Chapter 11.</b>	<b>Stretching Boundaries of Data Validation using AI .....</b>	<b>295</b>
11.1.	Relation between annotator agreement and model's performance .....	295
11.2.	Model's extrapolation to new chronicle.....	298
11.3.	Conductivity validation using Matrix Profile.....	301
11.4.	Water level validation using Matrix Profile .....	305
11.5.	Synthesis of Chapter 11 .....	310
	<b>Synthesis of Part III : Results and Discussion .....</b>	<b>311</b>
	<b>Conclusion and perspectives .....</b>	<b>315</b>
	Retracing steps : A comprehensive overview .....	315
	And what about the operational dimension? .....	318
	Acknowledging the Study's Limitations .....	320
	Mapping future perspectives.....	321
	<b>Résumé étendu.....</b>	<b>325</b>
1.	Introduction.....	325
1.1.	Contexte et enjeux .....	325
1.2.	Objectifs et contenu de la thèse .....	326
2.	État de l'art .....	327
2.1.	La validation des données en assainissement .....	327
2.2.	La détection des anomalies avec l'intelligence artificielle .....	329

3.	Matériel et méthodes .....	331
3.1.	Construction de la base de données .....	331
3.2.	Benchmark des modèles et des tests.....	334
4.	Résultats et discussion .....	338
4.1.	Évaluation de l'accord entre les experts (assistés par la redondance).....	338
4.2.	Évaluation du modèle Matrix Profile .....	339
4.3.	Évaluation du modèle ResNet.....	344
4.4.	Évaluation du modèle Autoencodeur.....	348
4.5.	Comparaison des différents modèles .....	354
4.6.	Généralisation et ouverture .....	355
5.	Conclusion.....	356
	<b>Bibliography.....</b>	<b>357</b>
	<b>Appendices.....</b>	<b>371</b>
	<b>Appendix A. Activation functions.....</b>	<b>373</b>
	<b>Appendix B. Neural Networks Architectures .....</b>	<b>378</b>
	<b>Appendix C. Stochastic Gradient Descent.....</b>	<b>381</b>
	<b>Appendix D. Survey on anomaly detection using AI in the hydrological field. ....</b>	<b>382</b>
	<b>Appendix E. Commonly used loss functions .....</b>	<b>386</b>
	<b>Appendix F. Time series decomposition .....</b>	<b>388</b>
	<b>Appendix G. Matrix Profile algorithms .....</b>	<b>389</b>
	<b>Appendix H. Class Activation Maps .....</b>	<b>395</b>
	<b>Appendix I. Visualizing Data using t-SNE.....</b>	<b>398</b>
	<b>Appendix J. Pairwise F1 score results between annotators .....</b>	<b>400</b>
	<b>Appendix K. ANOVA test.....</b>	<b>402</b>
	<b>Appendix L. ResNet results using the multivariable approach.....</b>	<b>403</b>

# Table of Figures

Figure 0-1: Different stakeholders and measurement objectives in wastewater networks.....	3
Figure 0-2: Sensors installed in wastewater network.....	4
Figure 0-3: Graphical representation of widespread malfunctions .....	5
Figure 0-4: Levels of artificial intelligence .....	8
Figure 1-1: Outline of urban wastewater network with the identification of different measurement points .....	15
Figure 1-2: Measurement chain.....	17
Figure 1-3: Example of supervision software: Eve'M .....	19
Figure 1-4: Example of a typical dry weather data pattern in a wastewater network .....	20
Figure 1-5: Examples of different type of anomalies in red. ....	23
Figure 1-6: Example of turbidity variations, recorded using three sensors.. ....	27
Figure 2-1: Example of pre-validation software .....	31
Figure 2-2: Example of graphical macro-analysis operated by an expert in order to validate flow measurement. ....	33
Figure 2-3: Illustration of the principle of statistical anomaly detection on synthetic data. ....	35
Figure 2-4: Example of 3D modelling of a double storm overflow .....	38
Figure 2-5: Stages of hydrodynamic modelling.....	39
Figure 2-6: Data validation: 3D modelled flow Vs on-site measurement .....	39
Figure 2-7: Overview of data validation approaches and their limits in wastewater networks	42
Figure 3-1: The history of artificial intelligence.....	44
Figure 3-2: Distinction between supervised and unsupervised learning.....	46
Figure 3-3: Perceptron architecture .....	48
Figure 3-4: Artificial neuron architecture .....	48
Figure 3-5: Architecture of a neural network .....	49
Figure 3-6: Learning process of a neural network.....	51
Figure 3-7: Example of loss function with a local minima different from the global one .....	52
Figure 3-8: AI Implementation process .....	53
Figure 3-9: State-of-the-art AI approaches for anomaly detection in the urban hydrological field with their occurrence in Appendix D.....	58
Figure 3-10: Example of MLP architecture for anomaly detection.....	61
Figure 3-11: Example of a Deep CNN for time series processing .....	63
Figure 3-12: The principle of OCSVM.....	64
Figure 3-13: Example of the IForest principle .....	65
Figure 3-14: Example of local density for LOF model .....	67
Figure 3-15: A diagram for a one-unit recurrent neural network (RNN).....	68

Figure 3-16: Architecture of Autoencoders .....	69
Figure 3-17: The basic scheme of a variational autoencoder.....	70
Figure 3-18: K-Means Principle .....	71
Figure 3-19: Example of Matrix Profile for anomaly detection .....	72
Figure 3-20: A residual block of the ResNet model.....	73
Figure 4-1: Saint Malo Agglomeration and number of inhabitants per municipality .....	79
Figure 4-2: The six main interceptors of Saint Malo Agglomeration .....	81
Figure 4-3: Example of defects at the interceptor "Goutte" .....	85
Figure 4-4: Data structure at the "Roosevelt" interceptor .....	87
Figure 4-5: Turbidity data decomposition.....	88
Figure 4-6: Data acquisition process for variables of interest.....	89
Figure 4-7: Organization of the validation process.....	90
Figure 4-8: Results of manual validation.....	93
Figure 4-9: Average turbidity at different sites, differencing valid and invalid data. ....	94
Figure 4-10: Overview of database preparation for models' evaluation.....	96
Figure 5-1: Application of the 3-sigma rule to turbidity data in Cottage .....	97
Figure 5-2: Diagram of the different tests established for each model from our benchmark. .....	100
Figure 5-3: Grid Search for hyperparameters tuning.....	101
Figure 5-4: Required calculation time depending on data length and acquisition frequency	103
Figure 5-5: A subsequence Q extracted from a time series T is used as a query.....	104
Figure 5-6: A time series T, and its self-join matrix profile P .....	105
Figure 5-7: Brute Force Matrix Profile.....	105
Figure 5-8: Examples of matrix profile interpretation.....	106
Figure 5-9: Matrix Profile for multidimensional time series .....	108
Figure 5-10: Example of output of the "Matrix Profile" model.....	110
Figure 5-11: Comparison of the model results and those issued from the manual validation .....	110
Figure 5-12: Ensemble model using majority vote. ....	111
Figure 5-13: Overview of tests related to Matrix Profile model .....	113
Figure 5-14: Error rate history on ImageNet .....	114
Figure 5-15: Convolutional Neural Network Architecture .....	115
Figure 5-16: Different padding approaches (here the stride = 2).....	116
Figure 5-17: Pooling approaches.....	117
Figure 5-18: Residual learning: a building block .....	118
Figure 5-19: Batch Normalization process.....	119
Figure 5-20: Architecture of the used ResNet model .....	119

Figure 5-21: Global Average Pooling principle.....	120
Figure 5-22: Classic data enhancement strategies.....	121
Figure 5-23: K-fold cross validation strategy.....	123
Figure 5-24: TensorBoard .....	124
Figure 5-25: Class Activation Map of a sequence.....	125
Figure 5-26: ResNet architectures used in this study.....	126
Figure 5-27: Transfer Learning strategies.....	127
Figure 5-28: Overview of tests related to ResNet model.....	128
Figure 5-29: Comparison of the results of dimensionality reduction using PCA and Autoencoder .....	129
Figure 5-30: Baseline of an autoencoder model .....	130
Figure 5-31: Deep autoencoder architecture for data validation.....	132
Figure 5-32: Different architectures of AE tested in this study.....	134
Figure 5-33: Example of output of the AE model .....	136
Figure 5-34: Latent space visualization of the same model at different epochs .....	137
Figure 5-35: Classification approaches: (Left): 3-sigma rule - (Right): PR Curve approach	138
Figure 5-36: Overview of tests related to Autoencoder model.....	140
Figure 5-37: Benchmark of the model tested for anomaly detection in turbidity data .....	141
Figure 6-1: Diagram representation of precision and recall.....	144
Figure 6-2: Relationship between MCC and F1 score, depending on the anomaly ratio. ....	147
Figure 6-3: ROC Curve.....	148
Figure 6-4: PR Curve.....	149
Figure 6-5: Confusion matrices based on the reference expert.....	152
Figure 6-6: Illustration of the principle of hierarchical clustering and the establishment of the corresponding dendrogram.....	154
Figure 6-7: Overview of section II: Materials and methods .....	160
Figure 7-1: Global pairwise confusion matrices issued from the validation pool.....	164
Figure 7-2: Anomaly rate by site according to each expert on the basis of data used by the validation pool.....	165
Figure 7-3: Pairwise F1 scores among the validation pool.....	165
Figure 7-4: Global Dendrogram .....	167
Figure 7-5: Dendrogram of the validation of Cottage - March .....	168
Figure 7-6: Anomaly identified by experts B, C & D .....	169
Figure 7-7: Dendrogram of the validation of Cottage - July.....	169
Figure 7-8: Global Pairwise Cohen's Kappa .....	170
Figure 7-9: Pairwise confusion matrices for Roosevelt - November.....	171
Figure 7-10: Pairwise Cohen's Kappa for Roosevelt – November.....	171



Figure 7-11: Expert validation results of November chronicle - Roosevelt .....	172
Figure 7-12: Pairwise confusion matrices for Roosevelt – May .....	173
Figure 7-13: Pairwise Cohen's Kappa for Roosevelt – May .....	173
Figure 7-14: Ratio of timesteps having a minimum level of annotator agreement .....	174
Figure 7-15: Ratio of invalid timestamps following the level of agreement .....	175
Figure 7-16: Expert validation results of March chronicle – Découverte.....	177
Figure 7-17: Overview of Chapter 7 - Evaluation of annotator agreement among the validation pool .....	179
Figure 8-1: Results of different missing values imputation techniques. ....	183
Figure 8-2: The results of downsampling on anomaly detection. ....	184
Figure 8-3: Grid Search Results to identify best hyperparameters for Cottage Dataset. ....	188
Figure 8-4: Matrix profile for turbidity data with a window length of 2 hours and an anomaly ratio of 0.5%. Red stars point potential anomalies .....	189
Figure 8-5: Example of manual invalid sequences and of the bias introduced by the filtering phase and the redundancy criterion.....	190
Figure 8-6: Comparison of validation results by the domain expert and the algorithm using Boolean time series. ....	191
Figure 8-7: Example of anomalies fusion while the in-between subsequence is valid.....	191
Figure 8-8: Example of bias introduced by a fixed window size using Matrix Profile .....	192
Figure 8-9: Recall of matrix profile according to the defined anomaly rate .....	193
Figure 8-10: Example of trivial repetitive anomalies non-identified by the MP model.....	194
Figure 8-11: Anomalies delimitation problem between the expert end the algorithm validation .....	194
Figure 8-12: Rainfall history at Saint Malo .....	195
Figure 8-13: Example of a false positive subsequence. ....	195
Figure 8-14: ROC Curves for the three window sizes used in ensemble model .....	198
Figure 8-15: Sensitivity of the ensemble model with majority vote to the anomaly ratio .....	199
Figure 8-16: Sensitivity of the ensemble model with minority voting to the anomaly ratio ...	199
Figure 8-17: Validation results at the daily sequence level using our best monovariable MP model.....	201
Figure 8-18: Grid search results to identify best hyperparameters for Goutte dataset.....	202
Figure 8-19: Grid search results to identify best hyperparameters for Découverte dataset.	202
Figure 8-20: Example of two different abnormal sequences of 24-hours in Découverte dataset .....	203
Figure 8-21: Grid search results to identify best hyperparameters for Roosevelt dataset....	203
Figure 8-22: Comparison of anomaly detection using P1 and P2 ( $w = 48$ hours and $k = 0.1$ ) .....	205

Figure 8-23: Calculation of P1 and P2 for anomaly detection .....	206
Figure 8-24: Data validation using the raw turbidities as input data .....	207
Figure 8-25: Anomaly detection using raw turbidity from the two sensors (T1 & T2) and P1 .....	207
Figure 8-26: ROC curve for bivariate matrix profile P1 using reconstructed turbidity and conductivity.....	208
Figure 8-27: F1 score using multivariate matrix profile depending on the window size .....	209
Figure 8-28: Precision and recall using adjusted multivariate matrix profile depending on the anomaly ratio .....	211
Figure 8-29: Anomalies such as identified on the three chronicles using the adjusted multivariate matrix profile .....	212
Figure 8-30: Metrics using global model depending on the anomaly ratio.....	213
Figure 8-31: Illustration of the global model process and its impact on the anomaly rate....	214
Figure 8-32: Overview of Matrix Profile tests and results for anomaly detection using turbidity data .....	216
Figure 9-1: Variation of the F1 score between different folds depending on the input database .....	220
Figure 9-2: Results for different input data using the ResNet model .....	221
Figure 9-3: Example of False Negative, biased by the filtering criterion .....	222
Figure 9-4: Example of False Positive sequence with saturation .....	222
Figure 9-5: Example of False Positive sequence with null data .....	223
Figure 9-6: Example of False Negative sequence .....	224
Figure 9-7: Number of sequences according to their inherent anomaly rate .....	225
Figure 9-8: Results of sensitivity tests to the input sequence size .....	228
Figure 9-9: Classification results of all sequences using the model issued according to different loss functions.....	230
Figure 9-10: Synopsis of the classification task enhanced with pre-validation .....	233
Figure 9-11: Enhanced model results combining the ResNet and a pre-validation phase...	234
Figure 9-12: Performance metrics according to the classification threshold applied to the expert 5-minutes scale validation and the results of the enhanced ResNet model.....	235
Figure 9-13: Multiclass classification using ResNet and a threshold of 20% .....	236
Figure 9-14: Multiclass classification using an adjusted score .....	237
Figure 9-15: Multiclass classification using ResNet and a threshold of 40% .....	238
Figure 9-16: Histogram of anomaly rates predicted by the model versus true anomaly rates .....	238
Figure 9-17: Comparison of true anomaly rate per sequence and the predicted anomaly rate .....	239

Figure 9-18: CAM results compared to true anomaly rate. the CAM output in red, its average per sequence in green, and the anomaly rate identified by the expert in blue .....	240
Figure 9-19: From anomaly rate per sequence to classification: F1 score results according to the invalidation threshold .....	241
Figure 9-20: Confusion Matrix of the ResNet model trained on Cottage and evaluated on Roosevelt .....	243
Figure 9-21: Comparison of the learning curves using different training strategies .....	245
Figure 9-22: Data validation results of Cottage data using a multivariable approach .....	248
Figure 9-23: Overview of ResNet tests and results for anomaly detection using turbidity data .....	250
Figure 10-1: ROC Curves for input sensitivity test on the autoencoder .....	253
Figure 10-2: Confusion matrix and performances depending on the input data – <i>Left</i> : reconstructed data, <i>Right</i> : raw data .....	255
Figure 10-3: Performance metrics according to the ratio of input data used for training .....	256
Figure 10-4: Confusion matrix and performances depending on the ratio of database .....	257
Figure 10-5: Results of sensitivity to the latent space size using the 3-sigma rule .....	259
Figure 10-6: Results of sensitivity to the latent space size using the PR curve approach ...	260
Figure 10-7: ROC Curves for the different models of architecture 1 .....	261
Figure 10-8: Learning curves for the model n°6 .....	262
Figure 10-9: Tests of overfitting based on model's complexity .....	262
Figure 10-10: MCC results of Deep-AE architectures following the total number of neurons .....	263
Figure 10-11: MCC results of AE models with 3 hidden layers .....	264
Figure 10-12: Learning curves depending on the model's complexity .....	265
Figure 10-13: ROC curves comparing models with the same code size but different architectures .....	266
Figure 10-14: Performance metrics for models with the same code size but different architectures .....	266
Figure 10-15: Reconstruction of valid sequence n°121 and valid sequence n° 119 using the models 6 and 10 .....	267
Figure 10-16: Reconstruction of valid sequence n°122 and invalid sequence n° 129 using the models 6 and 10 .....	268
Figure 10-17: F1 score for different window sizes according to the trained model and the classification approach .....	270
Figure 10-18: Performance metrics for different strides using Model B .....	271
Figure 10-19: Performance metrics at the measure scale for different strides using Model B .....	272

Figure 10-20: Performance metrics according to the database size.....	273
Figure 10-21: Histogram of MSE on training data samples (in red) and the corresponding normal distribution (in blue) using Model B .....	275
Figure 10-22: Performance metrics according to the classification threshold as an anomaly ratio per sequence. 0 refers to an invalid sequence from one invalid time step.....	276
Figure 10-23: Ratio of false negatives for different anomaly ratios such as identified by model 5 and using a classification threshold of 0.04.....	277
Figure 10-24: Noisy saturation sequence invalidated by the expert and the AE.....	277
Figure 10-25: Drift sequence invalidated by the expert and the AE.....	278
Figure 10-26: Invalid sequences according to the expert, validated by the AE model.....	279
Figure 10-27: Correlation between reconstruction error (MSE) and anomaly rate per sequence .....	280
Figure 10-28: t-SNE visualization of the code of Model A and Model B .....	281
Figure 10-29: Confusion matrices using consensus and based on the PR curve approach	282
Figure 10-30: Results using the 3-sigmars rule (solid bars) compared to the results of the ensemble model using the PR approach (dotted bars) .....	282
Figure 10-31: Synopsis of the classification task enhanced with pre-validation .....	283
Figure 10-32: Enhanced model results combining the AE and a pre-validation phase.....	284
Figure 10-33: Heatmap of the F1 score according to the classification threshold applied to the expert 5-minutes scale validation and the results of the enhanced model.....	285
Figure 10-34: Results at different sites and their comparison according to their anomaly rate .....	287
Figure 10-35: Evaluation of the generic model on the whole dataset and on the specific data of each site .....	288
Figure 10-36: Performance metrics of specific models, evaluated on their site database using the PR curve approach .....	289
Figure 10-37: Performance metrics of specific models, evaluated on their site database using the 3-sigma rule.....	289
Figure 10-38: F1 score results using dense layers. T3 refers to the reconstructed turbidity	291
Figure 10-39: F1 score results using convolutional layers. T3 refers to the reconstructed turbidity.....	291
Figure 10-40: Overview of Autoencoder tests and results for anomaly detection using turbidity data .....	293
Figure 11-1: Results of the AE model using different baselines issued from different experts .....	296
Figure 11-2: Results of the AE model using different annotator agreement's rates.....	297
Figure 11-3: Manual validation results of Cottage turbidity data.....	299

Figure 11-4: Ratio of normal training sequences .....	300
Figure 11-5: Evaluation results of the model trained on the complete Cottage database ....	300
Figure 11-6: Anomaly detection on conductivity data using matrix profile .....	301
Figure 11-7: Zoom in on the most abnormal sequence in the conductivity dataset using a 12- hours sequence .....	302
Figure 11-8: Overall validation of conductivity data using Matrix Profile.....	303
Figure 11-9: Typical conductivity pattern in wastewater networks during rainy events .....	304
Figure 11-10: True anomaly identified by Matrix Profile .....	304
Figure 11-11: False anomaly identified by Matrix Profile.....	305
Figure 11-12: Spillway configuration with the localization of the US sensors .....	306
Figure 11-13: Prioritization of anomalies by concatenating results from different window sizes .....	307
Figure 11-14: Concatenation of data validation of sensor 1 results using a multi-window size approach and $k = 10$ .....	307
Figure 11-15: Zoom 1: The anomaly with the highest score using a multi-window size approach .....	308
Figure 11-16: Zoom 2: The main anomaly during the rainy weather using the multi-window size approach .....	308
Figure 11-17: Concatenation of data validation of sensor 2 results using a multi-window size approach and $k = 10$ .....	309
Figure 1: Benchmark of the evaluated models.....	316
Figure 2: Résultat des meilleurs architectures d'autoencodeur pour la détection des anomalies .....	350
Figure A-1: Binary step function .....	373
Figure A-2: Identity activation function.....	374
Figure A-3: Sigmoid function .....	375
Figure A-4: Tanh activation function .....	376
Figure A-5: ReLU activation function .....	376
Figure A-6: Leaky ReLU activation function.....	377
Figure B-1: Simple architecture of GAN.....	379
Figure B-2: Infographic with different neural networks architecture (2016).....	380
Figure C-1: Gradient Descent Principle .....	381
Figure H-1: Standard approach, where the feature maps are flattened in order to fed them into a dense layer .....	395
Figure H-2: Global average pooling operation .....	396
Figure H-3: Linear combination of the weights and feature maps to obtain the class activation map .....	396

Figure J-1: Pairwise confusion matrices issued from the validation pool per month.....	400
Figure J-2: Pairwise confusion matrices issued from the validation pool per site .....	401
Figure K-1: Types of variation analyzed using the ANOVA test .....	402
Figure L-1: Heatmap using both raw turbidity data (2T) as input.....	403
Figure L-2: Heatmap using raw turbidity and reconstructed turbidity data (3T) as input .....	404
Figure L-3: Heatmap using raw turbidity and conductivity data (2TC) as input.....	405
Figure L-4: Heatmap using raw turbidity, reconstructed turbidity and conductivity data (3TC) as input.....	406

# Table of Tables

Table 1: Examples of measurement probes in wastewater networks.....	16
Table 2: Some references for urban hydrology data validation using statistical approaches.	36
Table 3: Analysis grid for different data validation tools / models.....	76
Table 4: Turbidity data statistics. The unit of all values is FNU .....	84
Table 5: Pearson Correlation Matrix .....	86
Table 6: Labels of the reconstructed turbidity T, considering the labels of T1 and T2.....	92
Table 7: Results of the 3-sigma rule for anomaly detection .....	98
Table 8: Benchmark of the tested AI models .....	99
Table 9: Nomenclature for the evaluated multivariable approaches and their input data.....	102
Table 10: List of AE models evaluated in this study .....	135
Table 11: Overview of different tests .....	142
Table 12: Confusion Matrix.....	143
Table 13: Interpretation of Cohen's Kappa coefficient.....	152
Table 14: Hardware specification .....	156
Table 15: Model performance and annotator agreement metrics.....	158
Table 16 : Mean F1 score differences for each expert.....	166
Table 17: Smyth's lower error bound for site and month- specific scenarios .....	176
Table 18: Missing values imputation techniques results .....	182
Table 19: Downsampling results.....	184
Table 20: Data smoothing results .....	185
Table 21: Performance metrics for different input data using the same hyperparameters...	186
Table 22: Performance results of MP using different input data.....	189
Table 23: Results of the combination of anomaly detection using raw data and selecting only common defects .....	197
Table 24: Majority voting Results.....	199
Table 25: Minority voting results .....	200
Table 26: Inherent anomaly ratio of different database .....	201
Table 27: Matrix profile generalization results to other sites.....	204
Table 28: Matrix profile results using redundancy .....	205
Table 29: Interpretation of P1 and P2 results.....	206
Table 30: Matrix Profile Results on turbidity and conductivity .....	209
Table 31: Results of multivariate matrix profile for a window size of 24 hours.....	209
Table 32: Results of adjusted multivariate matrix profile .....	211
Table 33: Results of ResNet model for different enhancement approaches.....	219
Table 34: Results of ResNet model using different sites as input .....	219

Table 35: List of tests applied to input data.....	225
Table 36: Training results of ResNet using different inputs and a 5-folds cross validation ..	226
Table 37: Evaluation results of ResNet model using F score as a loss function.....	229
Table 38: Results obtained with the a posteriori adjusted threshold.....	230
Table 39: F1 score obtained with the adjusted threshold after re-training .....	231
Table 40: F1 score obtained with approximated adjusted thresholds.....	231
Table 41: Direct evaluation of ResNet on other sites of SMA.....	242
Table 42: Results using a ResNet model trained on the whole database from different sites .....	244
Table 43: Results using a total-fine tuning with data from all sites .....	245
Table 44: Results of Roosevelt data validation using a specific model and different learning strategies.....	246
Table 45: Results of the multivariable approach according to the input data.....	248
Table 46: List of tests conducted to evaluate AE's sensitivity to input data preprocessing ..	252
Table 47: Results of input processing - autoencoder .....	252
Table 48: Results for different input data using the PR approach .....	254
Table 49: Evaluation of the x sigma required to achieve the best performances.....	274
Table 50: Ensemble model results using the average MSE combined to a PR curve approach .....	281
Table 51: Performance metric of direct evaluation of the best models on other sites data ..	286
Table 52: Evaluation results of the Cottage-trained model on the Cottage and New Cottage databases.....	298
Table 53: Evaluation results of the New Cottage-trained model on the Cottage and New Cottage databases.....	298
Table 54: Overview of model's strengths and weaknesses for anomaly detection in wastewater data .....	312
Table 55: Best results of the evaluated models using 24-hours sequences .....	317
Table 56: Synthèse des modèles évalués .....	355
Table 57: Mueen's Algorithm for similarity Search (MASS).....	390
Table 58: The Stamp Algorithm .....	390
Table 59: The STOMP Algorithm.....	392
Table 60: Time required for motif discovery varying the dataset length n ( $m = 256$ ) .....	392
Table 61: The mSTAMP Algorithm .....	393
Table 62: Concise description of the matrix profile algorithms used in this study.....	394
Table 63: Comparison between t-SNE and PCA for dimension reduction.....	399



# ***Table of Equations***

Equation 1: Autoregressive model equation .....	35
Equation 2: Pearson Correlation Coefficient .....	85
Equation 3: Consistency criterion for turbidity validation based on redundancy .....	90
Equation 4: Normalization formula.....	101
Equation 5: Standardization formula.....	101
Equation 6: Residual mapping approach .....	118
Equation 7: Mean Squared Error (MSE) .....	132
Equation 8: Root Mean Squared Error (RMSE) .....	133
Equation 9: Accuracy.....	144
Equation 10: Precision.....	144
Equation 11: Recall .....	145
Equation 12: $F_{\beta}$ Score .....	145
Equation 13: Matthew's Correlation Coefficient .....	146
Equation 14: True Positive Rate .....	148
Equation 15: False Positive Rate.....	148
Equation 16: Cohen's kappa coefficient.....	151
Equation 17: Cohen's kappa coefficient for binary classification .....	151
Equation 18: Error lower bound .....	153
Equation 19: Binary Cross Entropy Loss function .....	386
Equation 20: Categorical Cross Entropy Loss Function .....	387
Equation 21: L2 Loss Function .....	387
Equation 22: L1 Loss Function .....	387
Equation 23: Normalized Euclidean Distance .....	389

## Acronyms and abbreviations

<b>WWTP</b>	Wastewater treatment plant
<b>DERU</b>	Council Directive 91/271/EEC of 21 May 1991 concerning urban waste-water treatment
<b>DCE</b>	Directive 2000/60/EC of the European Parliament and of the Council of 23 October 2000 establishing a framework for Community action in the field of water policy
<b>CNIL</b>	Commission nationale de l'informatique et des libertés
<b>IoT</b>	Internet of Things
<b>SMA</b>	Saint Malo Agglomeration
<b>SS</b>	Suspended solids
<b>COD</b>	Chemical oxygen demand
<b>BOD</b>	Biological oxygen demand
<b>FNU</b>	Formazan Nephelometric Unit, a unit of turbidity measurement
<b>FAU</b>	Formazan Attenuation Unit
<b>CFD</b>	Computational Fluid Dynamics
<b>AR</b>	Autoregression
<b>ARIMA</b>	Autoregressive integrated moving average
<b>AI</b>	Artificial intelligence
<b>ML</b>	Machine Learning
<b>DL</b>	Deep Learning
<b>k-NN</b>	k-Nearest Neighbors
<b>SVM</b>	Support Vector Machine
<b>OCSVM</b>	One Class Support Vector Machine
<b>IForest</b>	Isolation Forest
<b>LOF</b>	Local Outlier Factor
<b>RF</b>	Random Forest
<b>MP</b>	Matrix Profile
<b>RBF</b>	Radial Basis Function
<b>DNN</b>	Deep Neural Network
<b>MLP</b>	Multi-layers Perceptron
<b>CNN</b>	Convolutional Neural Network
<b>ResNet</b>	Residual Neural Network
<b>AE</b>	Autoencoder
<b>VAE</b>	Variational Autoencoder
<b>GAN</b>	Generative Neural Network
<b>RNN</b>	Recurrent Neural Network
<b>LSTM</b>	Long-short Term Memory
<b>CPU</b>	Central processing unit
<b>GPU</b>	Graphical processing unit

<b>IDE</b>	Integrated development environment
<b>MSE</b>	Mean squared error
<b>RMSE</b>	Root-mean-square error
<b>ROC curve</b>	Receiver Operating Characteristic Curve
<b>PR curve</b>	Precision-Recall Curve
<b>AUC</b>	Area Under Curve
<b>MCC</b>	Mathews Correlation Coefficient
<b>F<sub>ij</sub></b>	F1 score between two experts i and j
<b>FP</b>	False Positive
<b>TP</b>	True Positive
<b>TN</b>	True Negative
<b>FN</b>	False Negative
<b>TPR</b>	True Positive Rate
<b>FPR</b>	False Positive Rate
<b>GT</b>	Ground Truth
<b>GAP</b>	Global Average Pooling
<b>CAM</b>	Class Activation Maps
<b>MLE</b>	Maximum Likelihood estimator
<b>SGD</b>	Stochastic Gradient Descent
<b>ReLU</b>	Rectified Linear Unit
<b>PCA</b>	Principal Component Analysis
<b>t-SNE</b>	t-distributed stochastic neighbor embedding
<b>MASS</b>	Mueen's Algorithm for Similarity Search
<b>FFT</b>	Fast Fourier Transform
<b>ANOVA</b>	Analysis of variance
<b>STAMP</b>	Scalable Time series Anytime Matrix Profile
<b>STOMP</b>	Scalable Time series Ordered-search Matrix Profile
<b>mSTAMP</b>	Multivariate STAMP
<b>mSTOMP</b>	Multivariate STOMP
<b>ILSVRC</b>	ImageNet Large Scale Visual Recognition Challenge
<b>W</b>	Window size – Length of the sequence
<b>T1, T2</b>	Turbidity from the redundant turbidimeters
<b>P1, P1, ..</b>	Profiles issued from the multivariate Matrix Profile
<b>2T</b>	Multivariate approach using raw data from both turbidimeters
<b>3T</b>	Multivariate approach using raw data from both turbidimeters and the reconstructed turbidity
<b>2TC</b>	Multivariate approach using raw data from both turbidimeters and conductivity data
<b>3TC</b>	Multivariate approach using raw data from both turbidimeters, the reconstructed turbidity and conductivity

# General Introduction

---

## Context and issues

Let's take a moment to envision the coastal town of Saint-Malo, France, which is featured in this study for generously providing us with data. Picture this beautiful seaside town with its medieval architecture and the soothing sound of waves. However, behind this picturesque scene lies a different reality. When heavy rain hits, there's a real risk of flooding – rivers overflow, and the town faces the threat of being swamped by the sea [1]. However, this is just one facet of the threats that loom over Saint-Malo. In the heart of the city, a more insidious danger emerges through the sewer systems. During heavy rain, the water levels within these networks rise significantly, leading to overflows and surreptitious infiltration through low points in the system, impacting infrastructure and residents' quality of life.

In response to this latent threat, the city has deployed an array of control infrastructure, including valves, non-return valves, and retention basins, aimed at preserving residents' safety during potential flooding events [2]. However, this challenge is not exclusive to Saint-Malo but extends to all municipalities with combined sewer systems, as well as those with undersized stormwater networks [3]. Once the peak of precipitation has passed, whether treated or untreated, water must inevitably be returned to the natural environment. Treatment facilities are not designed to process all rainwater due to its high flow rate and lower pollutant concentration [4]. For swimmers, shellfish farmers, and aquatic ecosystem whose health and livelihoods depend on the quality of the natural environment, these discharges of polluted water pose a serious threat. In a global context marked by climate change and water scarcity, preserving the quality of natural environments becomes critical, both for the sustainability of our communities and the protection of biodiversity [5] - [6].

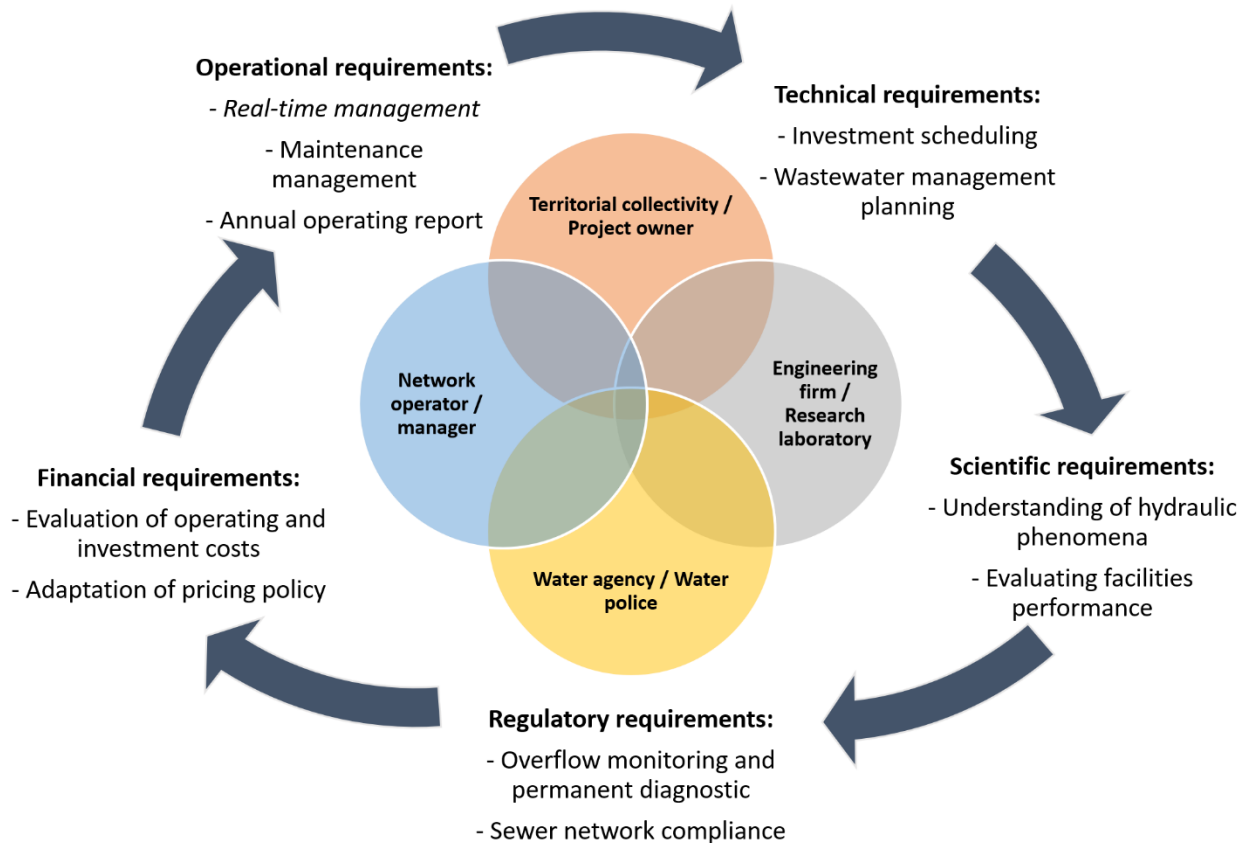
Thus, sewer networks serve a dual role, aiming to protect the population against flooding and to reduce pollutants discharged into receiving environments, including during rainfall events. This mission aligns with the ongoing revision of the European directives on wastewater (DERU) of 1991, the Water Framework Directive (DCE) of 2000, and the decree of July 31, 2020 [7] regarding overflow obligations and compliance criteria for combined sewer systems. According to these regulations, either the volumes of urban discharges or the pollutant flows released into the natural environment must be continuously monitored, not exceeding 5% of the total annual production of a sewer system. It is likely that this threshold will be reduced in the future [8].

In this context, the monitoring of sewer networks takes on crucial importance: how can we monitor and assess the proper functioning of a sewer network, ensuring the safety of residents while preserving the quality of natural environments? The answer to this complex question starts with the use of ...

**...sensors, a research area at the heart of this study.**

Beyond the response to regulatory constraints and the production of quantitative data to assess pollutant pressures on water bodies, the need for knowledge regarding the origins, transfers, and flows of pollutants in urban hydrology unites various stakeholders in the field [9]. This quest for knowledge assumes multifaceted dimensions contingent upon the distinct perspective of each participant (see [Figure 0-1](#)) [10]:

- *Regulatory Dimension for Local Authorities and Control Entities:* Local authorities and regulatory bodies are under the obligation of adhering to stringent directives governing water management and pollution reduction. In order to ensure compliance with environmental standards, they require precise data and insights into the sources and distribution of pollutants within sewage systems. This, in turn, assists in the formulation of appropriate policies and regulations aimed at safeguarding water bodies and public health.
- *Operational and Financial Aspect for Network Managers:* Network managers overseeing wastewater systems face substantial operational and financial challenges. Understanding the flows of pollutants is imperative for optimizing network maintenance, reducing the risk of flooding, minimizing unauthorized discharges, and ensuring the efficient use of resources. Enhanced management translates into significant financial savings while maintaining network performance.
- *Technical Component for Specialized Consultancies:* Specialized consulting firms are often engaged in designing infrastructure enhancements geared towards improving sewage network performance. To do so effectively, they must possess a comprehensive understanding of hydraulic functioning of the structure and pollutant flows. This technical knowledge is crucial for the design of appropriate infrastructure, tailored to the specific conditions of each structure and network.
- *Scientific Dimension for Researchers:* Researchers are focused on unraveling the dynamics of hydrological phenomena and comprehending the impact of pollution on aquatic ecosystems. Their quest for understanding extends beyond regulatory and operational requisites. They delve into the underlying mechanisms, develop predictive models, and contribute to advancing knowledge in the field of urban hydrology.



**Figure 0-1: Different stakeholders and measurement objectives in wastewater networks. Note: In italics, requirements using real-time data. Others use data in offline mode.**

Understanding the hydraulic functioning of these networks hinges on a baseline approach: the monitoring of key points (which may be completed by other approaches such as modelling). Measurement systems within wastewater networks typically rely on a set of permanent monitoring points [11]. This monitoring necessitates the acquisition of various data at relatively fine time intervals (typically a few minutes), including [12]:

- Precipitation intensity
- Water level, flow velocity & flow rate
- Quality measurement (turbidity, SS, temperature, H<sub>2</sub>S, pH, conductivity, etc.).
- Operating times of specific equipment (pumps, weirs, etc.)

This diversity of sensors used in the monitoring of wastewater networks has given rise to a complex measurement infrastructure. This complexity arises from the fact that even for the measurement of a single parameter, various types of sensors can be deployed [13]. For instance, the water level can be measured using ultrasonic probes, radar sensors, or piezoelectric sensors. Each of these sensors presents specific advantages and drawbacks in

terms of reliability, precision, measurement range, and cost. Furthermore, the installation conditions of sensors vary significantly. While some sensors remain permanently submerged, others are consistently above the water surface. Certain sensors exhibit a high degree of sensitivity to the installation environment (temperature, bubbles, suspended solids, ...), necessitating regular maintenance to ensure their proper functioning. This diversity in sensor types and installation conditions presents an additional challenge in the management of the measurement infrastructure.

### The wastewater network: an environment unlike any other



**Figure 0-2: Sensors installed in wastewater network.**

© (left) Duke's – (middle) Inside Water Magazine – (right) 3D EAU.

Wastewater networks present a notably harsh environment for sensors, giving rise to multiple malfunctions and substantial challenges:

- *Clogging:* Sensor clogging stands out as one of the most frequent issues. Owing to the presence of debris, sludge, grease, and other solid matter in wastewater, as well as microbial activity developing on immersed surfaces, sensors, particularly immersed ones, are prone to rapid obstruction.
- *Corrosion:* The sewage network environment is often highly corrosive due to the presence of aggressive chemical substances. Immersed sensors are exposed to waters with varying pH levels, chlorides, and other corrosive compounds, which can lead to swift deterioration of sensor components.
- *Electronic Failures:* Wet atmosphere and hydrogen sulfide emanations caused by bacterial activity generate challenging conditions for every equipment: internal electronic components of sensors, such as printed circuits, chips, or temperature sensors, can experience failures. Additionally, electrical, or mechanical connections between the sensor and the data collection system may exhibit defects, causing disruptions in data transmission.

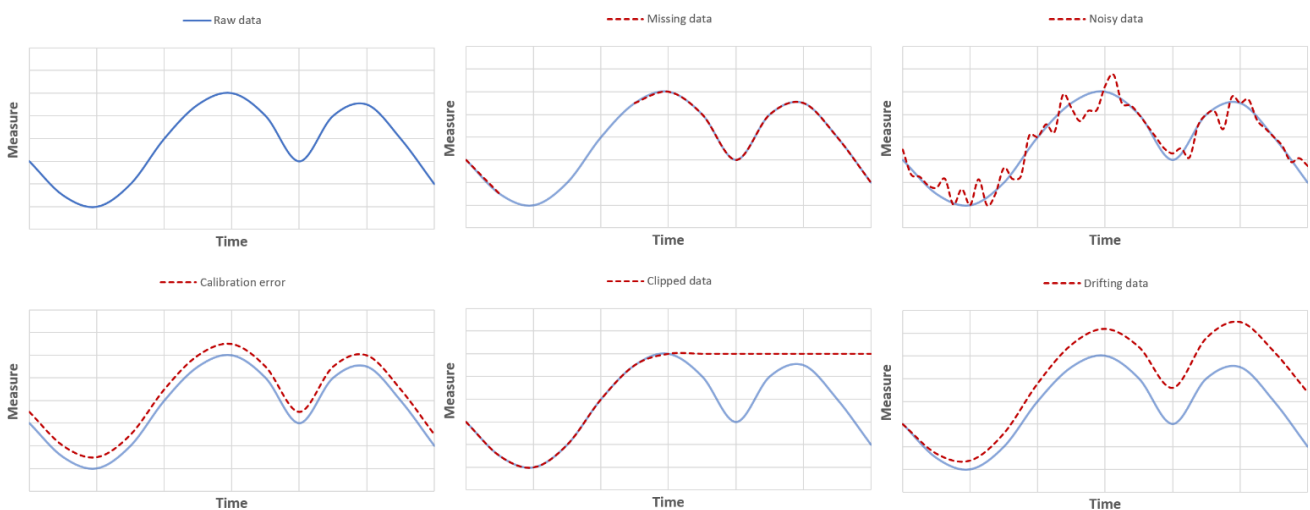
- *External Environment:* Exposed sensors, placed in pumping stations or inspection chambers, face exposure to weather conditions, temperature fluctuations, and extreme climatic factors. These factors can impact sensor reliability and necessitate adequate protection.

### A defective sensor equals invalid data

Typically, sensors are devices designed to transduce a physical quantity into a signal format interpretable by computer systems. They function as essential interfaces between a system and its external environment, providing insights into the state and behavior of the ongoing process. In the event of a defect, an inaccurate representation of the physical quantity being measured results. Consequently, a failure in the measurement system leads to the generation of imprecise and ineffective measured signals/data [14].

Among the types of invalid data, the following can be observed (see [Figure 0-3](#)):

- *Missing Data:* Absent values when they are expected according to the recording strategy represent a clear information loss which can compromise the integrity of data if a constant frequency is required.
- *Noisy Data:* Random fluctuations or interference that make the data challenging to interpret.
- *Calibration Errors:* Poorly calibrated sensors can provide inaccurate measurements, introducing bias or offset.
- *Saturated/Clipped Data:* Data points that reach the upper or lower limits of the sensor's measurement ranges.
- *Drifting Data:* Data that exhibits a gradual shift or change in values over time.



**Figure 0-3: Graphical representation of widespread malfunctions**



### Problems arising from invalid data

These types of data issues can significantly impact the reliability and accuracy of the information collected by sensors in various applications, including those in the realm of wastewater management and urban hydrology. Here are some concrete examples of data defects and situations in which incorrect data can cause problems:

- *Inconsistent Flow Measurements*: Incorrect flow data can lead to a flawed assessment of the network's ability to manage stormwater, potentially resulting in overflows, flooding, and unauthorized discharges into water bodies.
- *Erroneous Water Quality Evaluation*: Incorrect measurement of parameters such as turbidity, pollutant concentration, or pH can result in errors in water quality assessment. This can have serious consequences for aquatic organisms and public health.
- *Unnecessary Overloads at Wastewater Treatment Plants*: Incorrect data on the pollutant load in wastewater can lead to a deterioration in water treatment quality as well as unnecessary expenses for treatment at the wastewater treatment plant, with a significant financial impact.
- *False Overflow Alarms or Lack Thereof*: Overflow alarm sensors that trigger unnecessarily due to incorrect data can lead to inefficient resource utilization. Conversely, a sensor that fails to trigger during a potential overflow risk can compromise the safety of structures and individuals.

These examples illustrate how incorrect data can disrupt the effective management of wastewater networks, leading to additional costs, public health risks, and environmental damage. Ensuring data reliability in these systems is essential to avoid such issues.

⇒ *Irrespective of the purpose behind data utilization, whether for regulatory, operational, or scientific endeavors, the reliability of data is of paramount importance in the realm of wastewater management. It's worth noting that this reliability is not a given in wastewater networks due to the challenging nature of the installation environment, which can significantly amplify malfunctions and compromise data quality.*

### The quest for reliable data in the wastewater field: what are the current means?

Prior to using the data for hydraulic studies, regulatory document production, overflow monitoring, or modeling purposes, measurements must undergo a validation process aimed at ensuring their reliability [15]. The objective is to identify and eliminate aberrant data.

In most common cases, data validation procedures are carried out manually, using data processing and visualization tools [16]. Various data analysis techniques, ranging from simple statistical rules to more sophisticated methods, may be employed. Generally, two levels of validation are distinguished: automatic validation (pre-validation) occurs at real-time or near real-time supervision level (e.g., on a daily basis). It aims at detecting obvious faults by considering the physical range of the measured parameter. For example, the selection of an appropriate sensor can already filter out out-of-range measurements. Data loss can be easily identified through relatively simple rules, provided that the acquisition strategy has a regular data acquisition frequency. Moreover, measurement blocking or saturation can be identified based on measurement stability over a given period and measurement accuracy. A sudden variation is detected by evaluating the gradient between two measurements. These calculation rules can be directly implemented in supervisory software to automate this validation [17].

However, the pre-validation does not identify all potential defects. Thus, it is generally supplemented by manual validation performed by an operator, whose goal is to assess the overall plausibility of the obtained results. The operator examines a series of data involving multiple variables to understand the dynamics of phenomena and the context of each measurement. The pre-validation and manual validation operations are time-consuming, often requiring the involvement of a dedicated team or the use of an external service provider.

Given the high number of equipped points and the high data acquisition frequency in the context of wastewater networks [18], these manual approaches quickly become tedious due to the time required for repetitive work<sup>1</sup>. Moreover, fully eliminating subjectivity from the validation process and the inherent human error can be challenging. It is therefore important to develop new approaches that will facilitate the validation tasks carried out by the various stakeholders involved (operators, engineers, researchers, etc.), enabling them to use their expertise for more rewarding tasks. The aim is therefore to provide automatic or semi-automatic data validation tools.

### **Data validation in the era of artificial intelligence**

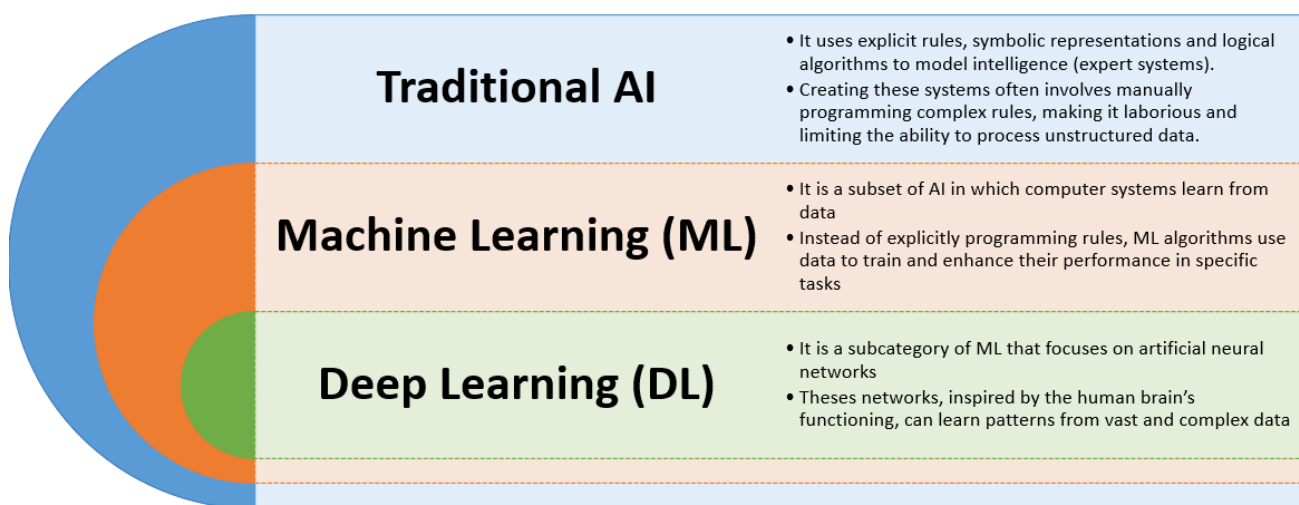
In the realm of Artificial Intelligence (AI), the issue of data validation is commonly referred to as "anomaly detection". AI has proven its efficacy in data validation across various disciplines. AI-driven algorithms are employed for data validation in areas like cybersecurity and medicine, where the volume of collected data is substantial, and the presence of anomalous data can

---

<sup>1</sup> By experience feedback, it takes around a month of work to an operator to validate the previous year's overflow monitoring data, issued from a wastewater system of around 10 000 EH

have significant implications. In such contexts, automated validation often fails short and manual validation would entail prohibitive time and cost. Being in a somewhat analogous operational validation context, AI emerges as a lever of action and a path to explore within the scope of this thesis.

According to the CNIL (Commission nationale de l'informatique et des libertés), AI is not a technology in its own right, but rather a scientific field in which tools can be included if they meet certain criteria. AI is a logical, automated process, generally based on algorithms, capable of performing well-defined tasks close to those of human reasoning [19]. In the realm of AI applications, it is common to distinguish three levels (see [Figure 0-4](#)).



**Figure 0-4: Levels of artificial intelligence**

In the field of hydrology, there are several instances of data validation based on AI, especially in assessing the quality of rivers and drinking water [20] - [21]. However, to the best of our knowledge, these tools have not yet been evaluated for wastewater data at the urban network scale. This thesis, as its title suggests, aims to explore AI tools, with an emphasis on both ML and DL, to improve the data validation process in the context of wastewater networks. The objective is to simplify the task of various stakeholders who utilize this data.

⇒ *Today, the main approaches used to validate data from wastewater networks combine automatic validation based on statistics with manual validation carried out by an operator at a later time. The former remains superficial in view of the range of potential faults. The latter is reliable (though not infallible), but costly and time-consuming. AI tools aim to streamline the process, making it more efficient for stakeholders. Although AI-driven data validation is common in hydrology, it's yet to be fully evaluated in urban wastewater networks.*

## Objectives and thesis outline

The motivation for this thesis stems from the fact that the installation of sensors in wastewater networks has become common practice to ensure rigorous management and direct interpretation of ongoing phenomena. However, due to the substantial volumes of data generated by these sensors, collected at variable time intervals, and acquired in harsh environments, the resulting data often suffer from inaccuracies. In this context, data validation becomes indispensable. The current landscape of deployed tools often highlights laborious approaches based on statistical rules, supplemented by domain-specific analysis.

The contribution of this research lies in its focus on automated validation of measurement data from a wastewater network. This approach explores AI techniques to detect sensor failures. The primary objective is to guide decision-making and streamline the validation process, making it more efficient. It is important to emphasize that this validation is exclusively carried out post data collection, a decision justified by the prevalent data processing domains (see [Figure 0-1](#)), which often operate with a time delay (e.g., regulatory document edition, phenomenological analysis, research and development).

To implement and evaluate these tools, we used pollution data from the wastewater network of Saint Malo Agglomeration (SMA). It is noteworthy that this type of data is not commonly encountered in wastewater networks. Typically, network managers focus on hydrometric measurements (such as water level, velocity, and flow rate). Those who assess pollutant flows rely on spot sampling or proportional sampling to the volume discharged. Nevertheless, the use of continuous pollution monitoring is starting to gain ground, supported by research needs where continuous pollution monitoring is frequently used. Over the past three decades, numerous sensors, including turbidimeters, have been deployed and tested to gain insights into pollution flows [11]. Hence, the choice of this data is motivated by two fundamental reasons. Firstly, from an operational perspective, the data from SMA represent an easily accessible database, for which a certain degree of "truthfulness" is guaranteed, rendering them particularly suitable for this study. Additionally, from a scientific standpoint, pollution data proves to be among the most challenging to validate due to their rapid and fluctuating dynamics, particularly for turbidimeters. This study serves as a proof of concept with the potential to extend to various measurements within wastewater networks.

The entirety of the results obtained during this research is presented in this PhD manuscript.

### **Part I: Literature Review**

This first section will provide an overview of the current state of research on data validation in wastewater networks by covering the existing methodologies and approaches as well as the associated challenges. In the first chapter, we will examine the data of urban wastewater networks, highlighting the specific characteristics of these data and defects that may occur in these systems. The second chapter will delve into existing data validation approaches, focusing on data quality checks and different validation methods, whether manual, statistics or based on hydraulic modelling. Finally, the third chapter will introduce a framework for data validation enhanced by artificial intelligence, exploring existing anomaly detection models using AI in urban hydrology.

### **Part II: Material and Methods**

The material and methods section is the essential foundation for understanding the implementation of our model evaluation. In chapter 4, we present the database used for the development of the various tests and for the evaluation of our models, namely turbidity data from the SMA wastewater system. We detail the process of data collection and acquisition, before looking at their statistical analysis and understanding their dynamics. Subsequently, we will develop an expert and manual data validation process that will be the baseline against which the results of the AI models will be compared. We will also highlight one of the key problems of this process namely human subjectivity, which drives us to organize a validation pool. Chapter 5 constitutes our benchmark of the models to be evaluated by examining their principles and how they are adapted to our case study, their architectures, and how each can be used for anomaly detection and data validation, by providing an overview of the various tests that will be conducted thereafter. Finally, in chapter 6, we dive into the metrics that will evaluate the performance of AI models vis-à-vis the reference, as well as the metrics that will evaluate the subjectivity among the different experts in our validation pool.

### **Part III: Results and Analysis**

The objective of this section is to present the results of the various tests, their implementation conditions, and a critical analysis of the results. The seventh chapter is an in-depth exploration of the concordance between the annotators via different metrics in order to evaluate their agreement and estimate the bias related to their disagreement. This process makes it possible to decide on the relevance of the manual approach of validation: does it have solid foundations or is it a random and/ or trivial validation ? Chapters 8, 9 and 10 provide the results of the

different tests using the three models of our benchmark, namely Matrix Profile, ResNet and Autoencoder respectively. We begin with an exploration of the sensitivity of results to input data, addressing aspects such as data preprocessing. Then, we focus on the fine-tuning of the hyperparameters of each model. The analysis and diagnosis of the results make it possible to identify the strengths and weaknesses of each model, leaving room for improvement strategies whose purpose is to explore different approaches that may improve the results. Finally, we evaluate generalization to other sites from the same agglomeration before investigating multivariate approaches.

The final chapter of this section serves as an extension of the research scope, exploring data from a different source, such as conductivity data from the wastewater network of Saint Malo Agglomeration and the water level data from the wastewater network of Wallonia, Belgium. It will detail the acquisition of this data, the validation methodologies applied, and the unique challenges associated with this data type. This chapter provides a comparative dimension that allows us to conclude on the potential of the developed tools with regards to a new type of data.

### **Conclusion and Perspectives**

The final chapter synthesizes the findings and insights from the previous chapters. It will draw conclusions based on the results obtained, assessing the research objectives, and addressing the research questions. Furthermore, this chapter will outline potential prospects for future research, highlighting areas where the study can be expanded or refined. It will underscore the scientific and practical implications of the research and its contributions to the field of wastewater network management and data validation.

### **Context of the thesis**

This PhD thesis was conducted within the company 3D EAU, supervised by the fluid mechanics laboratory ENGEES-ICUBE in Strasbourg. This laboratory uniquely brings together two scientific communities situated at the intersection of the digital and physical worlds.

3D EAU, as an engineering consulting firm, applies hydraulic modeling tailored to the specific context of each project. This includes using these models during the project's design phase to validate and optimize the proposed structure, in the diagnostic phase to analyze and enhance existing structures, and during the instrumentation phase to determine the number, position, and type of sensors to meet regulatory requirements.

With an ongoing commitment to innovation, supported by a strong collaboration with the ICUBE laboratory, 3D EAU has supervised four theses in various disciplines, all related to hydraulics

and the field of water. The acquisition of 3D EAU by Groupe Alcom in November 2022 has strengthened the connection between the environment and artificial intelligence, placing this thesis at the heart of the group's strategic development by establishing bridges between different entities.

### Scientific contributions

During this PhD, several contributions to the research field have been achieved. Two articles as first author and three conferences proceeding have been published.

#### Conferences

- Congrès ASTEE – Dunkerque 2022 – *Comment l'intelligence artificielle peut simplifier le processus de validation des données ?*
- Journées Information Eaux 2022 - *Développement de méthodologies et d'outils de validation de données – Application aux données d'autosurveillance et de diagnostic permanent des réseaux d'assainissement.*
- Journées Doctorales en Hydrologie Urbaine 2022 - *Utilisation de l'intelligence artificielle pour la détection d'anomalies dans les mesures de pollution.*

#### Scientific articles

- Techniques Science et Méthodes 2022 - *Utilisation de l'intelligence artificielle pour la validation des mesures en continu de la pollution des eaux usées.* <https://doi.org/10.36904/tsm/202211039> (Prix des lecteurs de TSM 2022).
- Water science and technology 2023 - *Validation of wastewater data using artificial intelligence tools and the evaluation of their performance regarding annotator agreement.* <https://doi.org/10.2166/wst.2023.174>



# Part I

## Literature Review



## ***Introduction of Part I***

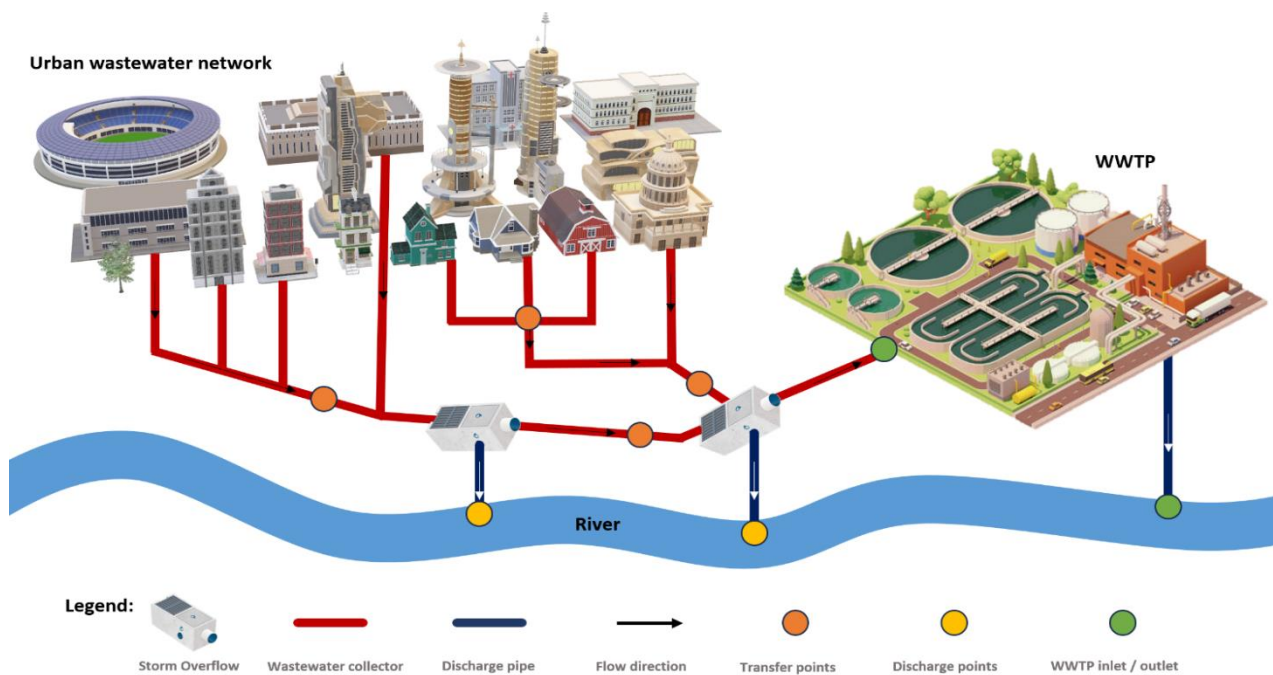
The objective of this section is to provide an in-depth overview of the current state of research on data validation in urban wastewater networks, the existing methodologies and the associated challenges by answering the following questions:

- **What specific characteristics of urban wastewater system data should be considered when validating data, and what types of defects may occur in these systems?**
- **What are the current approaches to validate wastewater data, starting from automatic pre-validation checks to validation methods, whether manual, statistical or based on hydraulic modelling?**
- **How can artificial intelligence be integrated into a data validation framework, and what are the existing AI-driven anomaly detection models in urban hydrology?**

# Chapter 1. Wastewater data in urban networks

In response to a diverse range of regulatory, technical, and scientific requirements (see [Figure 0-1](#)), it is becoming increasingly common to intensify the deployment of sensors within wastewater networks. This intensification is referred to as a **measurement network**, which consists of a series of sensors, their configuration being tailored to reflect the structure of the wastewater network. So, sensors are placed at strategic locations (see [Figure 1-1](#)). A measuring point can aggregate data collected from one or more sensors. Several types of measuring point can be distinguished [22]:

- *Transfer points*: These points are designed to measure flows or concentrations of pollutants along the sewer network, transferred from upstream to downstream. Sensors are thus installed in transit pipes, pumping stations, storage basins, etc.
- *Discharge points*: These points, corresponding to storm overflows and overflows, are designed to evaluate flows discharged into the environment without treatment.
- *Treatment plant inlets and outlets*: These measuring points, at wastewater treatment plants (WWTPs), are used to assess water quantities and pollution levels in order to adjust treatment processes.



**Figure 1-1: Outline of urban wastewater network with the identification of different measurement points**

Measurements cover a wide range of parameters. In wastewater networks, excluding mechanical elements such as digital sensors, pump operations or inclinometer openings, measurable parameters can be categorized into two main groups: hydrometry (flow) and quality (pollution). The selection of parameters to be measured is closely tied to the chosen measurement point and the underlying issues. This includes considerations such as measurement conditions, costs and reliability. It should be noted that identical information can be obtained by measuring different parameters [23].

**Table 1: Examples of measurement probes in wastewater networks**

Category	Assessment methodology	Probes used
Hydrometry	Flow measurement	Magnetic flow meter ...
	Level to Flow conversion	Level probes (ultrasonic sensors, pressure sensors, ...)
	Level-Velocity-Flow conversion	Level probes + Velocity probes (doppler, profilometer, ...)
Quality	Tracing / Gauging	Tracer injection and monitoring
	Sampling campaign	Spot sampling (SS, CDO, BDO <sub>5</sub> )
	Correlations	Turbidimeter, conductivity meter

Among the aforementioned measurements (see [Table 1](#)), those that are obtained directly and continuously include water level, velocity, turbidity, and conductivity. However, once these sensors are in place, they generate a substantial volume of data. The key challenge is to integrate them into a harmonious and efficient data monitoring system.

### 1.1 Measurement network monitoring

Like any measurement system, the transition from sensor to data involves a measurement and transmission chain that must be carefully designed, considering the specific characteristics of the installation environment. These characteristics encompass the availability of electrical power, the network coverage, the geographical location of the site (urban, rural), as well as considerations related to investment and operational costs. This chain can be divided into three distinct phases: data acquisition, transmission, and supervision (see [Figure 1-2](#)). For the purposes of this study, the focus will be exclusively on the data acquisition and supervision phases, while guidelines and best practices for data transmission are already available in the specialized scientific literature [24] , [13].

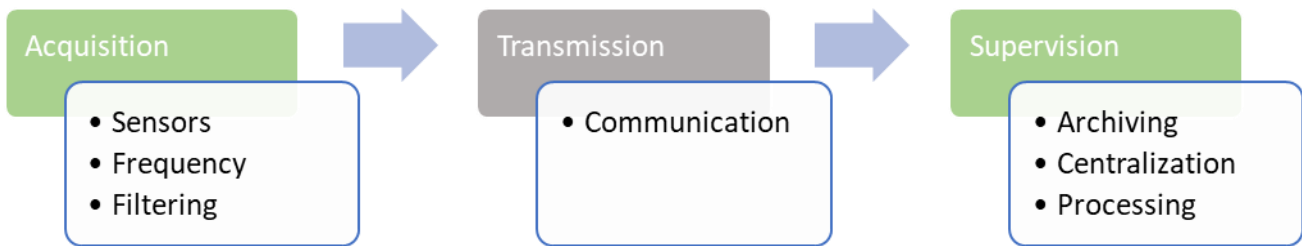


Figure 1-2: Measurement chain

### 1.1.1 Data acquisition

The initial stage of the measurement process, i.e., data acquisition, is of fundamental importance [25]. First and foremost, the appropriate choice of sensor is crucial to ensure accurate measurement of the phenomenon of interest. This selection is based on an analysis of the structure's configuration, leading to the identification of the appropriate sensor, the required measurement range, considering any dead zones, and the optimal location, aimed at reconciling the representativeness of the measurement and the accessibility of the device.

The second phase of the process involves configuring the sensor's frequency and acquisition strategy. This configuration is based on two major concepts, which are programmed by means of algorithms within the sensor itself:

- **Sampling frequency:** This involves interrogating sensor transmitters and temporarily storing one or more successive values before the final recording, which can be made at a lower frequency. Sampling frequency plays a crucial role in the system's ability to detect variations in measured values. Consequently, it must be adjusted according to the speed of the phenomenon under observation. For example, for rapid phenomena such as a discharge, a fine scanning frequency is required, whereas for slower phenomena, such as water level variation in a storage basin, a lower scanning frequency may be appropriate. Power consumption and battery capacity constraints must also be considered, particularly in the case of remote devices.
- **Recording (or transmission) frequency:** This frequency is constrained to prevent unnecessary overloading of data storage, transfer, and processing capacity. It is, by definition, equal to or lower than the sampling frequency. The recorded value is often pre-processed data, typically the average or median of the scanned values. Typical frequencies in wastewater management usually range from one minute to an hour. Higher frequencies are commonly associated with monitoring or analyzing specific phenomena, while lower frequencies are more suitable for the purposes of overall assessment or system dimensioning.

From an operational point of view, the recording frequency can be adapted according to the occurrence of a specific phenomenon. Let's consider a scenario involving a water level or velocity sensor in a discharge pipe. In the absence of rainfall events, this pipe is generally inactive, and the sensor records mainly zero values. Consequently, we don't need a detailed representation of the phenomenon. The recording of a few values to show that the sensor is indeed operational is sufficient. But, when a rainfall event occurs, the pipe is put under stress, and measurement becomes essential. Spills are often of short duration, which requires a fine recording frequency.

To reconcile these two constraints, a **non-constant recording frequency** may be adopted, which is based on a threshold value exceedance criterion. For example, the use of an overflow detector (digital sensor) in the inlet pipe can trigger a change in recording frequency as soon as the water level exceeds a predefined threshold. Typically, the recording frequency is changed from approximately 15 minutes to intervals of 1 or 5 minutes. This strategy spares data storage and processing, whereas a constant recording frequency makes the detection of missing values much easier

Therefore, understanding the **data acquisition strategy** is of crucial importance before beginning to process the information gathered. It is essential to know the origin of the data, i.e., how it was collected, as well as the representativeness of the measurement. This prior knowledge provides a foundation for correctly interpreting the data and drawing meaningful conclusions.

### 1.1.2 Data management (supervision)

Once measurements are carried out, they are transferred to the supervision level, creating a time chronicle, i.e., a continuous **temporal sequence of data**. In this context, it is essential to retain the time stamp of each recorded value, associating a precise time indication with each data point. This guarantees the chronological integrity of the information, which is essential for subsequent analysis and for understanding the evolution of the measured phenomena over time. In addition, it is important to maintain timestamps of missing data, as in many cases data are sampled asynchronously, meaning that measurements are not taken at regular intervals, and their frequency can vary according to circumstances. These data constitute a dynamic database, which must be properly archived for future use. It is therefore essential to associate each time sequence of data with its static characteristics, commonly referred to as metadata. This **metadata** includes information such as the location of the measurement, the quantity measured and its unit, as well as the data acquisition strategy. This association is achieved by assigning a unique identification code to each time sequence. The aim is to ensure data traceability.

Subsequently, data from all sensors is aggregated into a centralized system, enabling large-scale analysis and comparison of collected data. For this centralization and data processing, supervision software, such as Topkapi or Eve'M for wastewater management domain, can be used. These programs facilitate the management, storage, and visualization of data for macro-analysis. In addition, it is worth noting that information on on-site interventions can also be valuable, and their traceability is ensured by maintaining a logbook. The logbook is used to record actions taken, adjustments and maintenance operations, thus helping to comment on the collected data. This data is then processed according to specific objectives and needs. This is generally the stage at which **data validation** takes place.

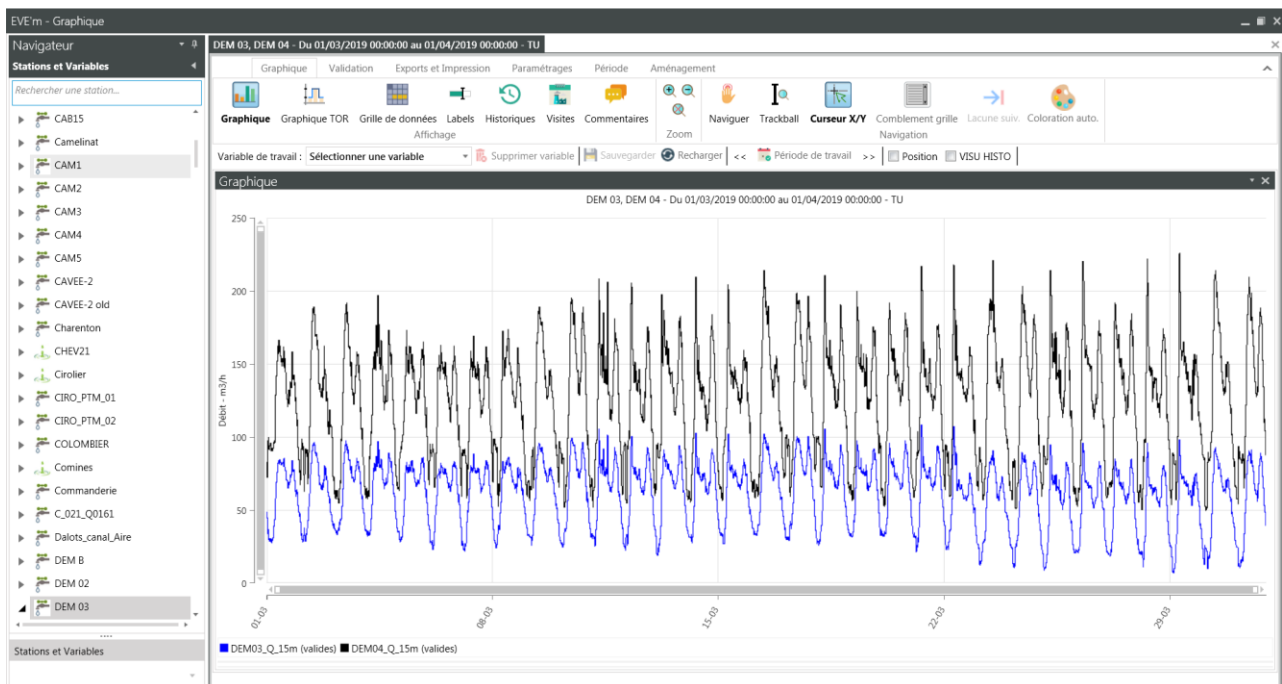


Figure 1-3: Example of supervision software: Eve'M - © SIGT

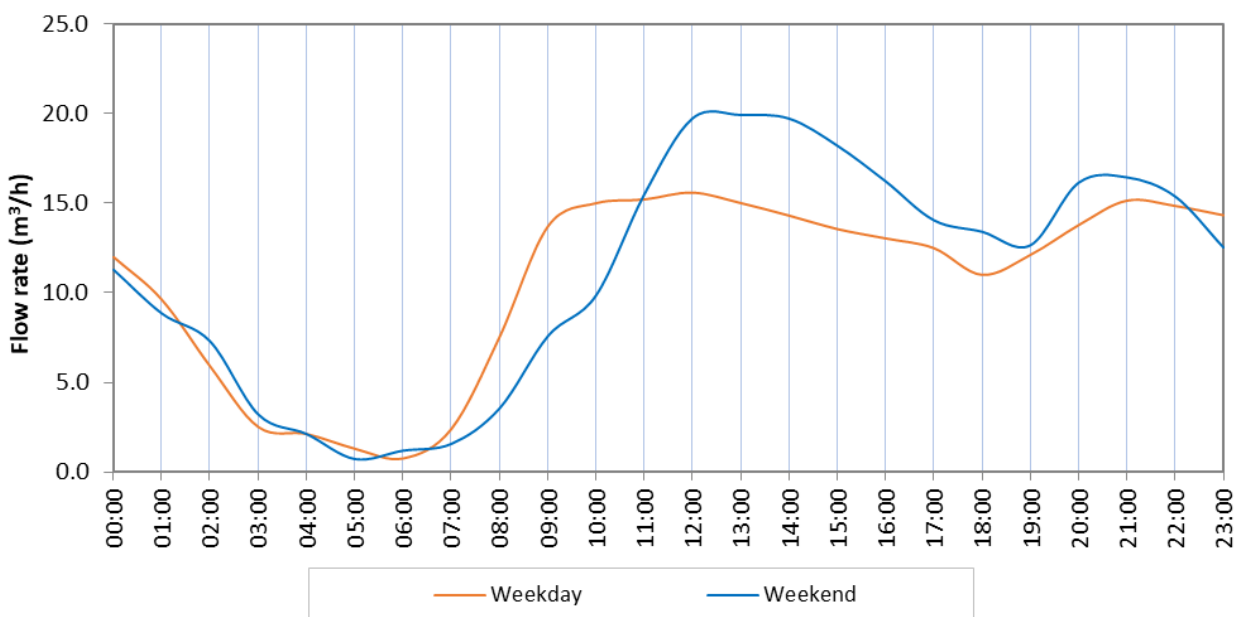
### 1.2 Special features of data in wastewater networks

Compared to conventional temporal data, wastewater data has distinct characteristics. The pattern of these data can vary considerably due to a number of factors, including weather conditions, human behavior, and the intrinsic characteristics of the wastewater system itself. In fact, the sewer network can be supplied by two main means: residential / industrial wastewater pipes and rainwater evacuation drains. In combined sewer networks, the two flows are mixed, and it is therefore imperative to consider the interactions between wastewater and stormwater dynamics, in terms of data structure.

The first significant characteristic of these data is their **seasonal nature**, closely linked to the succession of periods without precipitation (dry weather) and periods with rainy events (rainy weather), as well as evapotranspiration and the conditioning of effective rainfall.

In **dry weather**, combined sewer systems mainly used for wastewater collection present a distinctive daily pattern. It is characterized by significant variations throughout the day (see **Figure 1-4**). A daily flow peak occurs in the morning and early evening, corresponding to periods of high domestic activity, when water consumption and wastewater generation are high. It should be noted that flow peaks and this temporal profile during the weekdays differ from those on weekends and holidays due to variations in residents' habits, such as waking up late, for example. As a result, the structure of dry weather wastewater data shows a dual seasonality, i.e., a day/night variation and a variation according to weekdays and holidays.

During the overnight, a drop in flow rate is observed due to the substantial reduction in domestic activity. It should be noted, however, that this night-time drop is not null (see **Figure 1-4**). In fact, this is related to two main phenomena, namely the transit time in the wastewater system and the inflow of parasitic clear water due to imperfectly sealed networks and non-compliant connections [26]. For the former, transfer times from upstream to downstream can be significant, extending over periods of 10 to 20 hours. As a result, daytime wastewater from distant areas is superimposed on nighttime wastewater from nearby areas, maintaining a continuous flow even during the night. On the other hand, inflow clear water is unpolluted water that is continuously present in wastewater systems. Its origin can be attributed to a variety of factors, such as water source intake, permanent groundwater drainage or drinking water leaks. This superimposition of the two phenomena must therefore be taken into account for a better understanding of hydraulic dynamics in wastewater systems.



**Figure 1-4: Example of a typical dry weather data pattern in a wastewater network**

When **rainy weather** occurs, combined sewer networks undergo important changes in their data structure. Two major phenomena are added to the dynamics observed in dry weather: runoff and drainage.

Runoff occurs when rain falls on the impermeable surfaces of urban areas, such as roads and roofs, and then flows towards the wastewater network via sewer drains. This runoff combines with domestic wastewater, resulting in a sudden increase in flow. This rise may be short-term, but can be very intense, creating considerable flow peaks in the data. It's also important to note that not all rainfall events are the same. Indeed, depending on factors such as rainfall intensity, the size of the watershed and the level of soil sealing, the impact on the wastewater system may differ. Rainfall events are generally categorized according to their return period, which is a statistical concept indicating the frequency with which a rainfall event of a certain intensity can occur [27].

Drainage occurs during and after a rainy period when urban surfaces begin to dry out. It is associated with a gradual decrease in flow through the network. The duration of this process can vary according to the intensity of precipitation, extending over several hours or even days.

Thus, during rainy periods, the data structure is characterized by rapidly rising flow peaks of varying amplitude, followed by a gradual decrease. Separating the components specific to wastewater and stormwater under these conditions can be complex but their overlapping should be considered in understanding the structure of the recorded data.

### 1.3 Defects in wastewater systems

As previously mentioned, the wastewater network represents a complex and challenging measurement environment, increasing the risk of failures and highlighting the imperative of implementing validation processes.

#### 1.3.1. Defining invalid data

Invalid data in wastewater networks are observations that do not accurately reflect the hydraulic behavior of the network at a given time. However, a distinction must be made between two categories of invalid data [28] - [29]:

- **Incorrect data**

Incorrect data generally refers to data points that do not reflect what is happening in the network. These are often described as errors or outliers and are generally attributable to failures in the measurement chain. The most common malfunctions causing anomalies are as follows:



- Loss of contact between the measuring sensor and the phenomenon being measured, due to problems such as fouling, clogging, or blockage.
- Mismatch between sensor output signal and measured variable, resulting from general sensor failure or parameterization errors. Problems such as a faulty sensor, electrical failure, acquisition electronics malfunctions, drifts, de-calibrations can cause these types of errors.
- Time registration errors, due to clock drifts, incorrect time setting, or recording problems. These problems can lead to a temporal shift in the data.

In other words, incorrect data in wastewater networks is often the result of various hardware and sensor failures.

- **Non-representative data**

Non-representative data, as opposed to incorrect data, refers to observations that are different or unprecedented. Unlike anomalies, these unusual data are not necessarily wrong. It is possible to obtain accurate measurements, but these do not adequately reflect the phenomenon of interest due to disturbing events that mask them. Non-representative data can result from various situations, including:

- Sensor maintenance, involving operations such as on-site calibration or zero checking.
- Wastewater network maintenance, such as pipe cleaning.
- Changes in network configuration affecting wastewater flow, for example, when effluent is diverted for construction work.
- Special hydrological events, such as extreme rainfall or exceptional tides.
- Downstream influence when it has not been considered when setting up the instrumentation.

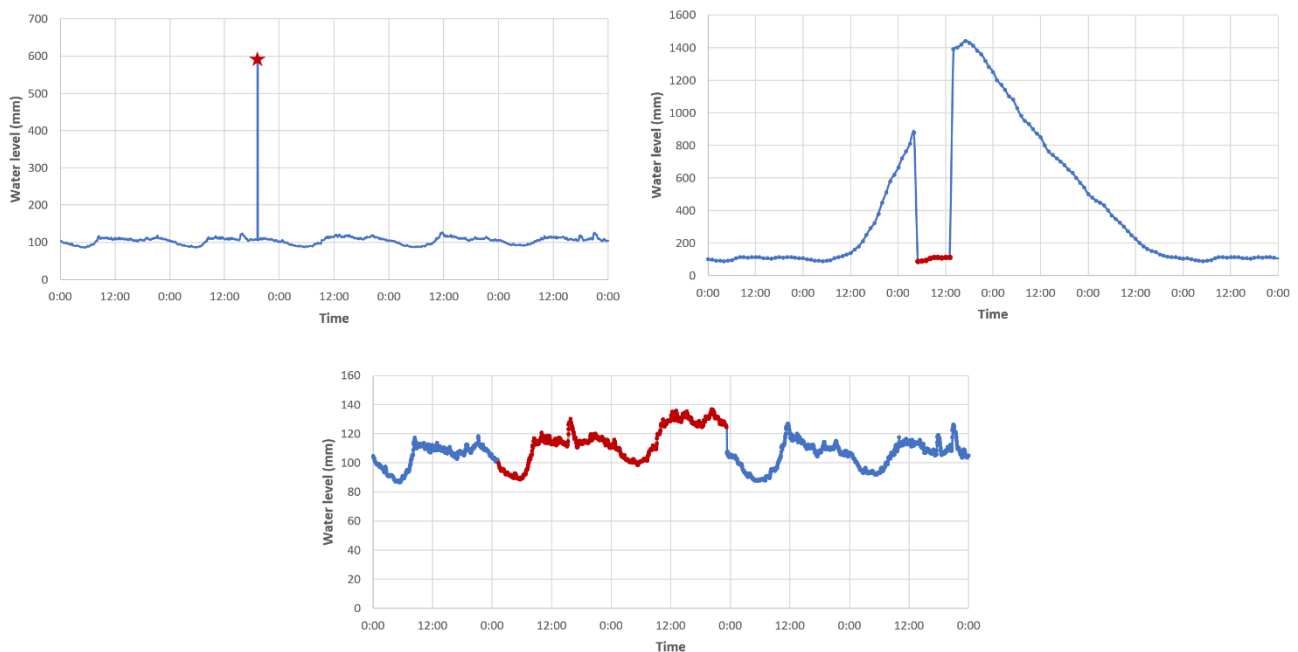
Although these events do not affect the accuracy of the data, they do not reflect the normal operation of the sewer network. Consequently, their interpretation requires an approach distinct from that applied to usual measurements. From an operational point of view, identifying these data requires access to exogenous data, such as the logbook with information on network management, meteorological conditions, the level of the receiving environment, and other relevant factors.

In the context of this thesis, **all forms of invalid data, including both incorrect and non-representative data, will be collectively referred to as "anomalies."** For admittedly different reasons, both categories of invalid data encompass observations that do not accurately represent the hydraulic behavior of the network, and it is therefore important to isolate them for operational considerations. The central focus of this work is on the

identification of these anomalies during the validation process. **Subsequent steps, such as the removal of invalid data or their potential replacement, fall outside the scope of this thesis.** The aim here is to perform rigorous data validation and pinpoint defects, while decisions on how to manage these anomalies are left to the discretion of the user and for future works.

### 1.3.2. Categorizing anomalies

Anomalies in non-sequential data are often defined as data instances that significantly deviate from the majority of instances. However, defining anomalies in time series data is challenging due to temporal correlations among observations [30]. Existing studies often adopt outlier definitions from non-sequential data. Specifically, they define outliers in sequential data through behavior analysis and categorize them into point, contextual, and collective outliers [29] - [31]. **Figure 1-5** illustrates these three types of outliers that often serve as a de-facto standard.



**Figure 1-5: Examples of different type of anomalies in red.**

**(Top left) Point outlier - (Top right) Contextual anomaly – (Below) Collective anomaly**

- *Point outliers* are individual instances that are anomalous with respect to the rest of the data. Let's imagine water level data collected in a 500 mm diameter pipe by a US probe. In the middle of the measurement chronicle, the water level suddenly rises to 600 mm, well above the normal range and measurement capacity of the sensor. This single, significantly higher measurement is an example of a point outlier.

- *Contextual outliers* are individual instances that are anomalous within a specific context. Contextual outliers typically have relatively larger/smaller values within their specific context but not globally. Some points may be normal in one context but detected as anomalies in another. Let's take a look at the water level data collected in a sewer system. During a rainfall event, the water level rises in line with the intensity of the rain and the response of the hydrological basin, then gradually falls in response to network drainage. However, during a rainy event, we can observe water level that drops sharply and can remain low for several hours, while still being of the same order of magnitude as the dry weather pattern. In this context, this drop is abnormal, as it does not correspond to the seasonal trend. However, if we consider this sequence during the dry season, it may appear normal. This is an example of a contextual outlier.

- *Collective outliers* are defined as collections of related data instances that are anomalous in relation to the entire dataset. Individual points within a collective outlier may not be anomalous by themselves, but their co-occurrence constitutes an outlier. Collective outliers are common in sequential data due to the often-strong dependencies among time points. There are cases where individual points are not anomalous, but a sequence of points is labeled as an anomaly. If wastewater level data in a sewer system show that every day for a week, the water level rises slightly but steadily at a constant rate, this may seem normal at the individual level, since each individual value falls within the usual measurement range. However, when we look at the week as a whole, we realize that the sequence of constant rises is not expected and does not correspond to habitual behavior. In this case, the sequence of constant rises over several days is an example of a collective outlier.

While the above categorization covers both individual instances and sequential instances, defining collective and contextual outliers can be complex due to context ambiguity [30]. For simplicity, **contextual and collective data will be collectively referred to as sequence anomalies** since they both involve outliers across multiple time points. Identifying the latter is often considered more challenging than point outliers and is extensively explored in the literature [32] - [33]. Having a priori knowledge of the type of anomaly in the data helps data analysts select the appropriate detection method. Some approaches that can detect point anomalies may fail to identify collective or contextual anomalies. The complexity of wastewater data lies in the fact that it can exhibit both point outliers and anomalous subsequences.

### 1.4 Focus on turbidity data

Wastewater effluent quality can be characterized by numerous parameters. Usually, this information is accessed via laboratory analysis, requiring on-site samples, which proves to be

a time-consuming, costly method, and not well suited to regular monitoring, particularly during periods of heavy rainfall. Such analyses provide only a limited view of the phenomena, due to their significant temporal variability. Hence, turbidity measurement proves to be the most practical technique for obtaining continuous, real-time information on effluent quality over long periods. Turbidity sensors installed at transit points, discharge points or at the inlet/outlet of a WWTP provide data on the particulate load, the main vector of pollution in wastewater systems, whether for raw effluent in dry or rainy weather. Continuous measurement of turbidity makes it possible to effectively monitor these dynamics. The main advantage of turbidity lies in its ability to be measured continuously, offering excellent temporal sampling, unlike traditional parameters such as suspended solids (SS) and chemical oxygen demand (COD), which require limited sampling [34].

- **Turbidity measurement**

Turbidity in effluent is mainly due to suspended solids (SS), which are particles larger than 0.45  $\mu\text{m}$ . In wastewater effluent, turbidity values are closely related to SS concentrations. Studies have shown that most turbidity in wastewater is due to particles in the 10 to 20  $\mu\text{m}$  range [35]. Technically, turbidimetry is based on measuring the transparency of a liquid, without requiring the use of reagents. A turbidimeter evaluates the effluent's ability to absorb or scatter light [36]. Two techniques commonly used in wastewater treatment are attenuation and diffusion, and the value measured depends on the technology employed, hence the use of distinct units such as Formazin Turbidity Units (FAU) and Nephelometric Turbidity Units (FNU). In the wastewater domain, the recommended measurement ranges are 0 to 2000 FAU for attenuation measurement, and 0 to 1000 FNU for diffusion measurement. In practice, turbidity values are generally between 50 and 1000 FAU or between 25 and 500 FNU.

- **Turbidimeter sensitivity**

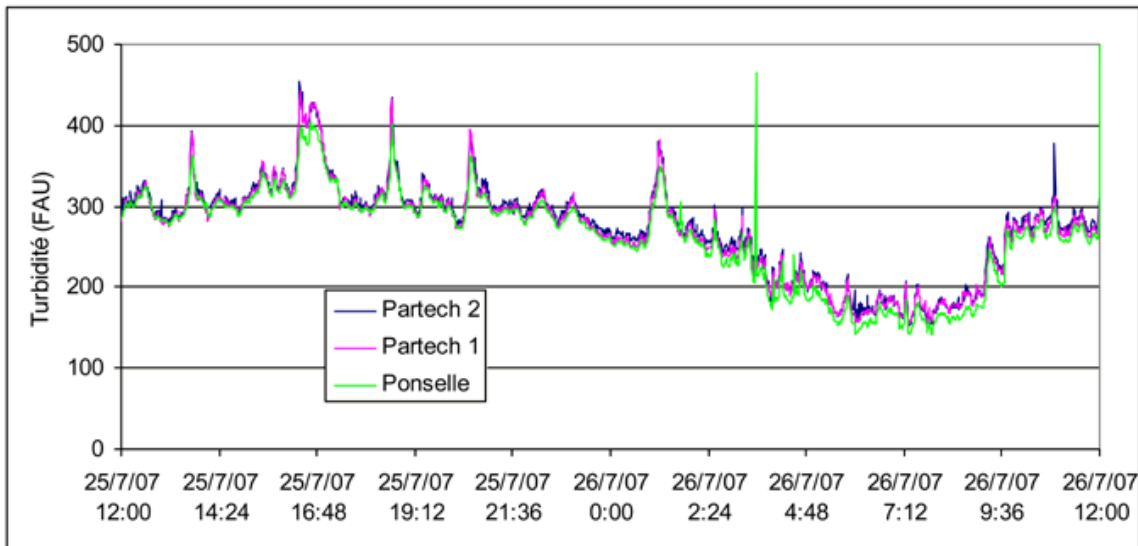
Nowadays, turbidimetry is proving to be a useful management tool for wastewater systems. However, it is essential to note that its use requires constant maintenance and control, along with budgetary resources. To guarantee reliable measurements, it is advisable to choose locations with adequate effluent mixing (without any bubbles), thus ensuring representative measurements, while remaining accessible to simplify maintenance operations. Turbidimeters, specifically designed for wastewater treatment, can be equipped with automatic cleaning systems, although they remain intrusive and require accurate installation and regular maintenance to prevent macrofouling. In general, maintenance should be carried out 1 to 4 times a month, while verification/calibration operations can take place every 6 months to one year.

- **Acquisition strategy**

Optimizing measurement reliability involves an appropriate acquisition strategy, including suitable signal processing. In many operational applications, it is not necessary to retain the full dynamic range of the signal, and it is often sufficient to work with average values, possibly weighted according to flow rates. To guarantee the reliability of these averages, real-time processing of data collected from redundant sensors is recommended. In the absence of redundancy, it is useful to record the standard deviation of the values that have contributed to the average, thus facilitating the identification of suspect recordings, which can then be invalidated. In this case, it is preferable to record data at a finer time step than is strictly necessary for the application, thus creating a safety margin. In the absence of real-time processing capabilities, it is advisable to record instantaneous values at a short time step, enabling representative averages to be calculated at a later date. For example, a recording with a time step of one minute is a minimum for calculating averages over periods of 5 to 10 minutes. Short time-step recordings are also suitable for research, enabling a detailed understanding of the dynamics of the phenomena under study [37].

- **Signal dynamics**

In addition to the seasonal and periodic characteristics of wastewater data, the turbidity signal, even in dry periods, exhibits rapid and significant fluctuations that reflect real variations in effluent quality. This finding is supported by recordings from three redundant sensors, sampled every 10 seconds (see [Figure 1-6](#)) [37]. It is important to note that the turbidity signal is highly variable and can show reproducible peaks on several redundant sensors. The challenge is to find a compromise that eliminates noise without eliminating the useful signal. This highlights the interest of having a sampling frequency that is fine enough - around one minute, or even less - to accurately capture these substantial variations.



**Figure 1-6: Example of turbidity variations, recorded using three sensors. © IFSTTAR - Duchesse Anne site, Nantes.**

What's more, disturbances - considered background noise – can occur and alter the signal. In addition to the random, zero-mean residuals that contribute to experimental uncertainty, asymmetrical noise is frequently observed, comprising high-amplitude positive peaks. These peaks, generally of short duration but sometimes longer, appear to be due to measurement artifacts caused by the temporary occultation of the measurement beam by particles or filasses. The frequency of these peaks varies over time and depends on the details of the sensor installation, reinforcing their artificial character. To guarantee reliable measurements, a validation process is needed to exclude these peaks and mitigate their potentially bias.

From an operational point of view, it is advisable to install redundant sensors to facilitate verification, at a reasonable additional cost in terms of investment and operation. Redundancy means installing two identical sensors in close proximity to ensure consistent measurement of the same effluents. This approach enhances measurement reliability in two ways. Firstly, when both sensors provide consistent values, there is a strong probability for them to describe a real phenomenon, even if the pattern may be somewhat unusual. Secondly, when sensors are not in agreement, at least one of them experiences a failure or displays an anomaly, usually the one with higher values. In this case, the other one may continue to provide relevant data. Redundancy should be considered where space constraints allow, provided that the sensors are similarly positioned in terms of transverse, longitudinal and depth to ensure consistent measurement. In addition, a conductivity meter is recommended to complement the information provided by turbidity, thus improving overall understanding of effluent quality.

### 1.5 Synthesis of Chapter 1

The measured parameters in wastewater systems can generally be grouped into two main categories: hydrometry (flow) and quality (pollution). The choice of parameters to be measured depends on the specific measurement point and the associated considerations and issues. The resulting measured data has several distinctive features compared to temporal data. First, the measurement strategy may be based on pre-processed data rather than point measurements, and the frequency of these data may vary and not be constant. Due to the multiplicity of sensors used, data synchronization becomes crucial to enable accurate comparisons. In addition, these data are subject to significant and multiple seasonality, reflecting variations linked to dry weather, rainy periods, day or night, weekdays or weekends. Rainfall events also vary greatly in terms of intensity, making network dynamics more complex with different classes.

Due to the installation environment, invalid data are highly probable within wastewater networks and comprise observations that fail to accurately represent the hydraulic behavior of the network at a specific time. These data can be further divided into two subtypes: incorrect data, which are outright errors, and non-representative data, which are unusual but not necessarily incorrect. It is therefore interesting to isolate and identify both types of invalid. Structurally, these anomalies can be classified as point, contextual, or collective outliers. The task of defining these anomalies can be complex due to temporal dependencies among observations and the ambiguity of the context.

Considering wastewater quality measurement, traditional laboratory analysis is costly and time-consuming, particularly during heavy rainfall. Turbidity measurement offers then a practical solution, providing continuous, real-time data at key points within the wastewater system. Ensuring accurate measurements, proper maintenance, and calibration of turbidimeters are essential. Redundancy is recommended for enhanced measurement reliability, along with the use of a conductivity meter, offering a better understanding of effluent quality. The acquisition strategy involves recording average values. In addition to the common features of wastewater data mentioned above, the turbidity signal exhibits rapid fluctuations, demanding a fine sampling frequency to capture variations accurately. Background noise can affect the signal, and a validation process is necessary to mitigate biases.

**Hence, what is being done today to validate data in wastewater networks?**

# Chapter 2. Exploring data validation pathways: State-of-art approaches

---

**“You can have all of the fancy tools, but if [your] data quality is not good, you're nowhere.”** -- Veda Bawo, responsible of Data, Risk & Control at Silicon Valley Bank

One of the most crucial tasks in data engineering is data validation and anomaly detection [38]. Anomaly detection methods are specific to the type of data being examined. For example, the algorithms used to detect anomalies in images differ from the approaches used for data streams [31]. This work focuses on methods for detecting anomalies in time series with an emphasis on urban hydrological data.

Wastewater networks are no exception to the data quality requirement, given the objectives linked to the use of measurement data (see [Chapter 1](#)). The high sampling frequency enabled by on-line water measurement networks leads to the collection of vast datasets covering several chemical-physical parameters. Due to the intrinsic properties of wastewater systems, datasets describing different parameters are often autocorrelated, not normally distributed and noisy [39]. Hence, it is important to find an anomaly detection approach adapted to the operating conditions of this fast-dynamic system. Since 1972 [40], the detection of anomalies in time series has long been a subject of interest. Consequently, many previously published studies have tackled this issue.

## 2.1 Data quality checklist: Pre-validation

Once data is acquired at the supervision level, regardless of the used software, it becomes imperative to undertake a pre-validation step for the identification of **trivial anomalies**. The early detection of these anomalies makes it possible to exclude them automatically and to direct the operator's efforts towards more subtle and complex anomalies. The pre-validation encompasses a range of standard checks, including but not limited to [41]:

- *Missing values* [42]: These are defined as data values that are not stored for a variable when they are supposed to be present according to the recording strategy. In order to save storage capacity a variable sampling interval can be applied. However, incorrectly programmed sampling intervals causes missing measurements. Loss of data can also result from limited storage in the sensor, problems with telemetry, loss of power supply or malfunctioning of equipment.



## Chapter 2. Exploring data validation pathways: state-of-art approaches

- *Out of range* [16]: This criterion combines two essential facets: the "physical range" and the "locally realistic range". This criterion establishes the boundaries within which valid measurements should fall, considering both the specific capabilities of the sensor and the characteristics of the measurement site. The physical range encapsulates values that, for physical reasons, cannot be exceeded by the sensor, typically aligned with the sensor's measuring capacity or the physical conditions of the environment. For example, a water level sensor with a measuring range of 0–2 meters cannot produce values that fall outside this prescribed interval. Water velocity in sewers cannot reasonably reach a value of 10 m/s, even if the measuring range of a sensor is able to exceed this value. Meanwhile, the locally realistic range narrows down this spectrum to encompass values commonly observed at the specific measurement site, shaped through available information and historical data. Any measurement falling outside this collective range is marked as a potential anomaly, necessitating further review and operator intervention.
- *Saturation / Blocking* [43]: This anomaly occurs when the sensor reaches its operating limits. It usually occurs when the measured signal exceeds the sensor's measuring range, pushing it to its upper or lower limit. Thus, saturation is closely related to the out-of-range anomaly. Similarly, a lock on a constant value may occur if the sensor is defective or exposed to extreme conditions that keep it at a single reading level.
- *Important gradient* [16]: Detecting sudden or irregular shifts in data values and unusual gradients due to physical processes and local conditions is crucial. These changes often result from sensor faults or non-representative phenomena, generating pronounced gradients. While simple threshold-based methods were initially explored, more advanced techniques combining filtering algorithms [22] were developed. Applying filters like moving averages smooths high gradients, and the difference between the original and filtered signals indicates abrupt changes. Customizing threshold values and window widths for specific measurements is crucial, involving iterative adjustments based on background noise, measurement uncertainties, and typical values.

In cases where a single site is equipped with two redundant sensors (e.g., two turbidimeters), a comparative analysis of their measurements and signal behavior can be conducted to identify unusual patterns or anomalies. When the difference between the two signals surpasses a predetermined limit, the associated data is questionable, necessitating further manual examination by an operator.

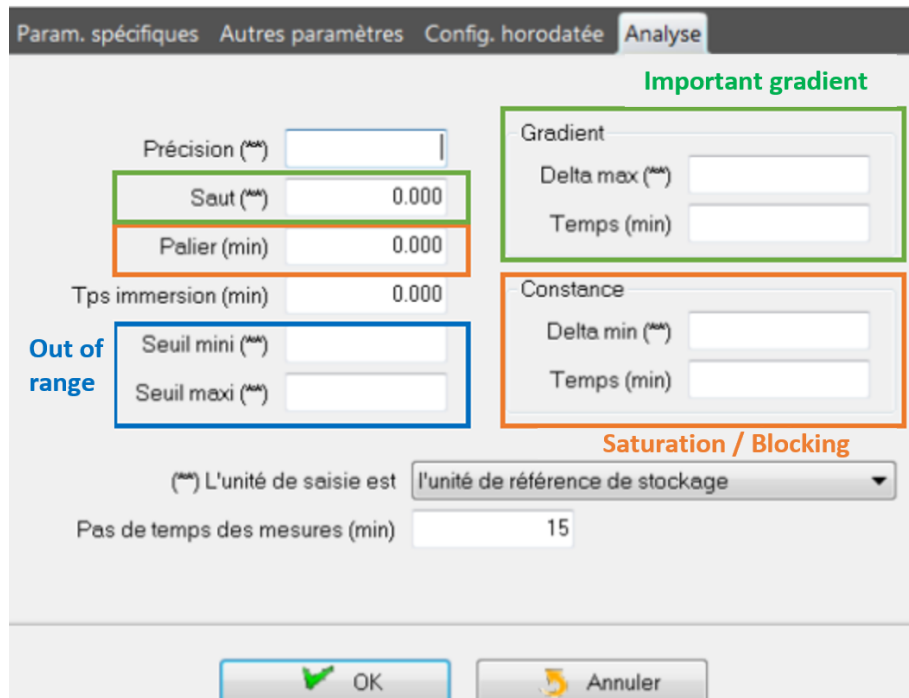


Figure 2-1: Example of pre-validation software - © Eve'M

During this pre-validation phase, basic statistical analysis techniques and/or advanced algorithms can be employed to identify conspicuous irregularities, including outliers (out-of-range), temporary data gaps (missing data), and significant deviations from anticipated theoretical patterns (important or no gradient). While these tests are usually already implemented in monitoring tools (see Figure 2-1), they represent the bare minimum before any operational considerations. The spectrum of possible defects is much broader, encompassing elements such as bias, drift, accuracy degradation, and other factors [44]. The complexity of these defects makes them more demanding and difficult to identify by conventional means. Hence, their detection requires more sophisticated and technically advanced approaches [45].

## 2.2 Validation phase

The pre-validation focuses on the analysis of the coherence of the individual signals from each sensor. It can be, then, considered as a local evaluation of the measurements provided by a particular sensor, and can be performed in real time. During this phase, the main objective is to identify obvious and immediate anomalies. It is therefore important to complete it with a validation phase, which instead adopts a more global and multivariate perspective, and is better performed offline. This approach makes it possible to detect inconsistencies or unusual functioning that may not be apparent during the individual analysis of the signals of each sensor or that could not be decided during the pre-validation phase. In short, pre-validation

focuses on obvious local errors, while validation focuses on more subtle anomalies at a global scale to ensure the quality and reliability of the data collected.

### 2.2.1. Manual Approach

The data validation phase usually requires the intervention of a competent operator, with a thorough mastery of the data in question and sufficient expertise to conduct an informed analysis and interpretation. We will refer to this operator by the term "**expert**".

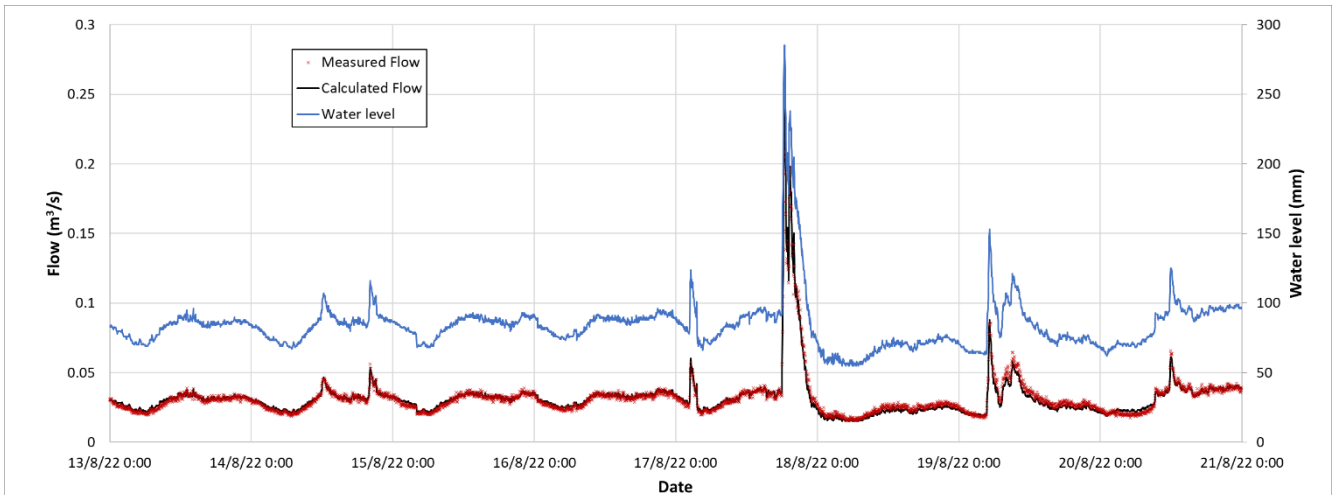
\*\*\*\*\*

*A little sociology: What is an expert?*

*Expertise consists in the production of action-oriented specialized knowledge, in a technical or professional setting. Recognized among his peers, the expert draws his competence both from the mastery of specific knowledge and his own experience [46].*

\*\*\*\*\*

The expert must have the necessary acuity to extract crucial information from data tables and charts presenting the complete chronicle. The ability to detect trends, breaks, seasonal patterns, or other important characteristics is essential for this task. The operator, through his analytical skills, examines the graphs to discern nuances and correlations. The expert can greatly benefit from the correlated nature of wastewater networks by using analytical redundancy in his validation process [47]. These correlated variable pairs can encompass either directly measured values, such as the water level and flow velocity within a specific sewer, or they can involve one measured value paired with a calculated or simulated value (see [Section 2.2.3](#)). An example of the latter would be the measured flow (using a flowmeter) paired with the flow estimated from water depth measurements (see [Figure 2-2](#)). It should be emphasized that throughout these comparative assessments, proper consideration must be given to the inherent measurement uncertainties associated with all relevant data [16].



**Figure 2-2: Example of graphical macro-analysis operated by an expert in order to validate flow measurement.**

Manual validation is a first step into data interpretation and conversion into information. However, while essential to ensure data quality, manual validation has some limitations that should be considered:

- *Need for expertise:* Manual validation requires considerable expertise in the subject area, which may limit the availability of qualified operators.
- *Time and resource cost:* Manual validation is often a time-consuming and laborious task. It can require considerable resources, especially when large amounts of data need to be analyzed.
- *Subjectivity:* Data interpretation may vary from operator to operator, introducing a degree of subjectivity. This can make it difficult to achieve consistency in data validation.
- *Possibility of human errors:* Operators may make errors during manual validation, such as typing errors, omissions, or misinterpretation, which may compromise the reliability of the results.
- *Limited detection of complex anomalies:* Subtle or complex anomalies can escape manual detection, especially when data is multidimensional or interactions between variables are complex.
- *Volume limitation:* Manual validation is feasible for relatively small amounts of data but becomes impractical when large data sets are involved.

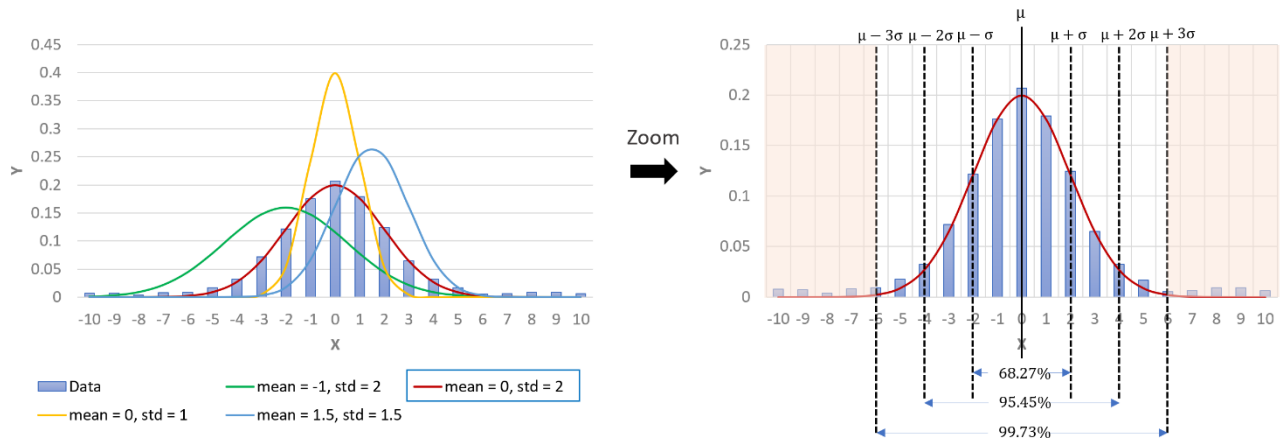
To overcome some of these limitations, automated approaches, such as the use of anomaly detection and validation models, are increasingly being adopted to complement or replace manual validation in some applications. They all rely on some kind of modelling (actually experts also rely on models, albeit implicit ones) and therefore the models to be deployed range from purely statistical models to complete hydrological and hydraulic models [48].

### 2.2.2. Statistical Tools

A preliminary overview of the statistical methods used in time series and signal processing has revealed that there are numerous approaches to data validation and anomaly detection within data series. The fundamental principle governing any statistical anomaly detection method is encapsulated by the statement: "An anomaly is an observation suspected to be either partially or entirely irrelevant, as it does not conform to the underlying stochastic model" [49]. Statistical techniques entail the fitting of a statistical model, typically representing normal behavior, to the available dataset. Subsequently, a statistical inference test is employed to assess whether an unseen instance is consistent with this model. Instances exhibiting a low probability of being generated by the learned model, as determined by the applied statistical test statistic, are identified, and classified as anomalies [50].

The first technique assumes that data follow or can be transformed into a **Gaussian distribution** and estimates model parameters using maximum likelihood estimates (MLE) [51]. The MLE is a statistical method used to determine the parameters of a probabilistic model, such as the mean and standard deviation in the case of a Gaussian distribution [52]. The aim of MLE is to find the parameter values that make the observed data most probable under the specified model. This method is based on the idea that normal data follow a specific pattern, and any observation that deviates significantly from this pattern is likely to be an anomaly. To assess whether a data item is an anomaly, the distance between it and the estimated model is calculated as an anomaly score. A threshold is then applied to these scores to determine which data instances are considered anomalies. Within this category, different techniques use various methods to calculate the distance to the mean and define the threshold. A common approach is to declare as anomalies all data lying more than  $3\sigma$  from the distribution mean  $\mu$ , where  $\sigma$  represents the standard deviation of the Gaussian distribution (see [Figure 2-3](#)). This range,  $\mu \pm 3\sigma$ , encompasses 99.7% of the data [53]. It should be noted that more complex methods, based on advanced statistical tests, have also been employed for anomaly detection, as discussed in references such as [54], [55], [56].

## Chapter 2. Exploring data validation pathways: state-of-art approaches



**Figure 2-3: Illustration of the principle of statistical anomaly detection on synthetic data. (Left): the MLE principle, with selected parameters framed in blue. (Right): anomaly identification using the 3-sigma rule (orange zone).**

Another frequently used statistical method for detecting anomalies in time series is based on the use of **regression models** [51]. The basic principle of the regression model-based anomaly detection techniques consists of two steps. In the first step, a regression model is fitted to the data. In the second step, for each test instance, the residual of that instance is used to evaluate the anomaly score. Here, we approach anomaly detection such as a forecasting task. Some of the well-researched regressive models suited to time-series are autoregressive models, and all their derivatives [31]. The autoregressive model stipulates that the output variable depends linearly on its own previous values and on a stochastic term. Consequently, this model takes the form of a stochastic difference equation (see [Equation 1](#)), where  $p$  is the preceding window length. The values of the coefficients  $a_1, \dots, a_p, c$  can be approximated by using the training data and solving the corresponding linear equations with least-squared regression. The error values  $\varepsilon_t$  are considered to be uncorrelated and have a constant mean of zero and constant variance  $\sigma$ . In this model, they are used to determine the anomaly score [57]. However, this method assumes that the data is stationary, which is not always the case in practice.

$$X_t = \sum_{i=1}^p a_i X_{t-1} + c + \varepsilon_t$$

**Equation 1: Autoregressive model equation**

The ARIMA (Autoregressive Integrated Moving Average) model is an extension of autoregression (AR) that was developed to handle non-stationary data, i.e., data whose statistical properties, such as mean and variance, vary with time [58], [59]. However, ARIMA models still assume that data follow a normal distribution. These models are based on the

principle that the current value of a time series depends both on its previous values and on past variations and errors [60] - [61]. After fitting the ARIMA model, anomalies are detected by evaluating the deviation of the predicted point to the observed one. Nevertheless, a major drawback of ARIMA models is that it requires an important data pre-processing and tuning step. It is essential to check data for stationarity and autocorrelation, as well as to determine optimal parameter values, often through iterations or a systematic search approach. Presence of anomalies in the training data can influence the regression parameters and hence the regression model might not produce accurate results [51].

In the field of urban hydrology, **Table 2** summarizes some references covering statistical models for data validation, ranging from simple to sophisticated univariate and multivariate tests. In these cases, operating conditions are controlled and in some cases, anomalies are introduced synthetically.

**Table 2: Some references for urban hydrology data validation using statistical approaches.**

Reference	Data	Statistical approach
[47]	Hydrological data (precipitation, flow, water level)	Filtering methods
[62]	Basin supply	Autoregressive models
[63]	Wastewater flow	Kalman Filter
[64]	Inflow in a reservoir	Filtering methods

If the assumptions regarding the underlying data distribution are valid, statistical techniques offer a justifiable approach for the detection of anomalies. The anomaly score produced by a statistical method is associated with a confidence interval, which can furnish supplementary information for decision-making regarding test instances. However, the principal limitation of statistical techniques lies in their reliance on the assumption that **data is generated from a specific distribution**, an assumption that frequently does not hold, especially in the context of high-dimensional real datasets [65]. In order to employ these methods securely, it is imperative to possess accurate and comprehensive information regarding the inherent characteristics of the system [66]. The application of pure statistical methodologies may encounter limitations, as not all system behaviors can be precisely encapsulated within statistical distributions [67]. The equations employed in these models often fail to capture the intricacies of time series data in practical scenarios. For instance, these models failed to detect anomalies in search response time data where the intrinsic structure has weekly periodicity but also daily periodicity, holidays and other factors [68]. The presence of unanticipated disturbances within the wastewater system renders the stochastic nature of

water quality parameters considerably uncertain, making **it challenging to establish a singular model capable of fully characterizing system behavior** [69]. In such cases, a combination of multiple detectors may be considered. Nonetheless, in the realm of water quality monitoring systems, determining the optimal number of potential models for system behavior becomes a complex undertaking. Moreover, the straightforward ensemble of these statistical detectors, such as majority voting [70] or normalization [71], proves to be ineffective according to [72].

When the statistical assumption can be reasonably justified, several hypothesis test statistics are available for anomaly detection, but **the selection of the most suitable statistic is often a nontrivial task** [73]. Consequently, individuals working with time series data must possess advanced statistical qualifications and make diligent efforts to select an appropriate algorithm and fine-tune hyperparameters for each specific time series.

### 2.2.3. Hydraulic modelling

To describe the complex dynamics in wastewater systems, a major effort has been made to develop mathematical models. This effort has been greatly aided by the development of IT resources. The aim of these models is to establish relationships, which enable output variables to be calculated as a function of input variables, and which also may involve other exogenous parameters. **The validation of on-site measurements can be conducted by analyzing their consistency regarding the calculated output data based on the known input data.** There are generally two types of models: specific 3D models which are local and 1D network models which are more global.

- **3D numerical modeling**

In recent years, the use of 3D numerical modeling for hydraulic engineering studies has intensified, thanks to the continuous improvement in computational capabilities and the development of user interfaces to facilitate the use of software [74]. There is numerous hydraulic modeling software on the market that exploit the rules of computational fluid mechanics (CFD), such as Open Foam, Flow 3D, Ansys Fluent, etc. They are based on the approximate resolution of the Navier-Stokes differential equations using the finite-difference method, based on a system of regular meshes representing the actual geometry of the structure. The position of the free surface is approximated by the finite volume method. The software enables dynamic calculation of the time step, so as to meet the criteria of numerical stability and convergence on the pressure calculation, at any point of calculation of the model provided boundary conditions are known. Defining these boundary conditions is rather challenging, but can harness the output of larger scale models, such as 1D network models.



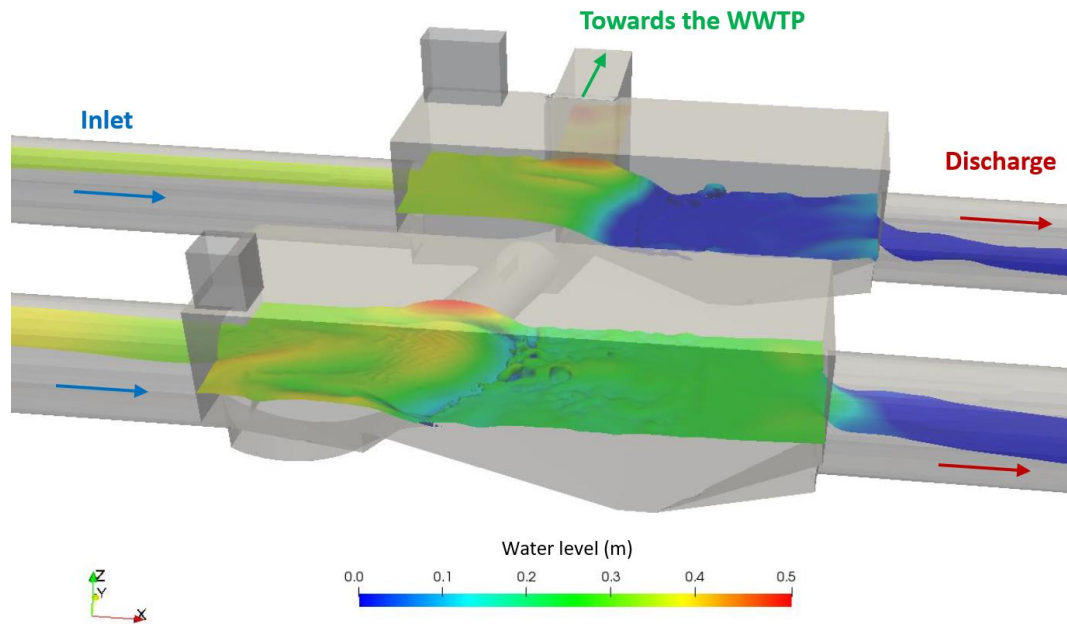
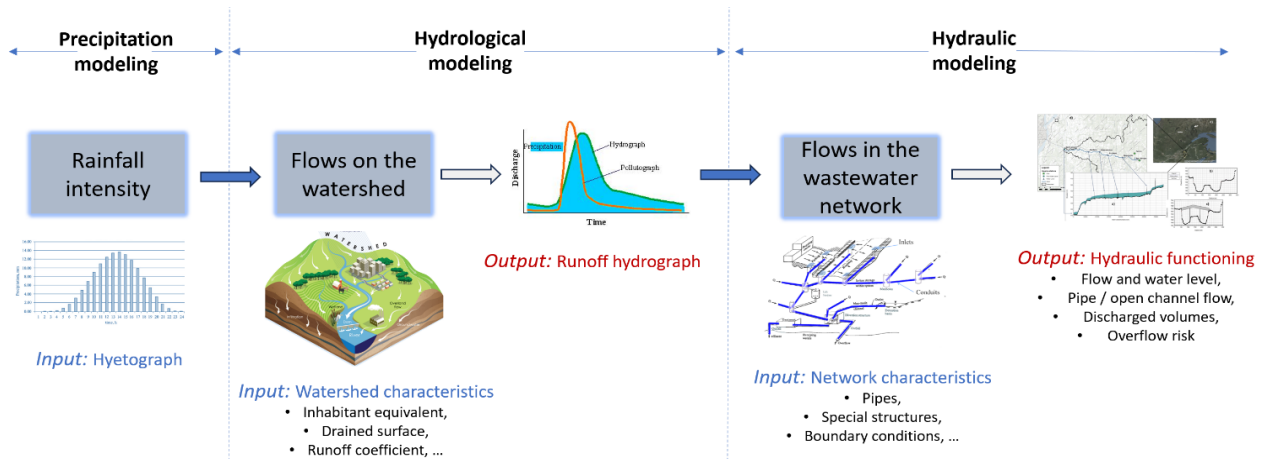


Figure 2-4: Example of 3D modelling of a double storm overflow

- **1D network modeling**

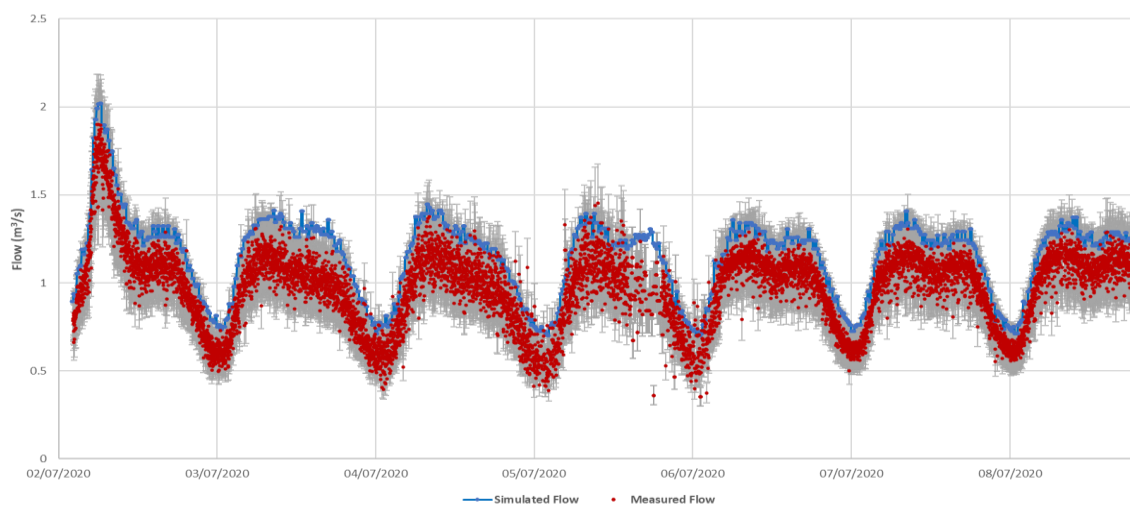
Unlike 3D models, which are generally limited to the scale of the individual structure and does not take into account variations in the hydraulic behavior of the network as a whole, the use of 1D hydraulic modeling of wastewater systems provides a global view of how the system works, using software such as Canoe, MikeUrban, etc. The aim of a 1D hydrodynamic model is to describe all the phenomena occurring within a given system, using equations from the fields of mechanics, hydraulics, chemistry, and biology. It thus offers the possibility of representing variations in water level and flow at any point of the network. Dynamic methods can be used to simulate the entire water cycle, from the moment of precipitation to the moment the water leaves the system. This dynamic modeling involves three key stages: firstly, precipitation modeling, which can take the form of project rainfall characterized by its total duration and duration of intensity, or the use of actual recorded rainfall data. Secondly, hydrological modeling, which involves describing the transformation of rainfall into flow according to the watershed characteristics such as drained surface and average daily wastewater profiles. Finally, the third stage, hydraulic modeling, focuses on simulating water flows within the network, while taking into account all its specific characteristics, including its mesh, branches, storm overflows, retention basins and specific boundary conditions [75].

## Chapter 2. Exploring data validation pathways: state-of-art approaches



**Figure 2-5: Stages of hydrodynamic modelling**

Using model-based approaches, one can generate simulated data for the operational conditions of the structure. By comparing these simulated data with the actual measured data, discrepancies and inconsistencies can be identified, which may indicate anomalies or problems that require further validation. For instance, consider a scenario where a measure is established to evaluate the flow rate of a wastewater pipe using a radar sensor. Simultaneously, a 3D modeling approach is employed to evaluate the flow rate under the same boundary conditions of water level and velocity (see [Figure 2-6](#)). The comparison between the flow rate measured on-site and the flow rate simulated through 3D modeling reveals an interesting level of agreement, with an error rate of less than 15%. This level of accuracy is akin to the uncertainty associated with a well-maintained Doppler sensor deployed under ideal conditions. It is relevant to note that, due to the costs associated with modelling approaches and the technical expertise required, their use for data validation purposes remains uncommon in practice.



**Figure 2-6: Data validation: 3D modelled flow Vs on-site measurement**

## Chapter 2. Exploring data validation pathways: state-of-art approaches

The use of these models, although providing methodical data validation perspectives, should not disregard their costs, especially since **one simulation is equivalent to one unique boundary condition**. Consequently, the validation of an extended chronicle with various conditions would require the multiplication of models. Moreover, the operation of these models requires a qualified operator to control the **entered boundary conditions**<sup>2</sup>.

Deploying these models involves considering a number of **technical problems**, since all modeling is subject to errors arising from both the structure of the model itself and the data, as well as their interactions during the modeling process. These problems include :

- *Errors linked to the structure of the model*, including theoretical limits and numerical approximations.
- *Data availability issues*, encompassing metrological and methodological challenges and their suitability for the specific needs of the model [76].
- *Model calibration and validation process*.

As far as hydraulic modeling is concerned, approaches such as the Barré de Saint Venant method and the Muskingum model are now commonplace in terms of development and testing. However, it should be noted that **pollution modeling presents a distinct challenge**. Pollution models remain largely underdeveloped.

---

<sup>2</sup> Generally, the use of these models leads to the subcontracting of specialized design offices.

### 2.3. Synthesis of Chapter 2

Data validation in the wastewater field involves two essential stages. First, **pre-validation** aims to detect trivial anomalies by means of **basic tests**. These anomalies include missing data, out-of-range values, and sensor saturation. However, the context of sewer networks can present more subtle and complex faults, such as biases, noisy data, and drifts. To address these faults, a second stage of **supervised validation** by an expert is necessary.

However, the **manual approach** carries inherent risks of subjectivity and human error, which become particularly worrying in the face of the massive volume of data generated by wastewater measurement networks. Consequently, the expert can opt for automated or semi-automated decision-support tools.

To validate data from these networks, **statistical approaches** can be used, offering real-time validation based on historical comparisons between predictions and measurements. Generally, these models focus on punctual anomalies (outliers). A distinction is made between simple statistical models, which are relatively ineffective at capturing the dynamics of wastewater data featuring non-stationary conditions, multiple seasonality patterns and partially autocorrelated time series. And on the other hand, there are more complex models, such as autoregressive models, which may be more suitable but are more difficult to implement due to their complexity and parameter sensitivity, which can lead to random / unstable performance.

On the other hand, **model-based approaches**, either local (3D structure modeling) or global (1D network modeling) can detect contextual and collective anomalies using exogenous data. However, their validation is carried out off-time, due to the time required for their deployment. Although a simulation corresponds to a specific boundary condition, the underlying model can be reused to reduce implementation times. However, the quality of the results depends heavily on the proper setting of the boundary conditions and the quality of the input data, such as the representativeness of the meteorological data used in hydraulic models.

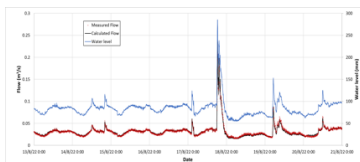
Thus, data validation in the context of wastewater networks remains a challenge due to large plethora of defects (punctual vs sequential, erroneous vs non-representative, etc.). Both statistical and model-based approaches have been extensively studied and analyzed in the literature, **our aim, through this research work, is therefore to use AI models and to test their ability to validate data issued from wastewater networks a posteriori**. The aim is to be able to model implicit structures identifiable in the data, in a more flexible way (with fewer assumptions) than statistical or hydrological models.

**Step 1: Pre-validation**

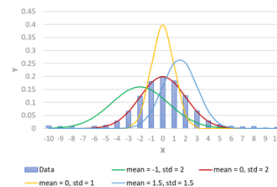
**Limits:** Does not identify all potential defects

**Step 2: Validation**

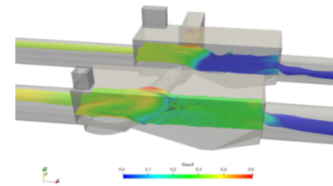
Manual Validation



Statistical Approach



Hydraulic Modelling



**Limits:**

- Tedious task subject to subjectivity and human error
- Requires significant business expertise
- Some approaches are costly
- Some approaches are inadequate for the nature of data in wastewater systems

**Figure 2-7: Overview of data validation approaches and their limits in wastewater networks**

# Chapter 3. Artificial Intelligence - Enhanced Data Validation Framework

---

**A number to start with: 496 010!**

This is the number of publications on AI in the world in 2021<sup>3</sup>. A value that doubled in 10 years, from 200,000 in 2010 to almost 500,000 in 2021 [77]. This volume is colossal, a veritable tide of information, which testifies to the place of AI in our world. It is therefore essential to start with an exploration of this field and a definition of its different concepts.

## 3.1 AI Vocabulary & History: A Primer

### 3.1.1. A timeline of artificial intelligence advancements

While it may be challenging to precisely pinpoint, the origins of artificial intelligence likely date back to 1950 when Alan Turing published his seminal article "Computing Machinery and Intelligence," outlining how to create intelligent machines and, in particular, how to test their intelligence [78]. The Turing Test, which is still considered a benchmark for identifying the intelligence of an artificial system, was born from this work: "if a human interacts with another human and a machine and is unable to distinguish the machine from the human, then the machine is considered intelligent". The term "**Artificial Intelligence**" was officially coined about six years later in 1956 when Marvin Minsky (co-founder of the MIT AI laboratory) and John McCarthy (a computer scientist at Stanford) organized the Dartmouth Summer Research Project [79]. This workshop brought together those who would later be recognized as the founding figures of AI. The Dartmouth Conference marked the start of nearly two decades of significant success in the field of AI. An early example is the well-known ELIZA computer program, created between 1964 and 1966 by Joseph Weizenbaum at MIT [80]. ELIZA was a natural language processing tool capable of simulating a conversation with a human and one of the first programs attempting to pass the Turing Test. As a result of these promising achievements, substantial funding was allocated to AI research, leading to an increasing number of projects [81]. In 1970, Marvin Minsky's interview with *Life Magazine* stated that "In from three to eight years we will have a machine with the general intelligence of an average human being". Regrettably, AI researchers had failed to appreciate the difficulty of the

---

<sup>3</sup> Total number of English-language and Chinese-language AI publications, including journal articles, conference papers, repositories, and patents

## Chapter 3. Artificial intelligence – enhanced data validation framework

problems they faced. Their immense hope had elevated expectations to an unattainable level, and when the anticipated outcomes did not materialize, AI funding vanished. This period marked the onset of the AI Winter [82]. In 1997, a major event reshaped the history of AI. IBM's Deep Blue chess-playing program managed to defeat the world champion, Gary Kasparov. For the first time, a machine had triumphed over a human. Deep Blue was reportedly capable of processing 200 million possible moves per second and determining the optimal next move by looking 20 moves ahead through a method called tree search [83].

Since 2010, the development of artificial intelligence has been accelerated by big data, which refers to massive number of datasets requiring sophisticated processing software tools. Artificial neural networks made a resurgence in the form of **Deep Learning** when, in 2015, AlphaGo, a program developed by Google, was able to beat the world champion in the board game Go<sup>4</sup> [84]. Deep learning algorithms have empowered computers to process and interpret complex data, leading to breakthroughs in fields like natural language processing and computer vision. The European Union has also recognized the importance of ethics in AI and established a 2024 deadline for companies to comply with new AI regulations [85]. In 2022, the release of ChatGPT, a large language model trained by OpenAI, marked a new milestone in the AI field [86].

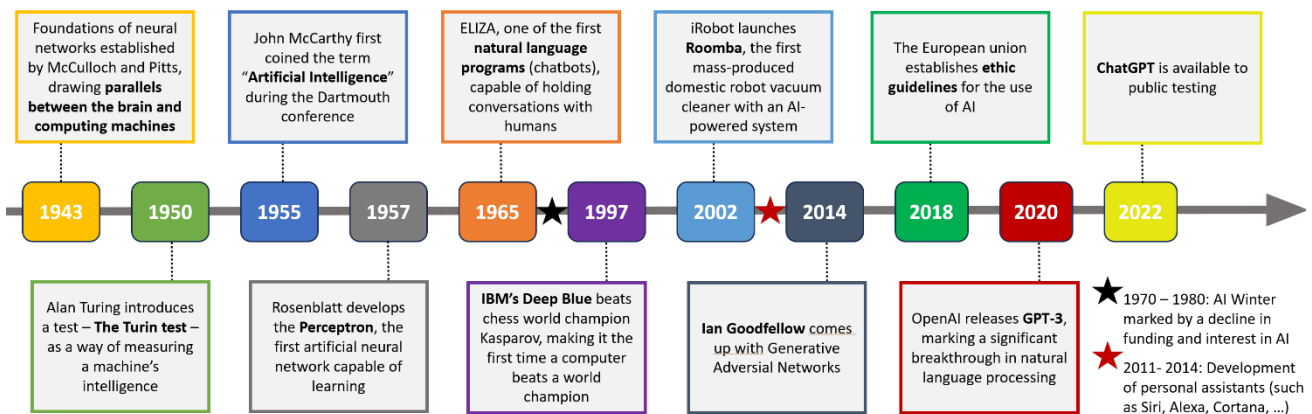


Figure 3-1: The history of artificial intelligence

### 3.1.2. Learning approaches

Today the term AI refers to a system's capacity to accurately comprehend external data, assimilate knowledge from this information, and subsequently apply these acquired insights to accomplish predefined objectives and tasks through adaptive methods. To achieve this, AI

<sup>4</sup> Go is considered to be more complex than chess, with 361 possible opening moves.

relies on three categories of learning processes, each with its unique characteristics and functionalities [87]:

- *Supervised learning*, where AI systems are provided with a labelled dataset. These labels act as guides, enabling the system to make predictions or classifications based on the patterns and relationships it identifies within the data.
- *Unsupervised learning* operates without labelled data. Instead, AI systems autonomously uncover hidden structures and patterns within the input data, grouping similar data points together.
- *Reinforcement learning* involves an AI system that learns by receiving feedback in the form of rewards or penalties for its actions, adapting and refining its strategies to maximize its cumulative rewards. This learning approach is analogous to trial and error, where the AI system learns by experiencing the consequences of its decisions and actions in a dynamic environment.

The choice of the learning approach in AI is closely tied to the nature of the problem at hand and the availability of input data. For instance, when it comes to classifying images of animals (see [Figure 3-2](#)), the supervised approach is often favored if a pre-labelled dataset is available. In this scenario, animal images come with labels indicating which animal is depicted in each image. The supervised learning algorithm utilizes these labels to learn how to identify distinctive features of each animal, thus enabling accurate classification of new images. We are talking about classification in this case. Conversely, if the dataset lacks labels, the unsupervised approach is more suitable. In this case, the model aims to discover similarities and groupings within the images without prior knowledge of animal categories. It can unveil underlying structures and allow the creation of groups of similar animals, albeit without specific labels. We refer to this task as clustering.



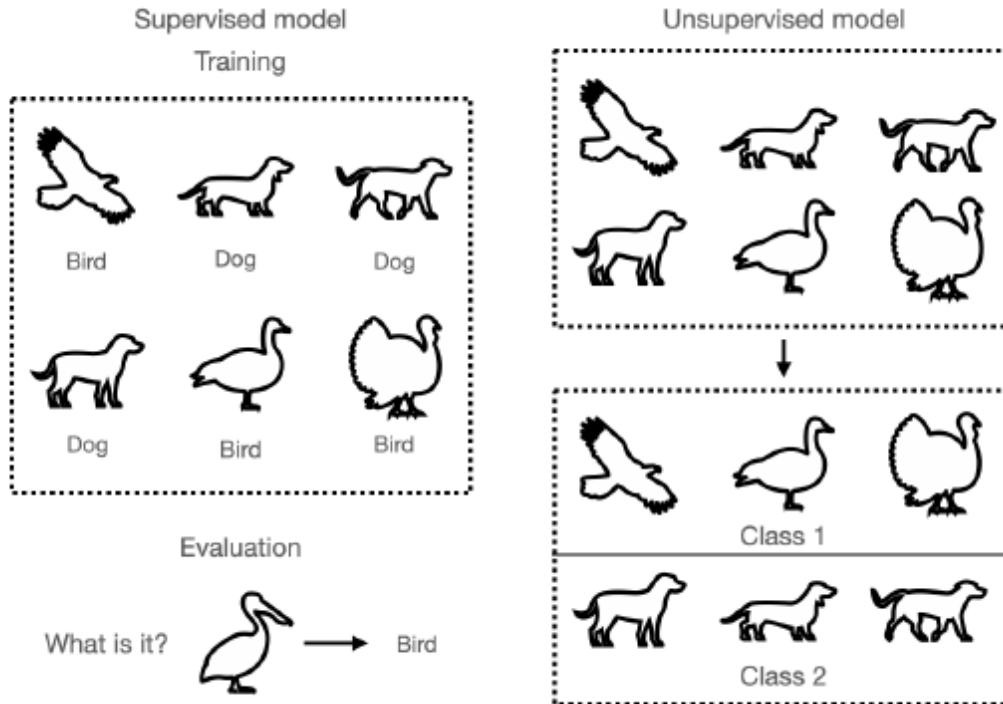


Figure 3-2: Distinction between supervised and unsupervised learning - © [88]

### 3.1.3. Traditional AI, ML, DL

The level of complexity of the model dictates the specific branch within AI at which we position it (see Figure 0-4).

**Traditional AI** is rooted in the utilization of rule-based systems, which subsequently gave rise to expert systems, often referred to as knowledge-based systems [89]. These expert systems are designed to imitate the decision-making capabilities of humans by employing a collection of predefined rules and conditional logic. Nonetheless, the application of rule-based systems for decision-making can entail the incorporation of a multitude of rules [90]. Traditional AI encounters limitations primarily due to the extensive quantity of rules that are requisite even to define seemingly simple objects, which can become unmanageable and challenging to maintain as the complexity of the tasks increases.

**Machine Learning (ML)**, as a subfield of AI, encompasses a range of techniques where algorithms learn patterns and relationships within data to perform specific tasks. According to Arthur Samuel in 1959 (who first introduced this term), ML gives "computers the ability to learn without being explicitly programmed." [91]. Therefore, ML delves into the examination and creation of algorithms capable of acquiring knowledge from data and generating predictions. These algorithms transcend the conventional reliance on static program instructions by deriving data-driven forecasts or decisions through the development of models based on input

samples [92]. Machine Learning frameworks enable researchers, data scientists, engineers, and analysts to generate consistent conclusions and results while revealing hidden insights by assimilating knowledge from past data relationships and trends [93].

An integral subdomain within the field of Machine Learning is **Deep Learning (DL)**. Deep Learning architectures are fundamentally underpinned by the perceptron, serving as the foundational building block for neural networks, which frequently operate on extensive or substantial datasets [90]. The term "deep" in DL indicates that there are many layers involved in the data transformation process. Hence, each level learns to transform its input data into a more abstract and composite representation. Importantly, a DL system can figure out on its own which features should go into each level for the best results [94].

### 3.1.4. From perceptron to neural networks

- **Perceptron**

In the 1950s and 1960s, scientist Frank Rosenblatt developed perceptrons, drawing inspiration from prior research by Warren McCulloch and Walter Pitts [95]. A **perceptron** is a computational unit that accepts multiple inputs, denoted as  $x_1$ ,  $x_2$ , and so forth, and generates a single binary output. Frank Rosenblatt proposed a straightforward algorithm to calculate this output. In this method, weights ( $w_1$ ,  $w_2$ , etc.) are introduced, representing real-valued coefficients that indicate the significance of the respective inputs in influencing the output. The decision regarding the neuron's output, which can be either 0 or 1, is contingent upon whether the weighted summation  $\sum w_i \cdot x_i$  is less than or greater than a predefined threshold value. Similar to the weights, the threshold constitutes a real number and serves as a parameter of the neuron. It is common to move the threshold to the other side of the inequality, and to replace it by what will be referred to as perceptron's bias  $b$  (see [Figure 3-3](#)).

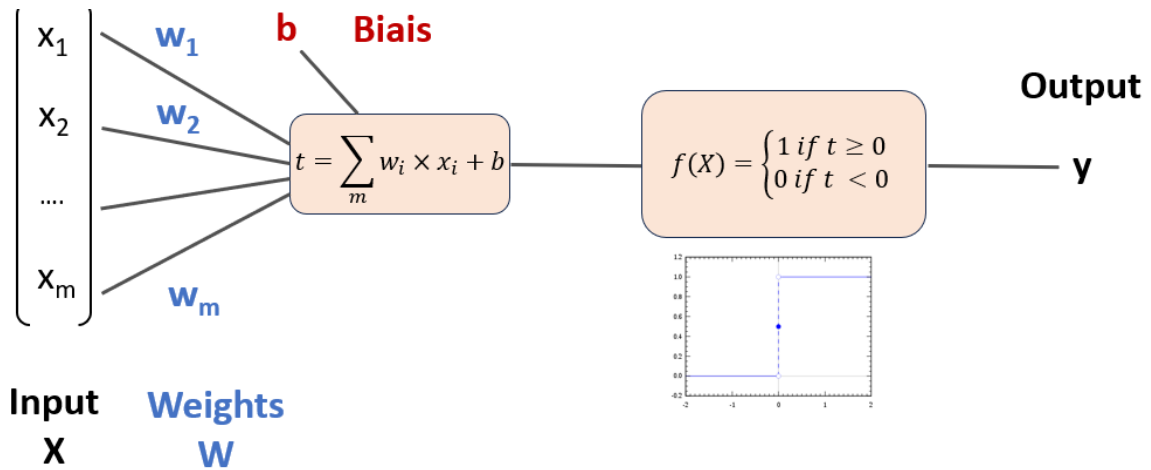


Figure 3-3: Perceptron architecture

Once the architecture of a perceptron has been established, the essential objective is to perform learning to adjust weights and bias so as to obtain a response corresponding to the model's expectations. However, this process can be tricky due to the inherent volatility of perceptrons. A slight change in weights can cause a sudden swing in results, making the model unstable and difficult to train consistently. This is where sigmoid neurons, also known as **artificial neurons**, come in. These processing units use a **sigmoid activation function**, enabling them to generate continuous, graded outputs (see Figure 3-4). The smoothness of the sigmoid function means that small changes in the weights and in the bias will produce a small change in the output. As a result, sigmoid neurons can model non-linear relationships more flexibly, making them more suited to solving complex problems [96]. This transition from perceptrons to sigmoid neurons has considerably improved the stability of neural network models.

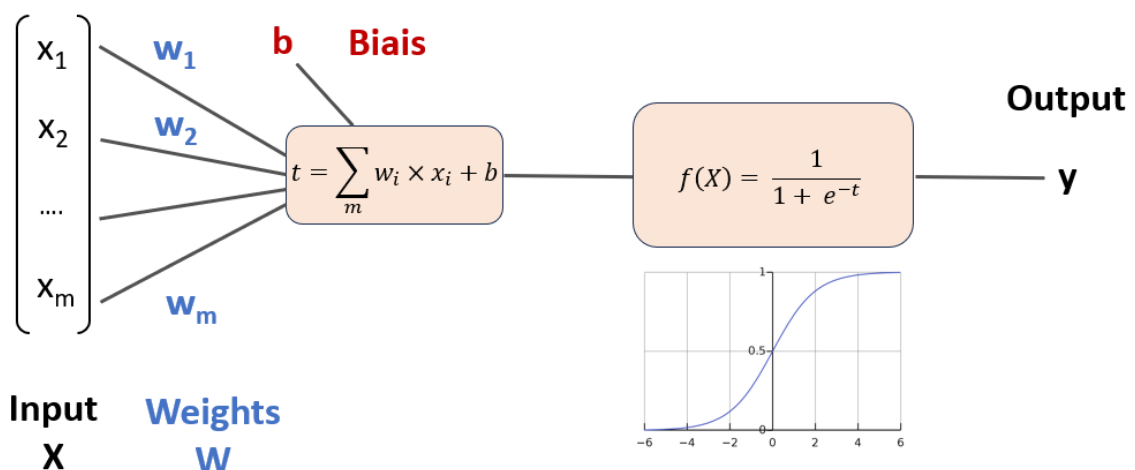
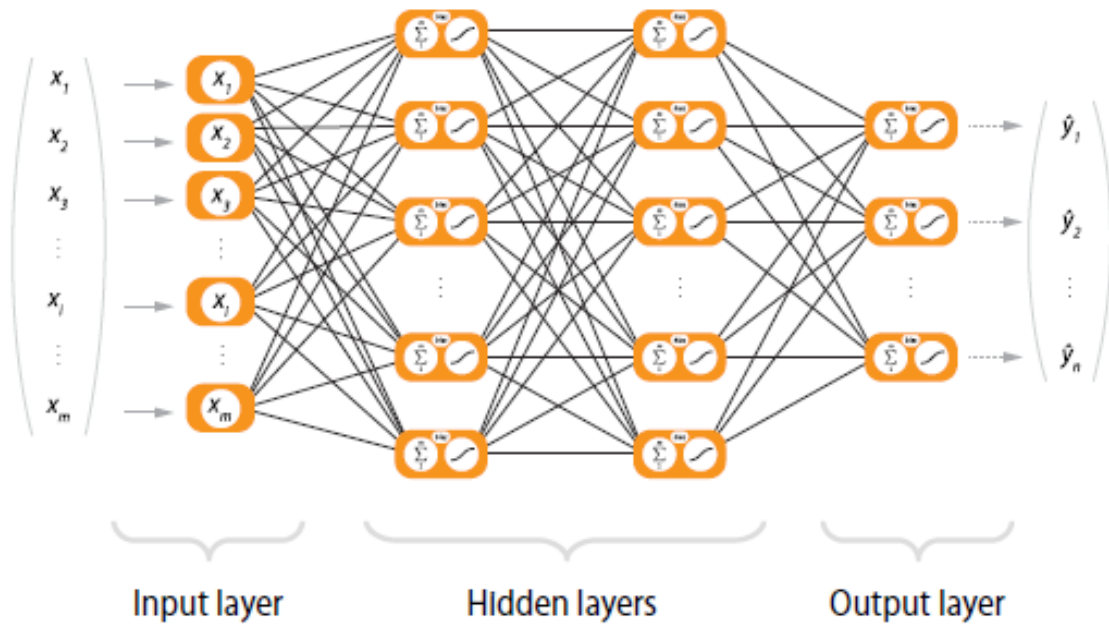


Figure 3-4: Artificial neuron architecture

- **Neural networks**

Whether it's the perceptron or the artificial neuron, they are both simplified versions of the human decision-making process. These models are limited in their ability to solve complex problems. To overcome this limitation, researchers developed **artificial neural networks**, which are hierarchical structures composed of multiple layers of interconnected neurons [97]. Each neuron performs a non-linear data transformation operation, enabling the network to capture complex, abstract relationships within the data (see [Figure 3-5](#)).



**Figure 3-5: Architecture of a neural network - © [98]**

The first layer, known as the input layer, receives raw data as input. Intermediate layers, known as hidden layers, progressively process this data, extracting increasingly abstract features as the information passes through the network. Finally, the output layer generates the network's final response or prediction. Each connection between neurons is associated with a weight that determines the relative importance of the information conveyed. The neurons integrate the incoming signals, apply an activation function to produce an output, and transmit this output to the neurons of the next layer. This layered structure enables neural networks to model complex relationships by learning hierarchical representations of data. This is the most basic neural network structure. In this presentation, we have mentioned the sigmoid and Heaviside (Boolean) functions, but we emphasize that there is a multitude of other activation functions, each adapted to specific contexts (see [Appendix A](#)). Similarly, when it comes to neural network architectures, there is a variety of configurations, each suited to particular tasks (see [Appendix B](#)). In this work, we will focus on those architectures and activation functions that are best suited to our research context. These choices will be thoroughly examined and

explained in detail in **Part II** dedicated to materials and methods, enabling a clear and justified understanding of the adopted approach for our study.

- **Learning of neural networks**

Learning is the process through which a neural network adapts and enhances its ability to perform a given task by assimilating insights from observed examples. It entails the fine-tuning of the network's connection weights and thresholds, with the aim of improving the accuracy of its outcomes (see **Figure 3-6**). This refinement is achieved by minimizing the errors encountered during the learning process. In cases of supervised learning, the network is compelled to converge towards a specific final state while being presented with corresponding patterns. In contrast, unsupervised learning allows the network to autonomously converge to various final states upon pattern presentation. Practically, this is implemented by defining a **loss / cost function** that is periodically evaluated during the learning process [99]. As long as the loss function's output consistently diminishes, the learning process persists<sup>5</sup>. Typically, the cost function is expressed as a statistic, such as Root Mean Square Error (RMSE) or Mean Squared Error (MSE). **Backpropagation** serves as a method to modify the connection weights in response to errors identified during learning. Technically, backpropagation calculates the gradient (i.e., derivative) of the loss function associated with a specific state concerning the network's weights. Weight updates can be executed using techniques like stochastic gradient descent (**Appendix C**). The learning rate plays a pivotal role in this process, determining the size of adjustments made by the model to rectify errors in each observation. A higher learning rate accelerates training but may result in lower ultimate accuracy, while a lower learning rate extends the training duration with the potential for higher accuracy. To manage this, **optimizers** are employed to dynamically adjust the learning rate as the learning process unfolds [100].

---

<sup>5</sup> Learning can also be forced to stop after a number of iterations / epochs.

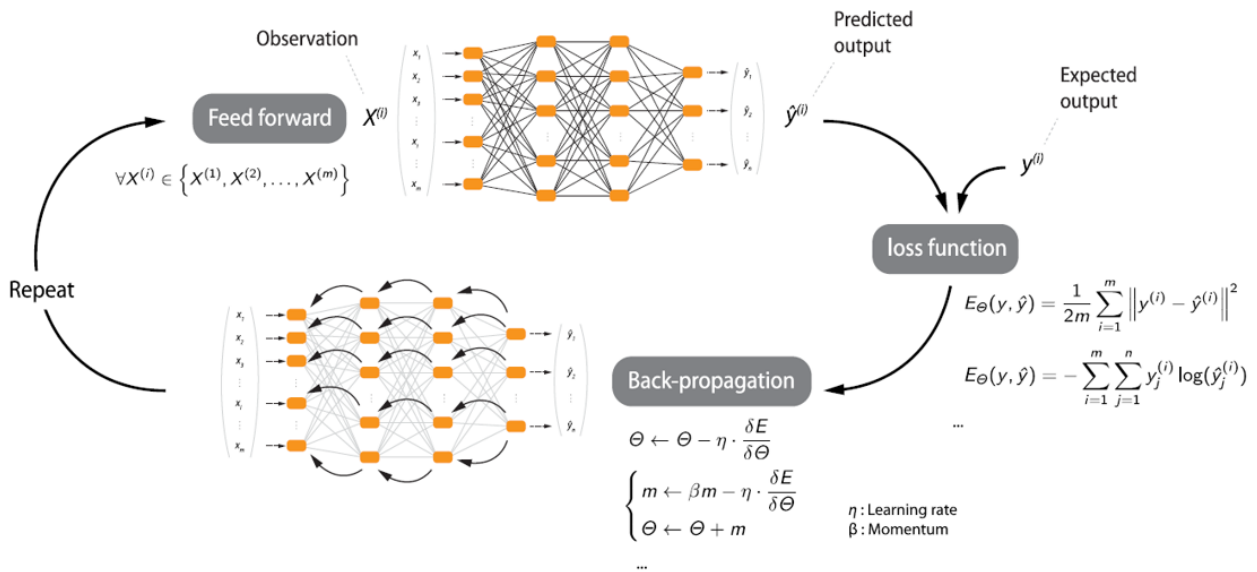


Figure 3-6: Learning process of a neural network - © [98]

When designing a neural network, the judicious selection of **hyperparameters** is crucial to ensure efficient learning. Hyperparameters include the number of hidden layers, number of neurons in each layer, activation function, learning rate, optimizers, and many others. Commonly used techniques for hyperparameters tuning include Bayesian optimization, random search, and optimization algorithms such as Grid Search and Random Search, which efficiently traverse the hyperparameter space in search of optimal configurations [101].

Inadequate tuning of these hyperparameters can lead to inefficient learning and poor results. Among the challenges commonly encountered when tuning hyperparameters is the risk of **overfitting**. The latter occurs when the model adapts excessively to the training data, even capturing noise present in the data, to the detriment of its ability to generalize on new data [102]. This results in excellent performance on training data, but poor performance on unknown data. Another major challenge when training a neural network is the risk of convergence to a **local minimum of the cost function**. Neural networks are characterized by highly complex and non-linear cost functions, which means that there are many peaks, valleys, and troughs in the parameter search space (see Figure 3-7). Hence, it is possible for the optimization algorithm to converge on a local minimum, thus leading to poor performance. Several techniques are used to mitigate overfitting and local minima. **Cross-validation** can be useful for exploring various configurations and identifying robust models, by initializing the network weights differently at each run, thus allowing a better generalization. The use of advanced optimization algorithms can also help overcome problems of convergence to local minima [103].

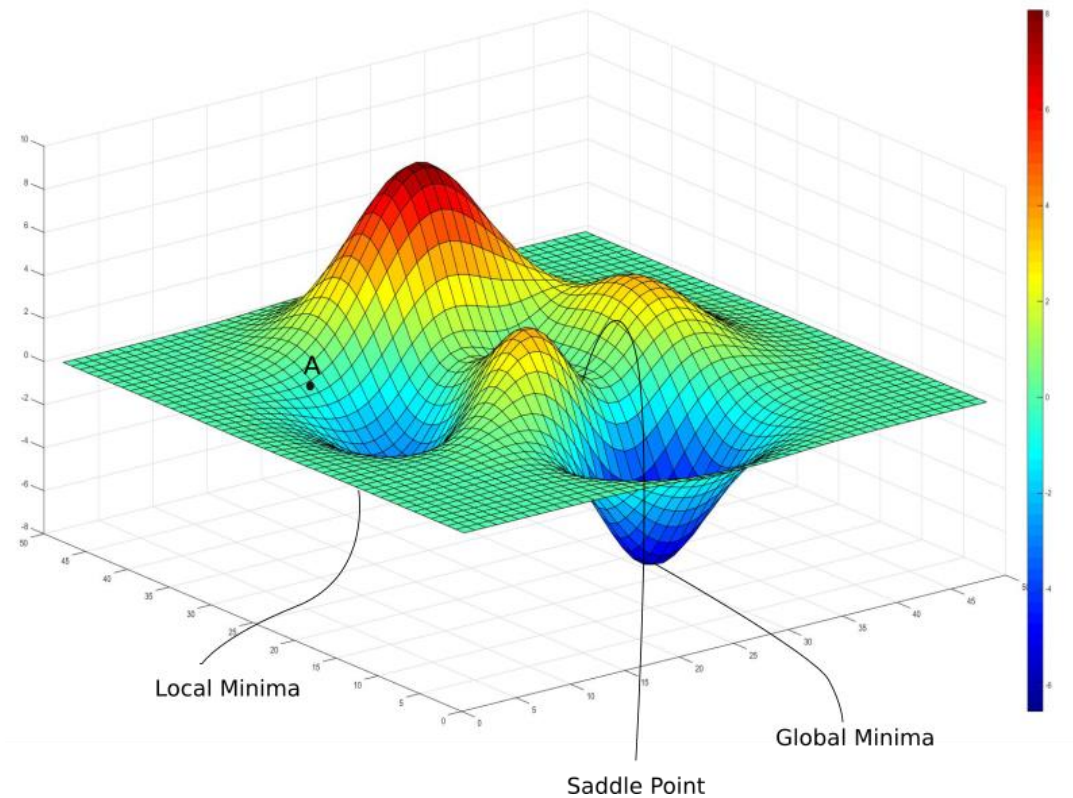


Figure 3-7: Example of loss function with a local minima that is different from the global one - © [104]

### 3.2 Key Considerations Before Implementing AI: Questions to Ask

Once **the objective** is set, the use or the development of AI models generally follows a three-step process. First of all, **input data** is collected and prepared, to serve as a learning base for the AI model. Next, the **model learning process** is launched, during which the AI examines this input data to extract useful features and information. This learning phase can be supervised (with labelled data) or unsupervised (without prior labels). Finally, once the model has learned from the data, it is able to generate **outputs** or predictions (see Figure 3-8).

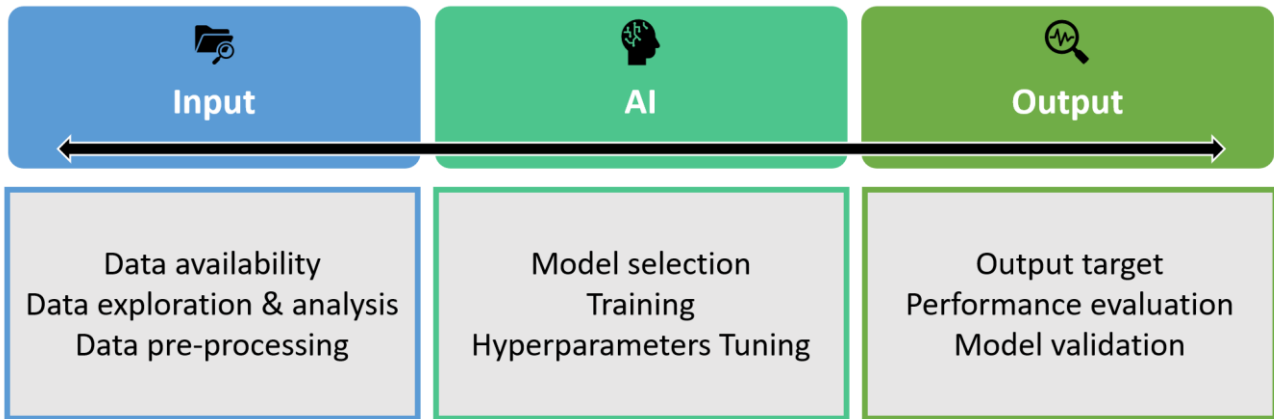


Figure 3-8: AI Implementation process

Before diving into model selection and the technical phases of analysis and learning, it's imperative to take a step back and ask three fundamental questions: “**What do we want to do?**”, “**What do we have as input?**” and “**What are we looking to produce as output?**”. A well-delimited objective is the cornerstone of any AI study. Moreover, a clear understanding of the available input data is essential to define the nature of the information available to the system. And similarly, having a clear vision of what output is required determines the purpose of the AI. These considerations then guide the choice of model, learning techniques, and the whole implementation process, ensuring that the AI is designed to respond effectively and relevantly to the specific needs of the task.

### 3.2.1. What do we want to do?

The aim of this study is to perform data validation and anomaly detection using AI models. Yet, **anomaly detection** constitutes a multifaceted field that has been extensively examined within the realm of AI. Numerous anomaly detection techniques have been purposefully developed to suit specific application domains, while others exhibit more generalized applicability [51]. The selection of an AI method primarily relies on the inherent characteristics of the input data. Input data can be broadly categorized into two distinct classes: **sequential data**, which encompasses voice, text, music, time series, and protein sequences, and non-sequential data, which includes images, tabular data, graph data and various other forms of data [105]. Researchers such as [51], [106] and [107] have offered comprehensive insights into anomaly detection techniques in general. However, our focus here is predominantly on **anomaly detection models specifically tailored for or adaptable to time series data**. Here, we are focusing on the analysis of time series from wastewater networks, more specifically turbidity data. These time series represent continuous records of measurements, captured at 5-minute intervals. This sampling frequency was selected to ensure adequate temporal resolution to capture the dynamics of events, while taking into account operational imperatives relating to



data management and processing. Our focus is exclusively **on time-delayed data validation**, an essential approach for a variety of applications (see [Figure 0-1](#)), including operational model calibration and collection system performance evaluation.

### 3.2.2. What do we have as input?

As previously mentioned, this manuscript focuses primarily on the analysis of sequential data, in particular time series from wastewater networks. In this context, we may be dealing with monovariate time series, where each series corresponds to an individual sensor, or multivariate time series involving combinations of sensors, from a same location or even different locations.

In the absence of precipitation, wastewater quality exhibits characteristic cycles on a daily and weekly scale. During rainy events, these cycles vary according to the intensity and duration of precipitation. Hence, the data must be pre-processed to enable the AI model to capture the inherent dependencies and patterns, which is made possible by **segmenting the measurement chronicles into subsequences**. However, determining the optimal sequence length poses a problem. By default, the dominant and basic seasonality in wastewater is 24 hours. This window size is therefore considered the default input size. Nevertheless, it is imperative to experiment to determine the window size that best captures the dynamics of the data..

Moreover, these input time series do contain inherent faults or anomalies, the second question is: **Do we have prior information on the location of these anomalies? Do we have labels or annotations for these data?** Labels are used as indicators to determine whether data falls within the normal behavior or is an anomaly. This implies a prior labelling process carried out by a human expert, following one of the approaches described in [Chapter 2](#), with all its inherent constraints: subjectivity, human error, and considerable cost. In addition, it should be noted that obtaining labelled data that is both accurate and representative of all types of behavior is often prohibitively difficult due to the infrequent occurrences of anomalies [51]. Moreover, anomalies are often dynamic in nature, for example, new types of anomalies may emerge for which there is no labelled training data. In some cases, such as fraud or malware detection, perpetrators are using increasingly sophisticated techniques [108], while in the environmental field, exceptional events linked to climate change are likely to become more frequent and unpredictable.

Hence, anomaly detection models can operate in one of three modes, depending on the availability of labels. In supervised deep anomaly detection, a supervised classifier is trained using labelled instances from both normal and anomalous categories. When faced with new

data, the model assigns it to one of these classes. However, supervised classifiers for anomaly detection may exhibit suboptimal performance due to class imbalance, with a significantly larger number of normal instances compared to anomalies. On the other hand, unsupervised anomaly detection techniques identify outliers solely based on inherent data properties. These methods are preferred when obtaining labelled data is challenging. A middle ground between the two is the use of semi-supervised approaches, which assume that the training data only includes labeled instances for the normal class. In this scenario, any deviation from the normal class is considered anomalous.

It must be emphasized that in any case, labelled data are necessary to evaluate the performance of the model during the development phase in order to guide its tuning. Although unsupervised models don't explicitly use labels to guide their learning, labeled data is nevertheless used as a baseline to assess the performance and pertinence of the results obtained. On the other hand, once the model has been tuned and validated for the specific use case, its deployment no longer requires the use of labels, whatever the learning method employed (supervised or unsupervised). In this phase, inputs are supplied directly to the model and labels are generated as outputs. If the model is to be improved / updated for other contexts, it can be re-trained using new data. If the ability to transpose the model to other contexts has been confirmed during the development phase, this learning update will not require the use of labels in the case of an unsupervised approach. However, if the initial model was supervised, the use of labels will remain essential for this re-training.

### 3.2.3. What are we looking to produce as output?

Admittedly, the fundamental aim of anomaly detection models is to pinpoint anomalies within the data as output. However, a crucial question arises as to how these models express these results. Depending on the **nature of the anomalies** (see [Section 1.3.2](#)), whether isolated points or sequences, the objectives of the detection model can vary significantly. The majority of existing work in the literature focuses on the detection of anomalous points, where the emphasis is on identifying individual instances that stand out from the rest of the dataset [51]. However, it is essential to note that the identification of abnormal sequences proves to be a more complex task. From an operational standpoint, certain applications such as the establishment of performance assessments for the sewer system may suffice with daily-scale validation, whereas other applications such as the calibration of a hydrodynamic model require a finely validated database at smaller time intervals. Consequently, it is crucial to clearly delineate **the scale at which faults are to be identified**. Hence, the objective of AI models is twofold: to detect invalid data at the acquisition time step and to identify sequences exhibiting a significant anomaly rate. The process can vary depending on the chosen models: either

validation occurs at the sequential scale followed by scaling down to the time step level by invalidating part or all of the time steps constituting the sequence, or alternatively, validation occurs at the time step scale followed by scaling up to the sequence scale based on a predefined threshold, beyond which a sequence is considered to have a sufficiently high anomaly rate for invalidation.

In general, the outputs produced by anomaly detection methods take the form of anomaly scores or binary labels. **Anomaly scores** are used to assess the probability of a data item being anomalous, providing a measure of the "degree of abnormality" of each instance. Consequently, the output from such techniques constitutes a ranked list of anomalies, from which an analyst can opt to examine the top anomalies or apply a predefined threshold to select anomalies. Conversely, **binary labels** offer a straightforward classification of data as normal or abnormal, without offering insights into the anomaly's severity. This approach can be valuable in scenarios where the key objective is to identify all anomalies rather than a predetermined subset.

### 3.2.4. Major Problem Complexities

By analyzing the various elements above, AI models for anomaly detection have to cope with some unique problem complexities [109] - [110]:

- *Unfamiliarity*: Anomalies are closely linked to elements that are not yet encountered, such as instances displaying unfamiliar abrupt behaviors. They often remain unknown until they actually materialize.
- *Heterogeneous anomaly classes*: Anomalies are inherently diverse, meaning that one category of anomalies may exhibit entirely distinct abnormal traits when compared to another category. For instance, in the domain of IoT-based environmental monitoring, anomalies related to air quality, humidity fluctuations, and temperature variations can each exhibit unique characteristics, necessitating specialized detection methods.
- *Rarity and class imbalance*: Anomalies typically represent data instances that are rare when compared to the prevalent normal instances, which tend to dominate the dataset. As a result, it can be challenging, if not entirely infeasible, to amass a substantial volume of labelled abnormal instances. This scarcity of extensively labelled data is a common issue across many applications.

### 3.3 Anomaly detection using AI in urban hydrology

Over the years, the **detection of anomalies in time series** has gained considerable importance in various fields, including cybersecurity [108], anti-fraud [111], and medical sciences [101]. However, the advent of the Internet of Things (IoT) has extended the scope of anomaly detection beyond these domains. With the prospect of "Smart Cities" and the proliferation of sensors to monitor the operation of water networks, time series analysis in the urban hydrology field has become an integral part of the IoT and can no longer escape its constraints [112]. Several major incidents, such as the compromise of the Maroochy water treatment system in Australia in 2000 [113], the direct terrorist attacks on US water supply networks [114], and the presence of *Aeromonas* in drinking water networks in Scotland and Turkey [115], have reinforced the need to use automated machine learning-based approaches for anomaly detection in urban network data series. These events have considerably stimulated research in this field, making it increasingly important to ensure the reliability of data from water management systems.

Indeed, the literature review conducted in support of this work is initially based on a survey undertaken by Dogo et al. [20], which lists articles about anomaly detection using AI in the field of drinking water quality from 2002 to 2018. Nevertheless, in order to maintain at the forefront of this constantly evolving research field, we have extended this review up to the year 2023 and expanded it to encompass the broader domain of urban hydrology, as delineated in **Appendix D**. It should be noted that in this survey, only publications providing a detailed description of their methodology and data source are taken into consideration.

With regard to **the fields of application** of these approaches, the analysis of the state of the art reveals a predominance of studies focused on monitoring water quality data in distribution systems, with an objective of detecting potential risks of contamination or intrusion. By contrast, wastewater-related studies account for a relatively modest 25% of the listed research, focusing mainly on data analysis within WWTPs. It is important to note that only one study [116], to the best of our knowledge, explored data from sewage networks, and even then, analysis was limited to data collected during the dry season. It is in this context that our research work finds its legitimacy and positions itself as a true contribution in this field.

Furthermore, when examining the **sources of data**, it is crucial to note that some databases originate from prototypes or testbeds [21], [117], which differ considerably from real-world conditions. In the laboratory, data is generated in a strictly controlled environment, following rigorous protocols, and collected at regular, reproducible intervals. In contrast, data collected in the field reflect real-world conditions, with uncontrolled sources of variability. What's more,

even in a field context, the dynamics of data within a treatment plant [118], where operation is controlled and exhibits stable dynamics with fewer fluctuations, differs from that of network data [119]. Moreover, even within the latter case, there are differences in the dynamics between drinking water and wastewater data. For example, drinking water data may show peaks during the dry season / summer, while wastewater data may show a more dynamic seasonality, influenced by various exogenous parameters. This understanding of the diversity of data sources is essential for adapting analysis methods and models to the specificities of each context.

Figure 3-9 summarizes the **different models** used in or more than two articles. The dominant approaches that emerge from this analysis can be classified as follows: classification using supervised approaches, unsupervised clustering approaches, and approaches using prediction.

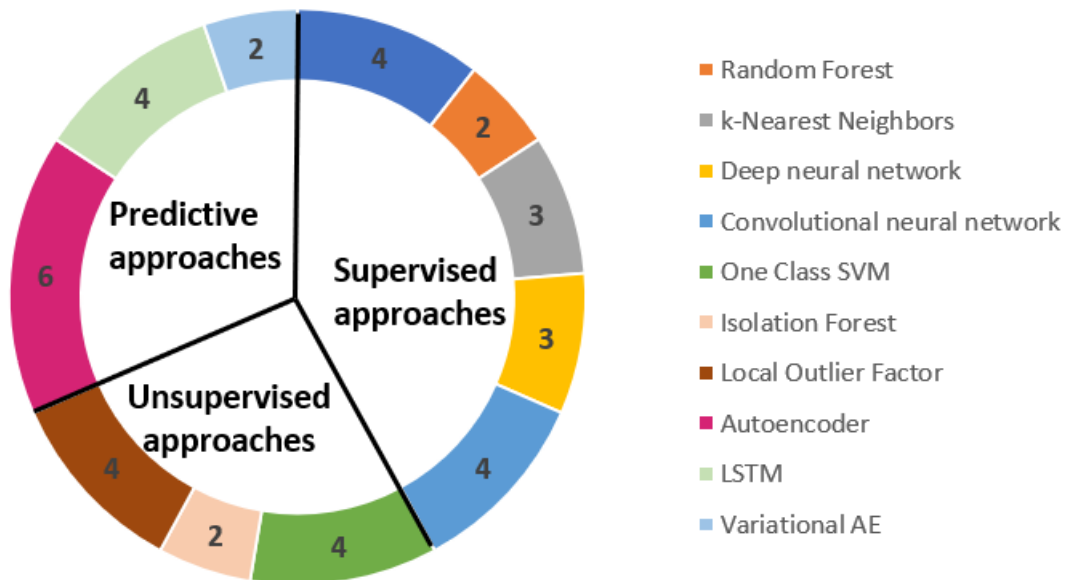


Figure 3-9: State-of-the-art AI approaches for anomaly detection in the urban hydrological field with their occurrence in Appendix D

### 3.3.1. Anomaly Detection through Classification Approaches

Supervised anomaly detection entails the process of learning a discriminating boundary from a set of labelled data instances during the training phase and subsequently employing this learned model to classify a test instance as either normal (often denoted as 0) or anomalous (often denoted as 1) during the testing phase. This approach falls within the category of classification-based anomaly detection techniques [120]. **The underlying assumption in these methods is that, given a feature space, there is a rule capable of distinguishing**

**between normal and anomalous classes.** Much research has investigated ML supervised models to improve the anomaly detection accuracy [121] , [122].

- **Traditional Machine Learning Models**

First of all, **Support Vector Machine (SVM)** stands as a state-of-the-art approach for classification tasks. Its fundamental principle lies in creating an optimal hyperplane, maximizing the margin between classes in a feature space, and categorizing data points based on their position relative to this hyperplane. In scenarios where the relationship between features and classes is not linear, SVM with kernel functions can be used. Kernels, such as the Radial Basis Function (RBF), Polynomial, and Sigmoid, transform the feature space to higher dimensions, enabling SVM to find complex decision boundaries that are capable of capturing non-linear patterns in the data [123]. [124] emphasized that the performance of a SVM kernel is contingent upon several factors, including feature selection and the nature of the dataset. When the input dataset exhibits linear separability, the linear SVM kernel tends to outperform other models.

Another state-of-the-art approach in ML for anomaly detection is based on the use of decision trees, in particular the **Random Forest (RF) model** [125]. The fundamental principle behind RF lies in the creation of a set of decision trees, where each tree is randomly constructed from a subset of the training data. These trees are designed to solve a classification problem by recursively dividing the feature space into subregions, thus creating decision rules. A test instance that is not covered by any rule is considered as an anomaly [126]. [124] proposed a performance analysis of RF using a real dataset retrieved from the water treatment station "Ghadir El Golla" of Tunis. The experimentation results are encouraging.

Last but not least, the **k-Nearest Neighbors (k-NN)** algorithm stands as another frequently employed supervised ML technique for addressing classification problems. At the core of the nearest-neighbor family lies the fundamental assumption that similar data points are close to each other, while outliers are typically far away from the group of similar data points. Data is presented as points in a multi-dimensional space, defined by the number of features used in the analysis. This allows to calculate distances between data points, usually using the Euclidean Distance. The dissimilarity between points depends on how far apart they are from each other [127]. In k-NN, we determine whether a data point is unusual by looking at its k-nearest neighbors, where k is usually a small number, ranging from 3 to 10. However, it's important to note that as the number of data points and the number of features increase, the efficiency of this method in making predictions significantly decreases [128]. Various research

studies have compared this model with other supervised ML or DL methods, and it has generally been found to perform less well than the latter [129], [119], [130].

Such traditional supervised machine learning approaches for anomaly detection have the advantage of being well documented, providing a solid basis for their understanding and application. Nevertheless, **there is much debate and controversy in the literature about the performance of these models, with results ranging from notable successes to major failures for identical models**. For example, [131] and [132] have evaluated the SVM model on the same database, but the results obtained show considerable variation, ranging from a mediocre performance with an F1 score<sup>6</sup> of 0.36 to a quasi-perfect performance with an F1 score of 0.99. This disparity in results is largely attributable to an initial data pre-processing phase that differs in the two cases. It is therefore clear that the performance of this model is subject to substantial variability, and its effectiveness is highly dependent on the quality of the input data. Studies that have applied these models to carefully constructed public data sets tend to achieve the most satisfactory results. However, in practical contexts, achieving such precision in labelling can be challenging. In addition, classification algorithms require an adequate distribution of data, both normal and abnormal, i.e., the data must cover the whole distribution to allow generalization by the classifier. New data can then be correctly classified, as classification is restricted to a "known" distribution. However, a new example from a previously unobserved region of the distribution (a new form of anomaly) may not be classified correctly, unless the generalization capabilities of the underlying classification algorithm are robust. These **models are often prone to over-fitting, a situation where the model overfits the training data, losing its ability to generalize effectively on new data**. In response to these challenges, the scientific community has gradually turned to more sophisticated approaches that make use of neural networks, offering a richer and more flexible learning potential.

- **Deep Neural Networks**

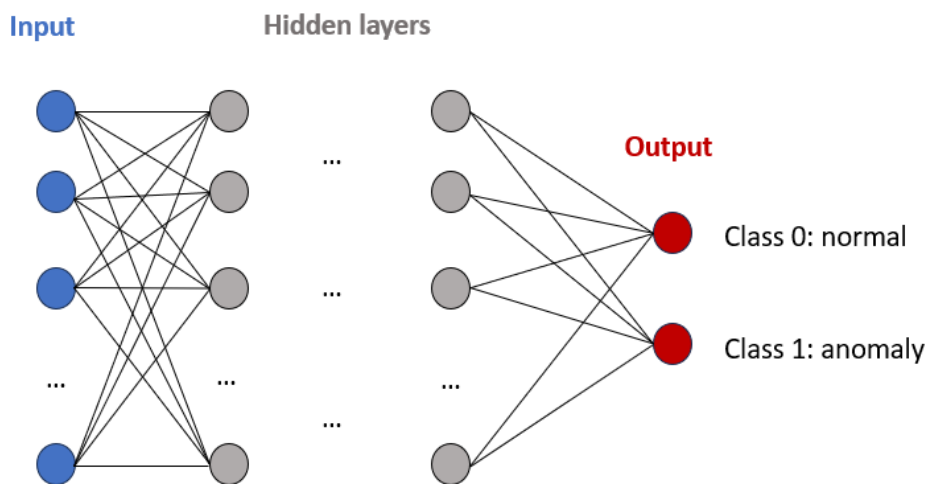
The increasing adoption of deep neural networks (DNN) in the field of time-series analysis and forecasting is largely driven by their high performance in computer vision tasks, such as object detection, classification, and segmentation [133]. Unlike traditional ML approaches, neural networks do **not require any prior assumptions about the underlying data generation process**. Their growing popularity can be attributed to the compelling results they have already delivered in practical applications [31]. The fundamental process of supervised classification

---

<sup>6</sup> This metric will be explained later. In general, it ranges from 0, indicating no performance, to 1, signifying perfect performance.

using DNNs involves two key stages. Initially, the neural network is subjected to learning with the known input-output training dataset to acquire knowledge about distinct normal and abnormal classes. Subsequently, each test instance is introduced as an input to the neural network, which, in turn, yields a probability associated with each class or directly identifies the most probable class. In this review, we focus on two predominant DNN architectures employed for the task of anomaly detection in the urban hydrology field through the classification approach: the Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN). These specific architectures are emphasized due to their common use in DL models for time series classification, as widely recognized within the field [134].

The most fundamental deep neural network architecture is the **Multi-Layer Perceptron (MLP)** [38] which is a fully connected neural network, where each layer's neurons apply a non-linear transformation to the input data, and this transformation is determined by the weights and bias associated with each connection (see **Figure 3-10**).



**Figure 3-10: Example of MLP architecture for anomaly detection**

For time series classification, the final layer typically employs a SoftMax activation function (see **Appendix A**), allowing the network to produce a probability distribution over different classes (Class 0: normal / Class 1: anomaly). The MLP learning process follows the methodology described in **Section 3.1.4**. It begins with an initialization stage, in which the network's weights and biases are generally defined at random. Then, during the training phase, the network evaluates the loss function, which measures the deviation between the model's predictions and the true labels of the training data. Generally, for classification tasks, we use categorical cross-entropy as a loss function (see **Appendix E**). To minimize this loss and improve model performance, backpropagation is used in order to calculate the gradients with respect to network weights and biases, step by step, working backwards from the deepest



layers to the input layers. These gradients are then used to adjust the weights and biases according to available optimizers, such as SGD, Adam-based optimization or other optimization algorithms. This iterative process continues until the model achieves satisfactory performance on the training set. As a reminder, the number of layers and the number of neurons per layer are considered as hyperparameters.

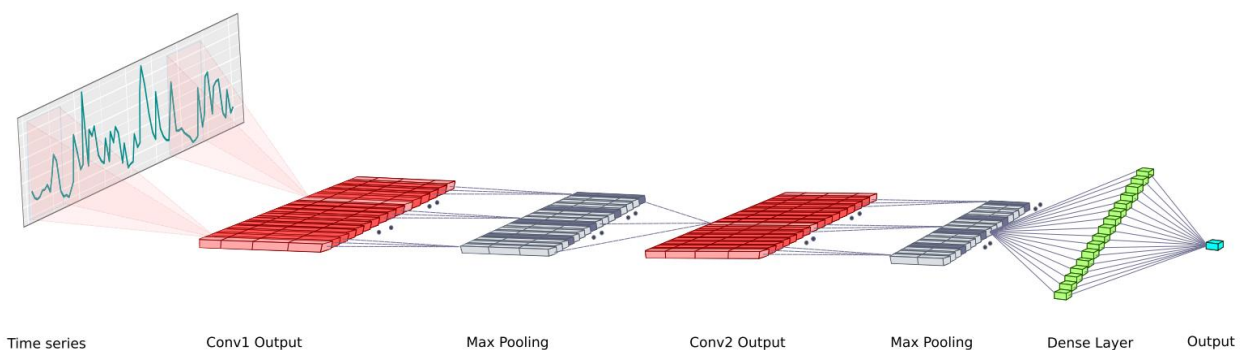
Numerous investigations have employed MLP for the task of anomaly detection in time series, yielding promising results [135], [136]. Other studies have compared MLP with other supervised machine learning approaches and demonstrated its superior performance [119]. This success has positioned MLP as a foundational model for this specific task. However, when considering the input of the MLP, it's crucial to acknowledge the fundamental distinction between classifying static patterns and time series data, which lies in the dimension of time. In static pattern classification, individual patterns are typically unrelated, allowing each one to be independently processed. However, this distinction poses a challenge when applying MLPs to time series data. In this case, **each timestamp in a time series is assigned its weight, resulting in the loss of temporal information.** Hence for time series classification, MLPs must be implemented using sliding windows [137]. In this approach, the window size aligns with the number of neurons in the MLP's input layer, facilitating the consideration of temporal dependencies.

To overcome the challenge of capturing temporal dependencies in MLPs, **Convolutional Neural Networks (CNN)** have started to gain traction in anomaly detection within time series data, despite their initial development for image processing [138]. According to [134], CNN is the most frequently utilized architecture for time series classification problems, largely owing to its robustness and relatively shorter training time compared to alternative architectures. This is precisely where CNNs prove highly effective, as they excel in learning spatially invariant filters or features directly from the raw input time series [133]. Unlike MLPs that employ fully connected layers, CNNs use convolutional layers which have partial connectivity. This characteristic leads to a reduction in the number of parameters and allows CNNs to achieve deeper architectures with faster training. A key distinction between CNNs and MLPs is their emphasis on local patterns within the data [31].

CNNs operate by sliding filters over time series data, applying a convolution operation, and using non-linear activation functions like the Rectified Linear Unit (ReLU) (see **Appendix A**). This results in a filtered time series that can be interpreted as a new set of features. Unlike MLPs, which treat each time stamp independently, CNNs benefit from weight sharing, meaning the same filter is applied to all time stamps, capturing temporal dependencies. Local or global pooling operations can be applied to further aggregate data and aid in convergence [139]. A

final discriminative layer, often implemented with a SoftMax operation, provides class probabilities (see [Figure 3-11](#)). CNNs are trained through a feed-forward pass followed by backpropagation, similar to MLPs, allowing them to discover intricate temporal features. Hence, the architecture of a CNN, including the number of convolution and max-pooling layers, stands as a critical hyperparameter. The specific configuration of these layers may vary depending on the dataset being processed.

Research in the field of urban hydrological anomaly detection has widely employed CNNs in combination with other predictive models such as autoencoders (AEs), variational autoencoders (VAEs) or long-term memory recurrent neural networks (LSTMs) (see [Section 3.3.3](#)). This combination aims to evolve from a model designed for image processing to one adapted to time series analysis [140], [117], [116]. However, it should be noted that this approach is not imperative, and it is possible to directly use a CNN model for processing temporal data by resorting to 1D convolution layers [141].



**Figure 3-11: Example of a Deep CNN for time series processing - © [141]**

### 3.3.2. Exploring Anomaly Detection in Unsupervised Mode

When working with datasets for which access to labels is not available (which is often the case due to the costs and constraints associated with obtaining them), it is common to turn towards unsupervised approaches for anomaly detection. These anomaly detection methods seek to identify unusual patterns and structures within the raw data without relying on labelled data. The model analyses the data as a stationary distribution. It is assumed that anomalies are distinct from 'normal' data, resulting in their detection as outliers [107]. This approach is particularly valuable when obtaining labels is difficult, costly, or simply impossible, enabling anomaly detection to be carried out in an unbiased way and without relying on external supervision.

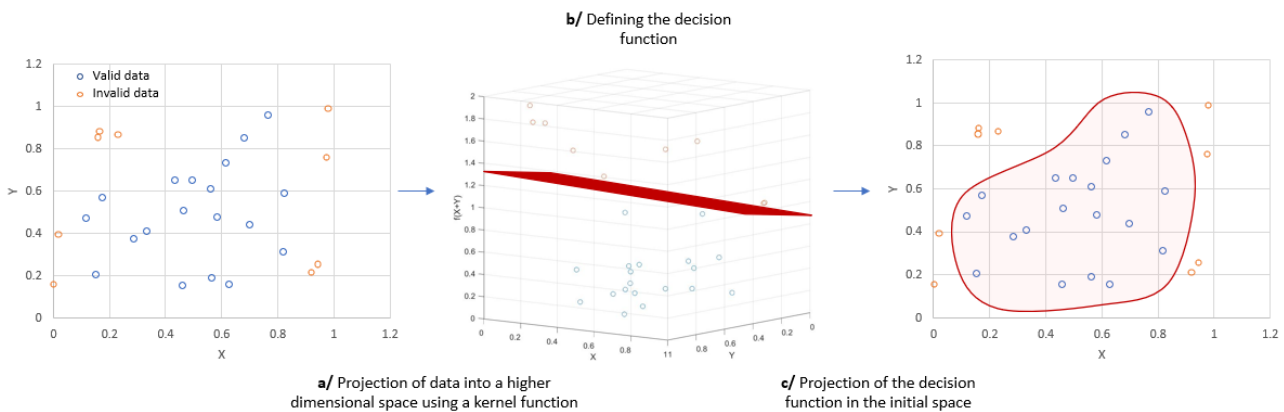
In this context, various models have been developed and studied, which are mainly non-deep models. It is important to note that the scientific community is constantly active in this field,

which frequently leads to the emergence of variants of the same models or principles [142]. These variants aim to meet various challenges, such as adapting to more complex data or improving performance in terms of computation time.

- **Majority modeling**

In an unbalanced clustering approach, one can consider that temporal data are generally grouped into a majority cluster that represents normality and anomalies are seen as values that deviate significantly from this main cluster.

A modification of the SVM algorithm was introduced to adapt it into an unsupervised learning algorithm, **One Class Support Vector Machine (OCSVM)** [143]. While the conventional supervised SVM algorithm aims to optimally separate two distinct classes of data in feature space using a hyperplane, OCSVM assumes that all training instances share a single class label and endeavors to separate the entire set of training data from the origin. In other words, it seeks to identify a small region where the majority of data points are concentrated and labels the data within this region as a single class, normal class. Any test instance that falls outside the learned boundary is designated as an anomaly (see **Figure 3-12**). The complexity of the models can be adjusted by employing different values for OCSVM parameters, such as the variance parameter of radial basis functions (RBFs) and the expected outlier rate.



**Figure 3-12: The principle of OCSVM**

The original OCSVM method was designed for detecting anomalies in sets of vectors rather than time-series data. Many research papers suggest a common practice of projecting time-series data into a vector representation for effective anomaly detection using OCSVM. [144] adapted this model to temporal data via pre-processing steps and evaluate its performance on telecommunication network data. Moreover, several studies have adopted the OCSVM model for anomaly detection in urban hydrology, with notable results. For example, in [21], the OCSVM was shown to identify attacks within a drinking water consumption chronicle, while in

[145], it was used to detect anomalies in water level variations in springs. Nevertheless, it should be noted that, in the first case, the anomalies are simulated, which implies that they follow a predictable dynamic, unlike reality where anomalies manifest themselves more randomly. With regard to the second example, the dynamics of water levels within springs are characterized by a slow evolution, substantially different from that observed in a sewage network.

On the other hand, the non-supervised variant of Random Forests is known as **Isolation Forests (IForest)**. These forests are constructed using randomized decision trees without any predefined labels, and their primary objective is to isolate outliers within sparse clusters [146]. IForest was developed based on the assumption that anomalies represent "few and distinct" data points in a given dataset. When constructing these trees, the method recursively selects random features and random split values as tree nodes, aiming to isolate the samples in the leaves of the tree. The measure used for quantifying the isolation of a sample is expressed as the path length throughout the tree. As anomalous samples are typically easier to segregate than normal ones, they tend to be closer, on average, to the root of the tree and consequently have shorter path lengths. Hence, path lengths serve as indicators of the normality of samples, and their reciprocal values are translated into anomaly scores (see [Figure 3-13](#)). However, the creation of a single tree may not provide a precise overview of the entire dataset. Therefore, multiple trees are generated, and the degree of outlierness of an object is determined based on the average path length of that object across all the trees. Consequently, samples with the shortest average paths are more likely to be identified as outliers [147].

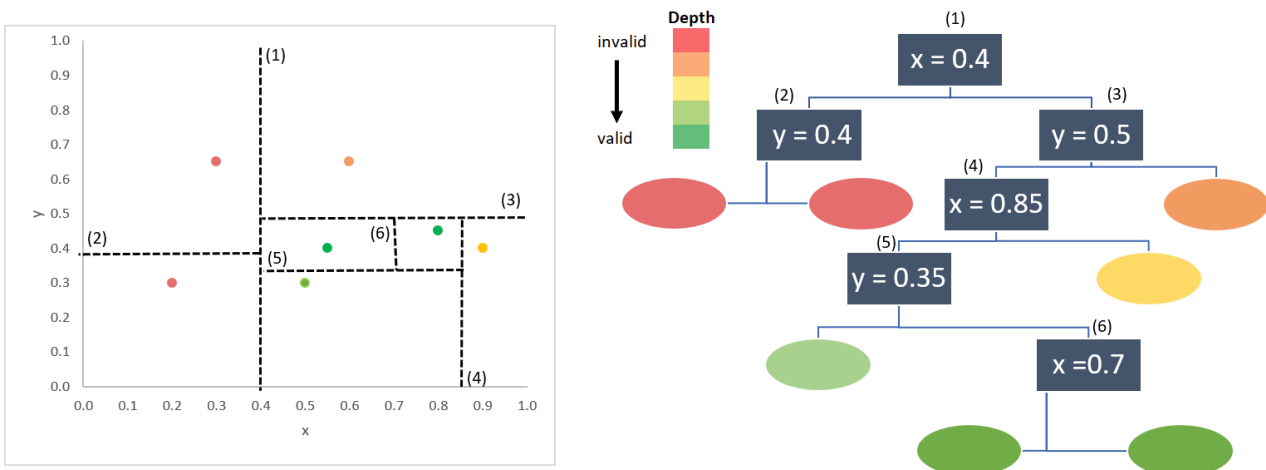


Figure 3-13: Example of the IForest principle

As with OCSVM, efforts have been made to adapt IForest to time series. [148] applied the sliding window methodology to segment time series data into subsequences. [130] highlights that, compared with other clustering approaches such as K-Means and k-NN, the IForest

algorithm is better suited to the characteristics of large hydrological time series, and ensures highly accurate anomaly detection. However, in the case of [149], the complexity of establishing a discrimination threshold between normal and anomalous data is highlighted.

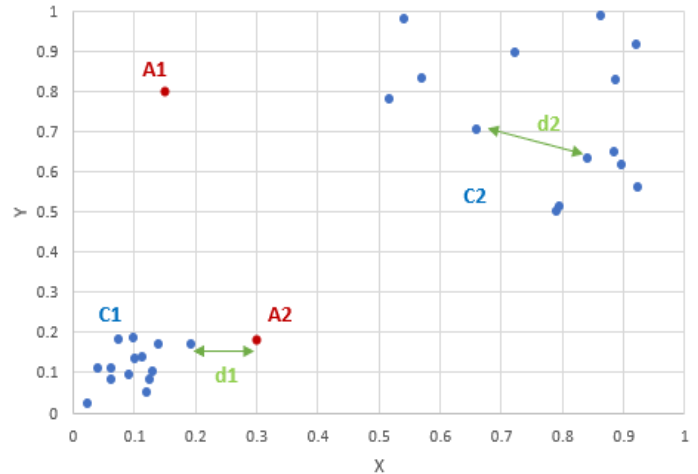
Both of these models (OCSVM and IForest) are categorized within the majority modelling approach, which operates on the assumption that normal data instances are tightly clustered in hyperspace [150]. This approach's objective is to delineate the decision boundary between outliers and normal data by characterizing the distribution of regular data [30]. However, **this assumption might not hold true in the context of time series data**. Time series data often exhibit temporal dependencies and variations that can lead to dispersed, less compact data distributions, especially when the data involves multiple variables or high dimensionality. As a result, OCSVM and IForest, originally designed for vector data, may not capture the temporal dynamics adequately. They could overlook properties related to temporal dependencies. These models may primarily detect global outliers, which tend to be located in sparsely distributed regions of the data. But they may not perform well in capturing local anomalies or anomalies influenced by high-dimensional features. **Their effectiveness in identifying time series anomalies can be limited by the failure to consider the temporal aspect and high dimensionality.**

- **Distance-based approaches**

Another approach is to use **proximity-based methods**, which rely on the distances between data measurements to distinguish abnormal from correct readings. Distance methods use specialized distance metrics to compare points or subsequences in a time series with each other. Abnormal subsequences are assumed to have greater distances to other subsequences than subsequences with normal behavior. For distance calculations, algorithms in this family can consider all other subsequences, only certain nearest neighbors, or certain cluster centroids as reference points for distances. A well-known proximity-based algorithm is the **Local Outlier Factor (LOF)** [151].

The LOF model operates under the assumption that anomalies tend to reside in low-density regions of the data. Therefore, it identifies outliers based on their local divergence from their neighboring data points. LOF calculates a density score for each data point by evaluating its proximity to a set of its  $k$  nearest neighbors. Data points with lower density scores are more likely to be classified as anomalies. To estimate the local density, LOF measures the typical distance at which a data point can be reached from its neighbors [152].

Consider a 2-dimensional dataset with two anomalies A1 and A2 as illustrated in **Figure 3-14**. The first is a global anomaly, as it belongs neither to cluster C1 nor to C2. A2 is a local anomaly, as shown by its distance from its nearest neighbor compared to the size of the nearest cluster, in this case C1. Yet its distance is still less than the distances that can be measured in cluster C2, made up of normal points. To address variations in data density, the LOF score of a specific data point is calculated as the ratio between the average local density of its  $k$  nearest neighbors and its own local density. If a data point is normal and located within a densely populated region, its local density will closely resemble that of its neighboring data points. Conversely, an anomalous data point will exhibit a lower local density in comparison to its nearest neighbors. Consequently, the anomalous data point will receive a higher LOF score. In the example, LOF successfully identifies both anomalies (A1 and A2) by considering the density of data points [17].



**Figure 3-14: Example of local density for LOF model**

LOF was initially designed to detect anomalies on spatial data [151]. However, in subsequent work, researchers, such as [153], extended the approach to time-series data. A number of research studies have demonstrated the effectiveness of the LOF model in various contexts, including the analysis of drinking water consumption data [129] and data from water treatment plants [154]. Although these contexts differ from the conditions present in wastewater networks, it is important to note that existing studies cover a wide range of measurement frequencies, from 2 hours [155] to 1 minute [149]. This diversity of frequencies suggests that the LOF model could potentially be adapted to data with significant dynamics. However, it's worth noting that LOF's performance is contingent on careful parameter tuning.

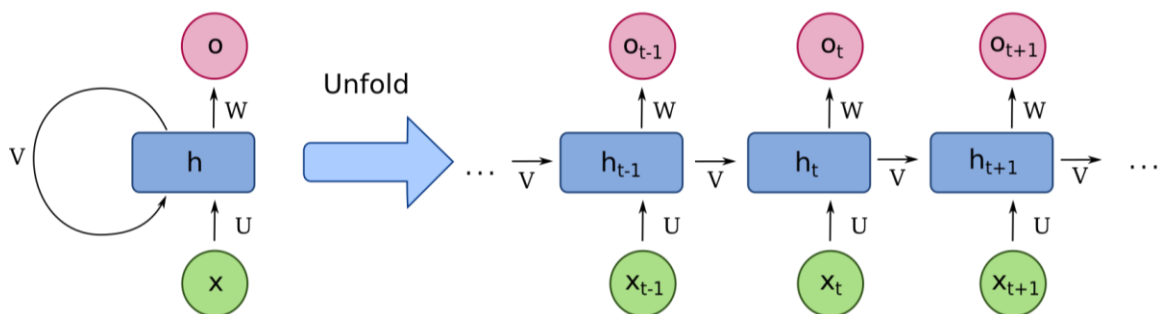
One notable benefit of proximity-based methods is their independence from any assumptions about the underlying data distribution, relying solely on the data itself. Nevertheless, **these techniques may encounter challenges in cases where normal instances lack a sufficient number of nearby neighbors or where anomalies possess a considerable number of close neighbors**, potentially leading to misclassification and thus overlooking certain anomalies.

### 3.3.3. Detecting Anomalies Using Prediction Models

Another way to approach the anomaly detection problem in time series is through **prediction**. In this method, instead of looking for anomalies directly, one predicts what should be expected in a data set and compares these predictions with the actual data. If the predictions closely match the actual data, then the observations are considered valid. However, if the actual data diverge significantly from the predictions, this may indicate the presence of anomalies or anomalous events. This approach is based on the idea that anomalies are often values that differ significantly from expected trends or patterns [30]. Actually, this approach is similar to the use of statistical or hydrological models presented in **Chapter 2**, but the models used here for describing normal conditions do not imply any assumptions about underlying processes or distributions.

- **Recurrent neural networks**

**Recurrent neural networks** (RNNs) are emerging as a logical choice for tackling anomaly detection using prediction [156]. RNNs are a class of neural network architectures designed to handle sequential data, making them particularly suitable for modeling time series. They are able to learn temporal patterns and create predictions based on previous observations. This method exploits the ability of RNNs to maintain a memory of past observations and use this information to make real-time decisions. This makes it a suitable tool for anomaly detection in time-series applications, where anomalies are often hidden over time and require a detailed understanding of sequential dependencies to be identified. In contrast to MLP and CNN, where the data is just flowing forward, RNN networks have a feedback connection enabling them to use the output information for the next input of the sequence.



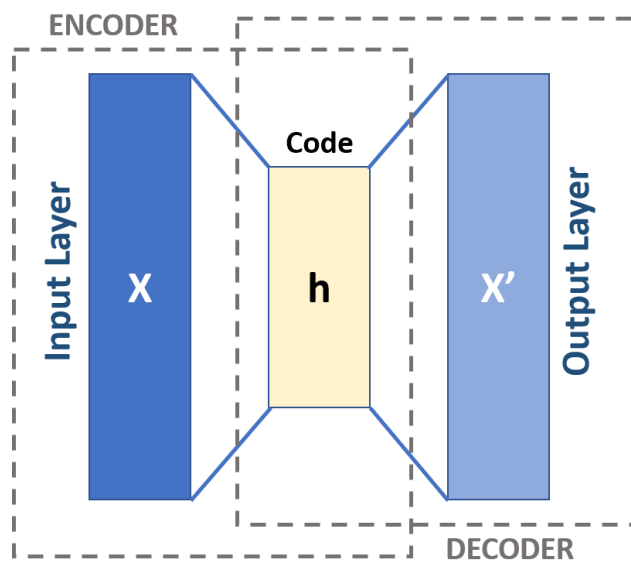
**Figure 3-15: A diagram for a one-unit recurrent neural network (RNN).**

**From bottom to top: input state, hidden state, output state. U, V, W are the weights of the network. Compressed diagram on the left and the unfold version of it on the right.**

Nonetheless, conventional RNNs found limited applications in time series classification, primarily because of two critical factors. Firstly, RNNs frequently encounter the vanishing gradient problem when trained on long time series data [157]. This problem arises when the gradients, which guide the weight updates, become exceedingly small as they are backpropagated through time. Second, RNNs are perceived as challenging to train and parallelize, discouraging researchers from their adoption due to computational constraints [158]. Hence, Long Short-Term Memory models (LSTMs) have been developed to overcome the problem of RNNs with long-term dependencies [159]. They introduce memory mechanisms that enable them to store information over longer sequences, making them particularly suitable for detecting anomalies in complex time series where unexpected events may occur after a long period. In [160], the use of LSTM neural network shows a stable classification behavior with a peak F1-Score of 80% and shows a superior performance compared to other models.

- **Autoencoders**

**Autoencoders (AE)** offer an interesting alternative to RNNs in the realm of anomaly detection within time series data [161]. The notion of employing autoencoders for detecting outliers stems from the empirical observation that outliers pose a more significant challenge for representation in a reduced feature space, which is the fundamental principle of dimensionality reduction [162]. Autoencoders are designed to reconstruct normal or inlier data effectively, but they tend to struggle when reconstructing abnormal or outlier data. Consequently,



**Figure 3-16: Architecture of Autoencoders**

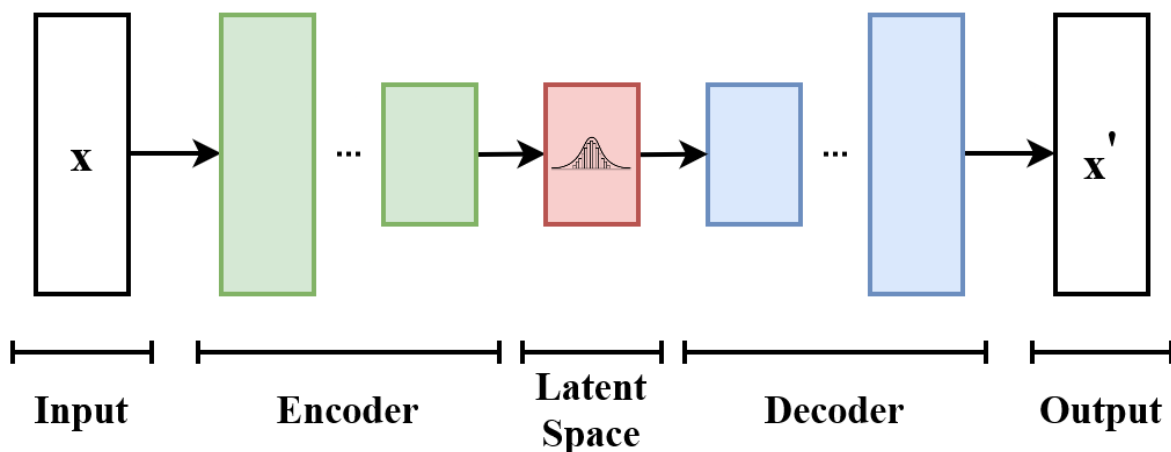
when autoencoders encounter outliers, the reconstruction errors become conspicuous. This can be understood as autoencoders attempting to compress input data into a smaller feature space, referred to as the latent space, which retains correlations among variables but cannot perfectly reconstruct the entire dataset (see [Figure 3-16](#)). This is due to the presence of multiple hidden layers acting as an information bottleneck, ultimately limiting the feature space [163]. The underlying assumption is that normal and abnormal data exhibit substantial differences in this feature space. Thus, projecting back to the original space will accentuate dissimilarities in certain data points, effectively highlighting anomalous instances.



Autoencoders are well-suited for anomaly detection, and can be applied to time series [164], [165]. The flexibility of autoencoders allows for various architectural combinations to suit the data's inherent characteristics. Researchers have found that hybrid architectures, which integrate convolutional and LSTM layers with autoencoders, can be particularly effective in identifying anomalies within the dataset [105], [118].

On the other hand, **Variational Autoencoders** (VAE) are an extension of conventional autoencoders. VAEs are data generation models that can learn to probabilistically represent the characteristics of time series [166]. The main distinction between VAE and AE lies in the fact that the VAE is a stochastic generative model capable of providing calibrated probabilities, while the AE is a deterministic discriminative model with no probabilistic basis [167].

VAE incorporates a probabilistic component through the distribution in the latent space (see [Figure 3-17](#)). In fact, this model has the ability to generate new realistic data by sampling in latent space, thus facilitating the identification of anomalies by comparing the observed data with those generated by the model. In addition, VAE models offer increased flexibility for anomaly detection as they are able to model various data distributions, including multimodal distributions. Several research papers have explored the combination of VAE and CNN demonstrating promising results [168], [117], by integrating the hierarchical representation capabilities of CNNs with probabilistic generation aspects of VAEs.



**Figure 3-17: The basic scheme of a variational autoencoder. The model receives  $x$  as input. The encoder compresses it into the latent space. The decoder receives as input the information sampled from the latent space and produces  $x'$  as similar as possible to  $x$ .**

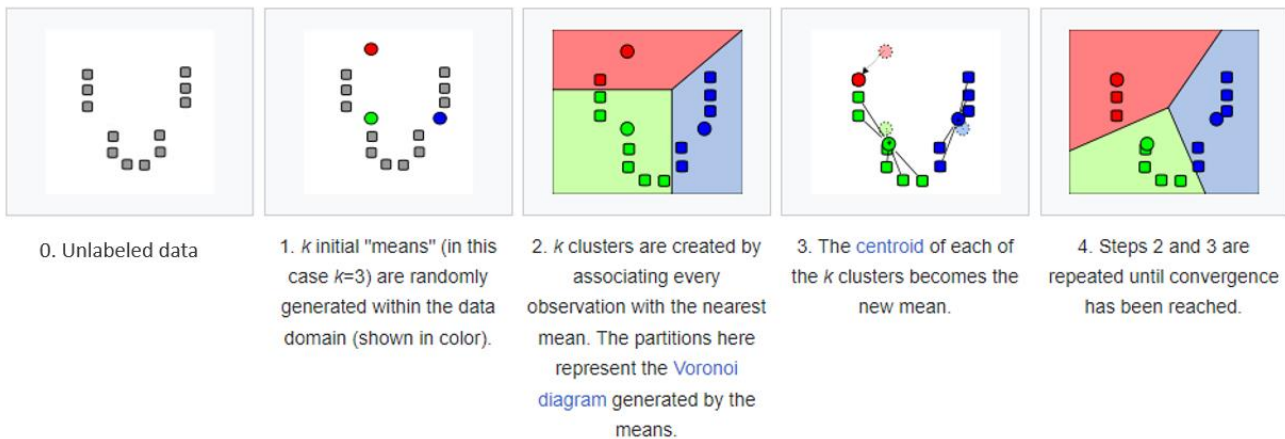
However, it is important to note that **these models are often more complex to implement than standard AEs**, which can make their use more demanding in terms of computational

resources and time. Due to their probabilistic nature, VAE results may be more difficult to interpret than AE results, which may complicate the understanding of detected anomalies.

### 3.4 AI-Powered Anomaly Detection: A Broader Outlook

In the field of anomaly detection within time series, many ML and DL models have been deployed to identify outlier data points / subsequences. We have looked at some of the most popular models in the field of urban hydrology, but it is essential to note that this field is evolving, and different models are emerging with more or less popular applications. Moreover, some models that have long been considered state-of-the-art for the detection of anomalies in time series have been abandoned to the detriment of other models.

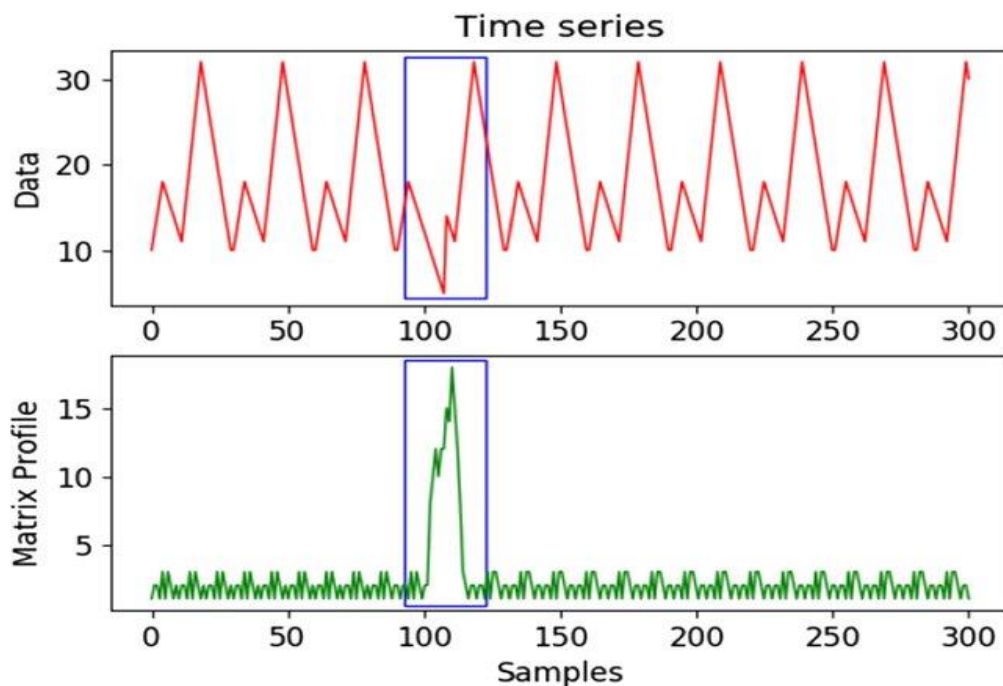
For example, one of the conventional clustering algorithms for anomaly detection is using **K-Means** clustering [169]. K-Means is an algorithm that groups data into homogeneous clusters based on similarities, using the Euclidean distance (see [Figure 3-18](#)). For anomaly detection, the idea is to group normal data into  $k$  clusters, thereby isolating data points that don't fit into any cluster or are far removed from existing clusters. The main inherent challenge of this approach is specifying an appropriate value  $k$ .



**Figure 3-18: K-Means Principle - © adapted from [170]**

However, [171] have demonstrated in their research that clustering time series sequences lacks meaningful significance. They provided evidence that the cluster centers obtained from multiple runs of the K-means algorithm on the same dataset do not exhibit significantly greater similarity to one another than cluster centers from a dataset generated by random walks. Numerous researchers have made mathematical analyses of this phenomenon [172], [173], [174] and many attempts have been made to address this issue or, at the very least, to identify time-series patterns that could be effectively clustered using K-Means [175], [176]. Nonetheless, the fundamental problems largely remain unresolved [177].

On the other hand, other models are now being used to detect anomalies in time series as a state-of-the-art approach, such as **Matrix Profile (MP)** [178]. This method focuses on identifying those parts of the data that have distinctly different characteristics from any other, making them potentially anomalous. The fundamental difference between Matrix Profile and many other anomaly detection approaches is that it has been specifically designed for time series data mining, incorporating an anomaly detection option from the outset. The MP algorithm was first developed in 2016 to address numerous time series data mining tasks such as anomaly detection and pattern identification [179]. Anomaly identification via MP is based on the matrix profile calculation, which annotates a time series with a vector of minimum Euclidean distances between each pair of subsequences in a time series. The subsequences with the greatest distances (framed in **Figure 3-19**) correspond to anomalies.



**Figure 3-19: Example of Matrix Profile for anomaly detection - © [180]**

In [181], the research conducts three distinct experiments under significantly varying conditions to illustrate the effectiveness of the MP approach. These experiments employ diverse datasets, including hydraulic simulations, power electronic converters, and cyber-security intrusion detection scenarios. Remarkably, the MP model demonstrates robust performance and high accuracy across these contexts with minimal parameter adjustments. In [182], the focus shifts to evaluating anomaly detection for abnormal heartbeats in ECG data. The results of this test are encouraging. Moreover, in [183], the study's findings suggest that, with straightforward parameter tuning, this detector delivers outstanding accuracy and performance across a spectrum of fault scenarios. To the best of our knowledge, this model has never been evaluated on urban hydrology chronicles, nor data from wastewater systems in particular.

### Chapter 3. Artificial intelligence – enhanced data validation framework

The field of DL evolves even more with new models and architectures that emerge regularly. For example, different variants of DNN, CNN, AE and RNN models were confronted and validated on a different public dataset's archives. When compared to seven deep learning architectures, and across a wide range of datasets comprising 98 different univariate and multivariate time series, the **Residual Network (ResNet)** model consistently outperforms the other methods [134]. This model belongs to the family of deep convolutional neural networks, that was introduced to solve the problem of "vanishing gradient". ResNets stand out for their innovative architecture that incorporates linear shortcut connections "skip connections" allowing information to cross layers more efficiently.

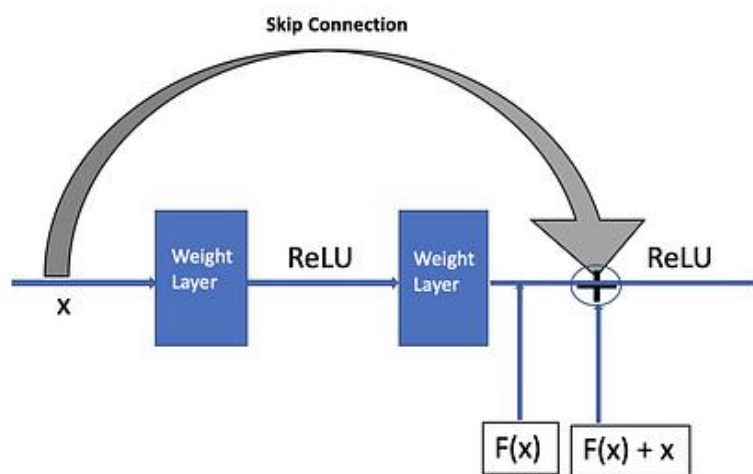


Figure 3-20: A residual block of the ResNet model - © [184]

Other approaches exist but are less frequently used for anomaly detection. For example, Generative Adversarial Networks (GANs) can serve this task, by generating synthetic data similar to real time series and identifying inconsistencies [185]. In addition to these methods, other innovative approaches are constantly emerging, including combinations of various models. It is therefore important to bear in mind that the field of time series anomaly detection is constantly evolving, with a wide range of methods to be explored to meet the specific needs of each application.

### 3.5 Synthesis of Chapter 3

The process of validating time series, particularly data from wastewater networks, involves three main phases. Firstly, given the temporal dependency of the data and the collective or contextual nature of the defects, it is essential to segment the data into **sequences using appropriate sizes** to capture the dynamics and seasonality of the data. These sequences are then subjected to artificial intelligence models for processing. Finally, a validity or invalidity label is assigned to the input data according to various operational considerations, which can be achieved at time step and/or sequence scale. Whatever the form of the model's output (binary labeling or anomaly score), the aim is to be able to switch from the time-step scale to the sequence scale and vice versa, depending on the needs and relevance of the approach.

The models used for data validation using AI in the field of urban hydrology can be categorized into three distinct classes: **supervised classification, unsupervised learning, and predictive models**. In the case of supervised classification, the scientific community increasingly favors DL models over ML models, because the latter are more demanding in terms of data quality and rigorous labelling. With respect to unsupervised learning, it appeared that the off-the-shelf models used in the field of urban hydrology may not be suitable for our case study due to the specificity of our data (highly dynamic and seasonal). However, new approaches such as Matrix Profile are emerging and may be of interest. Finally, in the last category, Recurrent Neural Networks and autoencoder models are used, although the latter are more popular, which may seem counter-intuitive in the processing of time series since the former are specific to this time of data. This preference is explained by the complexity and higher numerical needs of RNN compared to autoencoders.

## Synthesis of Part I

In the realm of wastewater systems, measured wastewater data diverges from typical temporal data due to factors such as significant dynamic and seasonal variations influenced by factors like weather conditions, time of day, and weekdays versus weekends. In addition to the unique characteristics of wastewater data, there is a multitude of potential defects, both in terms of significance (errors or non-representative data) and structure (point or sequential anomalies), especially regarding water quality measurement like turbidity. In addition, it is important to note that these defects remain by nature a minority and are scattered within a massive data stream. Nowadays, data validation in wastewater networks involves two stages: pre-validation, which detects basic anomalies, and supervised validation by an expert, which addresses complex issues. The manual approach carries the risk of subjectivity and human error, making automation a valuable alternative. Various automation methods have been examined in the literature, ranging from statistical models to model-based approaches, each with its own advantages and limitations. However, our aim here is to assess the effectiveness of AI-based models for carrying out this data validation task. A literature review of data validation using AI in urban hydrology categorizes models into three classes. In supervised classification, Deep Learning (DL) models are favored due to their higher accuracy compared to supervised ML models. For unsupervised learning, conventional models may not be suitable, but emerging approaches like Matrix Profile show promise. In the last category, autoencoder models are popular in time series processing.

**Table 3** proposes a comparison between the requirements related to the nature of the data to be processed and the capacity of each model to meet them. There are six such requirements, which can be formulated by answering the following questions:


- *Necessity of prior knowledge*: 'To what extent does this model require prior knowledge or priori information to perform effective data validation?
- *Dealing with time series dynamics*: "Is this model able to manage the complex temporal dynamics of the data and their different seasonality?"
- *Massive volume of data*: 'Is this model capable of efficiently processing very large amounts of data, or is it limited in terms of capacity?
- *Sensitivity to unbalanced data*: 'How sensitive is this model to unbalanced data, and how does it adjust its performance accordingly?
- *Assumptions on data distribution*: "Does the model make assumptions about data distribution?"
- *Detection speed*: "How fast is the model able to detect anomalies or defects in the data

Table 3: Analysis grid for different data validation tools / models

Criteria

	Necessity of prior knowledge	Dealing with time series dynamics	Massive volume of data	Sensitivity to unbalanced data	Assumptions on data distribution	Detection speed
<b>Manual validation</b>	Need of expertise			Important error risk		
<b>Statistical models</b>		Stationarity			Known distribution	
<b>Hydraulic modeling</b>						
<b>SVM</b>	Need of labels	Can be adapted to deal with time series			Two balanced and distinct classes	
<b>RF</b>						
<b>MLP</b>				Can be remedied		
<b>CNN</b>						
<b>OCSVM</b>						Majority and density-based approaches
<b>IForest</b>						
<b>LOF</b>						
<b>Matrix Profile</b>						
<b>RNN</b>						
<b>AE</b>						
<b>VAE</b>					Normal distribution	

Models



# **Part II**

## **Materials and Methods**



## ***Introduction of Part II***

The inherent complexity of wastewater data, related to the overlapping of several dynamics at different timescales and the influence of random rainfall events, creates unique challenges for validation. The previous section sheds light on the complexities arising from factors such as variable measurement frequency, dynamic seasonal variations and potential anomalies within the massive data stream. Aware of the limitations of manual validation and the shortcomings of traditional models (statistical and/or hydraulic), the adoption of automated solutions is revealing. This section lays the foundations for this analysis.

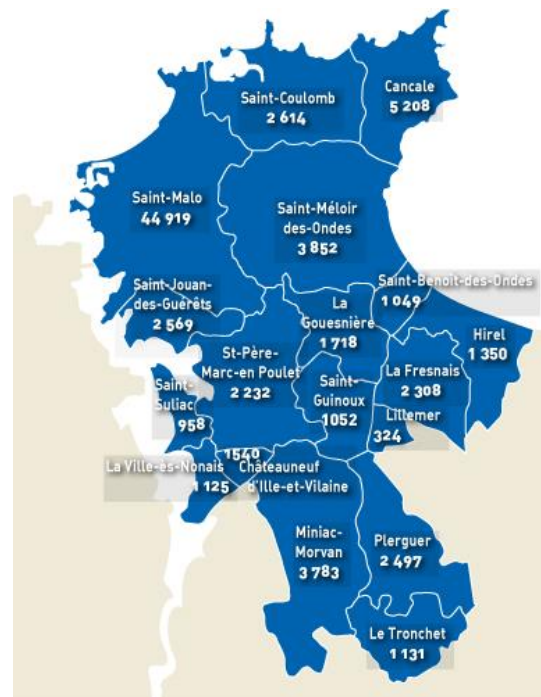
- In the absence of a public database, the first question is: **What database did we collect? And how did we go about collecting it, by examining its statistics and dynamics?**
- Faced with the multitude of available models and limited material and time resources, **which models were chosen for testing in this study? How do these models meet our key requirements, such as handling complex temporal dynamics, and what tests were carried out?**
- Finally, **what performance measures are used for evaluating the effectiveness of the different models and comparing them? What elements, ranging from the programming language to the hardware used, lie behind the scenes of AI models?**

# Chapter 4. AI's Backbone: Introducing our model evaluation database

Before implementing AI models for data validation in the field of urban wastewater, we face a major challenge : unlike fields such as image classification or intrusion detection, **it is difficult to obtain publicly available real-world datasets from urban wastewater utilities**. This difficulty stems mainly from concerns about confidentiality and current legislation. Our first hurdle, therefore, is to build a reliable, robust, and comprehensive database on which to base our future work on validating and evaluating AI models. This milestone was made possible thanks to the valuable contribution of **Saint Malo Agglomeration**, which generously provided access to their database. This collaboration allowed access to real operating data.

## 4.1. Introduction and background

In terms of infrastructure, the wastewater system of the city of Saint-Malo represents more than half of the assets managed by Saint-Malo Agglomeration (SMA). This wastewater system knows a significant presence of a combined sewer system, resulting in the establishment of multiple storm overflows. Within the municipality, there are currently thirteen storm overflows where the flow exceeds 120 kg BOD<sub>5</sub>/day, a regulatory threshold in France that requires instrumentation for discharge flow estimation. These overflows are equipped with sensors designed to provide estimates of the discharged volumes. Through this initial assessment, it was revealed that a noteworthy ninety-five percent of these volumes are concentrated around six key "interceptors" .



**Figure 4-1: Saint Malo Agglomeration and number of inhabitants per municipality**

In light of this insight, SMA decided to embark on a more extensive instrumentation effort concerning these six primary storm overflows. This instrumentation aims to evaluate the discharged pollutant flows. The main objective of this system is to adhere to the flow criterion

stipulated by the regulatory guidelines, where compliance is defined as not exceeding five percent of the pollution discharged in comparison to the collected pollution.

### 4.2. Sensors' network and data availability

#### 4.2.1. Data collection process

The instrumentation project of the Saint Malo interceptors started in tandem with the initiation of this thesis in February 2021. Consequently, **data collection occurred in synchronization with the project's progression**. Hence, the initial testing phases were limited by a smaller dataset spanning a minimum of 5 months. In contrast, the latter stages of evaluation benefited from a significantly larger dataset, encompassing up to 18 months of observations, starting from February 2021 and extending through August 2022. It's essential to acknowledge that the comparative analysis of the various algorithms carries a certain degree of bias due to the disparate input data used for evaluation. Initiating the tests with a limited dataset was imperative, given the impracticality of waiting for a complete database. For the sake of transparency and scientific rigor, we will explicitly specify the used dataset in all subsequent discussions related to the tested models. This practice ensures complete transparency regarding the data limitations and variations across the different testing phases. By clearly outlining the database's duration for each model, we aim to provide readers with a comprehensive understanding of the specific data conditions and constraints that influenced the model's performance and results.

The data collection process was conducted concurrently across all **six interceptors** (see [Figure 4-2](#)). Data is systematically archived by a SCADA system, facilitating record-keeping and ensuring long-term availability of historical data. Furthermore, a detailed logbook is carefully maintained by the network operator to document any interventions or maintenance activities carried out on-site. Particularly during the initial stages of sensor installation, it is not uncommon to encounter challenges related to data transmission and sensor functioning. Regular inspections and maintenance checks are pivotal in addressing these issues and fine-tuning the monitoring system's performance. This approach enables the adjustment of maintenance schedules based on the actual needs, ensuring that the data collection process remains reliable and uninterrupted.

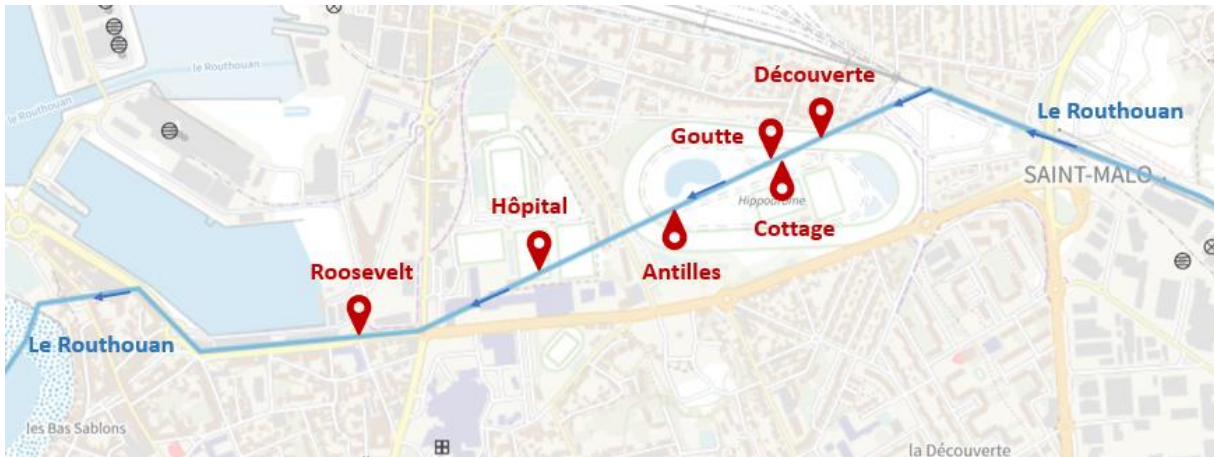


Figure 4-2: The six main interceptors of Saint Malo Agglomeration

### 4.2.2. An overview of sensors in place

The sensor field in Saint Malo is very varied with water level sensors, overflow detectors, automatic samplers, rain gauges and others. **Those of interest in this work are particularly the pollution sensors; namely turbidimeters and conductimeters.**

#### 4.2.2.1 Turbidity

The primary objective of turbidity measurement is to measure water transparency, or opacity. Under certain assumptions, an assessment of the concentration of particulate matter, and further of overall pollution parameters such as COD or BOD can be derived from this measurement. Typically, the turbidity signal exhibits significant dynamics, reflective of actual variations in effluent quality. Furthermore, measurement artifacts may perturb the signal, potentially resulting in an overestimation of the mean values (see [Section 1.4](#)). SMA has implemented **turbidity redundancy** as recommended in [37]. Redundancy enhances measurement reliability when both sensors provide consistent values and allows to question the accuracy of at least one of the probes (usually the one with higher values) in the case of disagreement.

The turbidity sensors used are of the Solitax SC type, manufactured by HACH LANGE. The measurement principle relies on the scattering of light at 90° with dual beams. The measurement range spans from 0.001 to 4000 FNU, with an accuracy of less than 1% or 0.001 FNU. Typically, in wastewater networks and due to operational constraints, point measurements are taken every 5 minutes. In the case of turbidity measurement, **the recording frequency is set at 5 minutes**, using a data acquisition strategy that computes average values over 5 minutes with a sampling interval of every 20 seconds.

Turbidity sensors demand a high level of maintenance and pose a challenge in terms of measurement verification. Dealing with sensor fouling is particularly complex since sensor

behaviors vary based on their location and the nature of the effluent. Multiple control tests and calibration procedures have been conducted in a preliminary phase. For instance, at the “Antilles” interceptor, turbidity sensors were lowered to mitigate observed grease deposits, leading to an improvement in data availability. Additionally, investigations into the frequency of brush sweeps for the turbidity sensors have been considered, with the optimal frequency identified as one sweep per hour. When a fully operational state is achieved, regular maintenance interventions are implemented, the frequency of which is tailored to each particular site

### 4.2.2.2 Conductivity

Conductivity is a property that characterizes the capacity of a material or liquid to conduct electricity. It is linked to the concentration of ions present in the liquid. Variations in electrical conductivity serve as indicators of the overall concentration of mineral dissolved matter in the wastewater network and can be employed for the detection of the intrusion of clear water, rainwater, seawater, or process water within the network.

The conductivity probes utilized in this system are HACH LANGE's 3798-S probes. These probes operate on the inductive measurement principle and offer a measurement range spanning from 250  $\mu\text{S}/\text{cm}$  to 2.5  $\text{S}/\text{cm}$ , with an associated uncertainty of  $\pm 1\%$  or  $\pm 0.004 \text{ mS}/\text{cm}$ .

The technology employed in conductivity probes is deemed reliable enough to counter the need for sensor redundancy. Consequently, one probe will be installed in each of the six interceptors. To ensure the measurement data's representativeness, the measurements will be recorded using the same method as employed for turbidity, by calculating the average of values collected over 5 minutes with readings taken at 20-second intervals. In general, these conductivity probes exhibit robust performance and do not demand excessive maintenance. However, it should be noted that erroneous readings may occur in the event of incorrect temperature calibration. Therefore, periodic calibration is required. Hence, regular maintenance is also scheduled to guarantee the proper functioning of the probes.

## 4.3. Data Exploration: Analyzing the Database

### 4.3.1. Data acquisition

Analyzing a measurement time sequence requires a structured approach, starting with an examination of the acquisition parameters. The first step is to identify the **start and end dates of the database**, delimiting the period covered by the measure. In our case, the sequence extends from February 1, 2021, to July 31, 2022, thus including a measurement period of significant duration (156 960 values per site).

The second crucial stage of the analysis focuses on **frequency exploration**. Although the theoretical frequency is set at 5 minutes, artifacts can occur. Among these, duplicates and non-constant frequencies stand out. In our case, there were 225 duplicates and 430 measurements where the frequency was less than 5 minutes. Out of 10 measurements, the frequency exceeds 5 minutes, half of which have intervals between 5 and 10 minutes, while the other half show more extended periods of lack, reaching up to an hour or even a full day (as on the 30<sup>th</sup> of June 2021). It is essential to note that these higher-frequency observations are considered as missing data, distinguished from timestamps with NAN values. The latter remain below 1% of the total acquired database, underlining the generally high acquisition quality of the data despite these variations in frequency. Missing data can result from various scenarios, such as technical problems during acquisition or planned interruptions. In our case study, we observe that these acquisition failures generally occur simultaneously across all sites. This synchronicity in acquisition failures reinforces the hypothesis of a central technical problem affecting the entire network.

Faced with these acquisition anomalies, measurement data requires some groundwork before it can be mapped by machine learning algorithms. The crucial issues are temporal regularity and dealing with missing data.

First and foremost, duplicate data must be eliminated to avoid any distortion in subsequent analyses. Data resampling with a constant time step is essential to homogenize measurement frequency. **Resampling** is then an important technique to ensure a round-the-clock frequency [186]. In addition, data synchronization is crucial to ensure temporal consistency, enabling accurate interpretation of events across the entire network and between the different sites. Consequently, before any modelling, we align the time series to **get fixed timestamps of 5 minutes**. Hence, the duplicated measures and the ones with a frequency less than 5 minutes are erased. On the other hand, the new fine-grained observations added during the disconnection period are filled in with zeros. These periods should be identified later as anomalies.

Another fundamental step in the data processing concerns the **imputation of missing values**. AI tools often struggle to process temporal sequences with gaps, which can compromise the performance of models. Consequently, they need to be replaced with judiciously chosen values before fitting a model. There are multiple imputation algorithms in the literature [187]. The objective here is not to do data reconstruction, nor to drown the missing data in the data stream. But our aim is to replace it so that the algorithms that are sensitive to it can run but still identify it as an anomaly later on. Hence, in our case, **missing data will be replaced by zeros**. The use of zeros preserves temporal information while clearly indicating points where no

measurements have been recorded, as zero is out of the range of physically possible values in the case of turbidity data within wastewater networks.

### 4.3.2. Understanding data statistics

The analysis of data statistics provides an in-depth understanding of the characteristics and inherent trends in a data set. By examining metrics such as mean, median, standard deviation and quartiles, we can gain insights into the distribution of the recorded values.

**Table 4** contains a compilation of descriptive statistics for the T1 and T2 variables at the various study sites, whose names are specified at the top of the table.

**Table 4: Turbidity data statistics. The unit of all values is FNU**

	Antilles		Cottage		Découverte		Goutte		Hôpital		Roosevelt	
	T1	T2	T1	T2	T1	T2	T1	T2	T1	T2	T1	T2
<b>Mean</b>	171	171	99	115	71	66	151	138	29	23	135	151
<b>Std</b>	285	284	107	150	141	133	257	232	97	78	97	129
<b>Min</b>	0	0	0	0	0	0	0	0	0	0	0	0
<b>Max</b>	6954	6224	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
<b>25%</b>	21	20	43	43	21	21	26	27	5	5	79	83
<b>50%</b>	46	43	81	83	34	33	53	55	8	8	123	129
<b>75%</b>	139	142	123	130	53	51	119	117	16	16	164	176

The analysis of these descriptive statistics reveals some interesting trends in the collected data. In particular, the high standard deviation for the "Antilles" and "Goutte" sites indicates significant variability. For "Antilles", outliers are observed at the maximum value, which is greater than 6000 FNU, given that the sensor's upper range is 4000 FNU. These outliers have a strong influence on the mean, but little impact on the median, which is much lower. For "Goutte", the mean is also significantly higher than the median, which may indicate the presence of a number of extremely high values where the mean is greater than the 75<sup>th</sup> quantile. Turbidity measurements at the "Antilles" and "Goutte" interceptors posed several challenges during operation. The "Antilles" site was confronted with remote operating problems, leading to signal alterations between the probe displays and data transmission to the supervisory level. An update of the system solved these problems and ensured reliable data transmission. In addition, as previously mentioned, the "Antilles" interceptor was affected by grease deposits, requiring the probes to be repositioned. Furthermore, the "Goutte" interceptor regularly shows measurement alterations lasting 2 to 3 days (see [Figure 4-3](#)). These interruptions can be attributed, according to the logbook, to interventions for

construction work carried out at this level, thus explaining the dropouts and gaps observed in the data.

The mean turbidity is very low at the "Hôpital" interceptor, at around 30 FNU, given that turbidity in sewage networks varies between 25 FNU and 500 FNU. Indeed, the hydraulic analysis conducted at this site reveals a predominant flow of clear spring water during dry periods with peaks of pollution originating from upstream collectors during rain events. Consequently, these peaks reach the upper limit of the sensor's range.

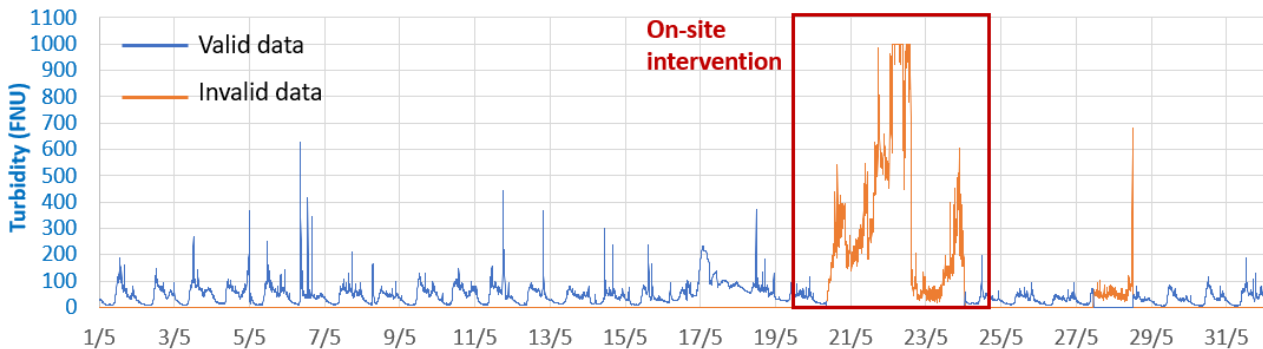


Figure 4-3: Example of defects at the interceptor "Goutte"

On the other hand, correlation analysis provides an overview of the linear relationships between the T1 and T2 variables at the different sites. The correlation coefficient used here is Pearson's coefficient. A value close to 1 indicates a strong positive correlation, while a value close to -1 indicates a strong negative correlation. A Pearson coefficient close to 0 suggests a weak linear correlation.

**Equation 2: Pearson Correlation Coefficient**

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Where: X, Y are two variables with n individual sample points  $x_i$  and  $y_i$  respectively.  $\bar{x}$  is the sample mean of X, and analogously for  $\bar{y}$ .

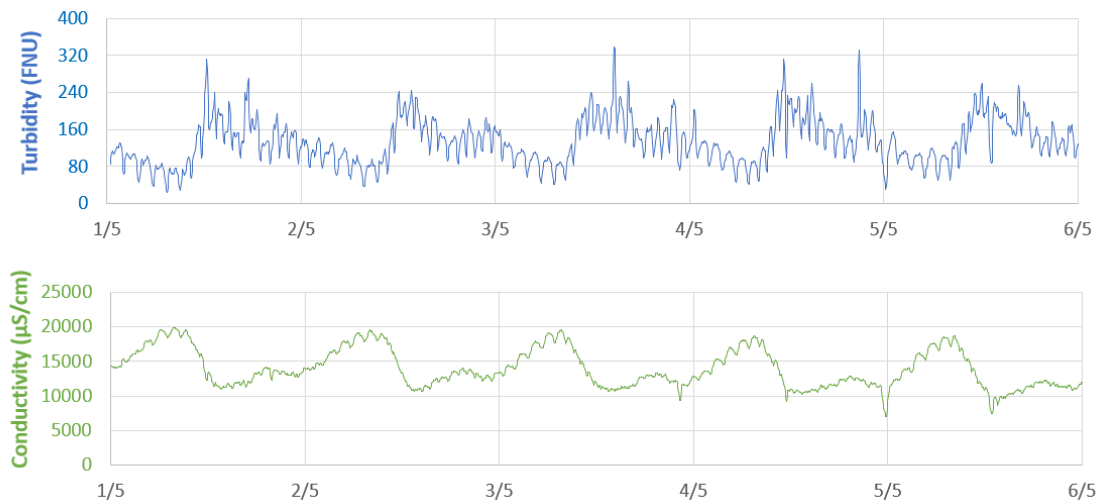
An analysis of the correlation matrix in **Table 5** reveals several observations. Firstly, the strongest correlations are observed between variables T1 and T2 of the same site since both are supposed to measure the same phenomenon. However, it is interesting to note that this correlation is far from perfect. Indeed, slight differences between both sensors can arise from short range variations of water quality. Larger differences denote a failure of at least one sensor, hence the need for data validation, which is not excluded by hardware redundancy.



Table 5: Pearson Correlation Matrix

		Antilles		Cottage		Découverte		Goutte		Hôpital		Roosevelt	
		T1	T2	T1	T2	T1	T2	T1	T2	T1	T2	T1	T2
Antilles	T1												
	T2	0.61											
Cottage	T1	0.09	0.09										
	T2	0.06	0.11	0.41									
Découverte	T1	0.35	0.28	0.09	0.07								
	T2	0.35	0.31	0.06	0.07	0.66							
Goutte	T1	0.27	0.28	0.10	0.15	0.32	0.31						
	T2	0.21	0.22	0.11	0.18	0.31	0.30	0.61					
Hôpital	T1	0.00	-0.03	0.05	0.05	0.02	0.00	-0.01	0.02				
	T2	-0.03	-0.05	0.07	0.05	0.00	0.00	0.01	0.03	0.49			
Roosevelt	T1	0.09	0.08	0.20	0.16	-0.01	0.02	0.10	0.10	0.00	0.02		
	T2	0.03	0.04	0.17	0.16	-0.02	0.00	0.09	0.11	0.01	0.03	0.41	

As for the relationships between the various sites, some correlations stand out. For example, correlations close to zero between "Hôpital" and the other sites indicate weak or non-existent relationships. This confirms the particularity of this site, which does not carry the same type of effluent. Weak correlations are also observed between "Roosevelt" and the other sites. Statistical analysis of "Roosevelt" data suggests that we have the same effluent quality as the other sites. For example, average turbidity is almost equal to that at "Goutte". But in fact, this weak correlation is due to the different dynamic of this site. In fact, there is a pumped discharge of seawater at this location, causing regular fluctuations throughout the day. These cycles determine the dynamics of the data, marked by regular oscillations (see [Figure 4-4](#)). Moreover, the site displays high conductivity, approaching that of seawater (50,000  $\mu\text{S}/\text{cm}$ ). This can be attributed to seawater intrusion, but does not clearly affect the dynamic pattern of turbidity.



**Figure 4-4: Data structure at the "Roosevelt" interceptor**

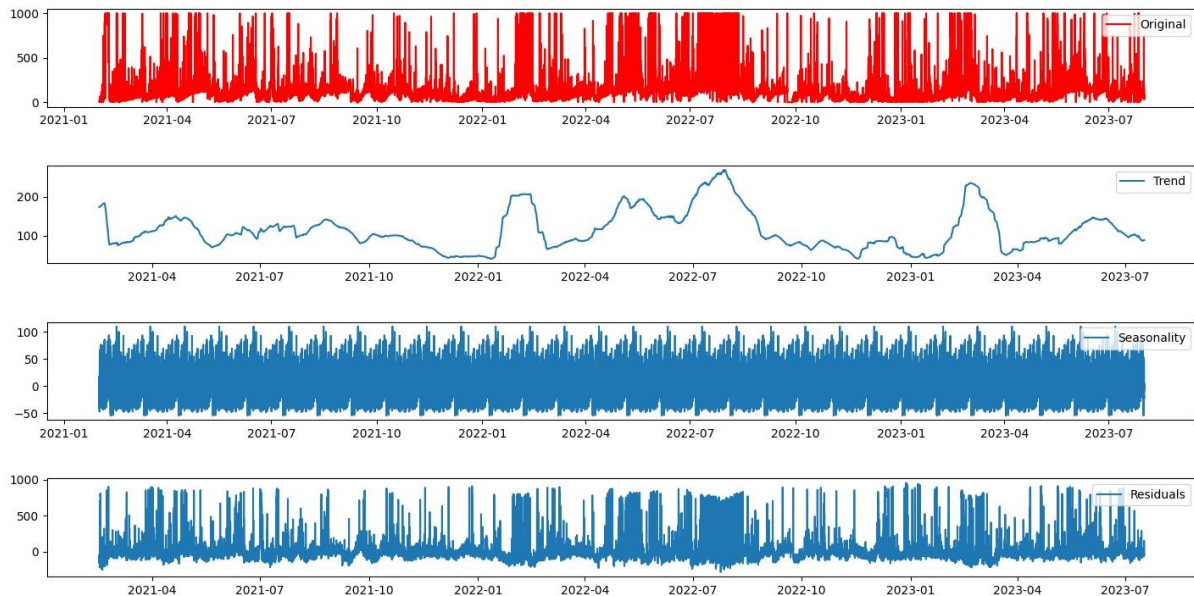
### 4.3.3. Understanding data dynamics

Before using any time series dataset, it is important to explore it, analyze its pattern and identify its underlying process. Time series are likely to be characterized by commonly observed structures:

- **Trend** is a gradual increase/decrease in data values as time passes starting from any point in time. A time series has a trend if its mean value is not constant but decreases or increases over time. The trend can be linear or not.
- **Seasonality** is a periodic structure, which oscillates around the general trend in a regular manner.

Decomposing a time series consists in separating its initial series into simpler subseries, each representing an essential aspect. A typical decomposition is the decomposition into 3 series: trend, periodic and residuals. The original series is found if we sum or multiply the 3-component series (see [Appendix F](#)). [Figure 4-5](#) shows the decomposition of turbidity data at "Cottage". A similar pattern is observed for turbidity data from other sensors in different localities. The first observation reveals a non-constant, non-regular trend, with no absolute direction of increase or decrease. This feature suggests variability and irregularity in the overall evolution of the data over time, qualifying the trend as "non-monotonic". This non-monotonicity indicates a complexity in the underlying patterns of temporal data, with fluctuations, frequent changes of direction, and non-linear patterns that cannot easily be characterized by a simple trend. The seasonality graph is not very explicit due to the presence of different seasons in the same chronicle, both daily and seasonal. What's more, the residuals are not random. The presence of an irregular component in a time series generally suggests that certain influences or variations cannot be explained by the seasonal or trend components. Analysis of a non-

monotonic trend and non-random residuals may require the use of advanced time series analysis techniques or more flexible models capable of capturing complex temporal patterns.



**Figure 4-5: Turbidity data decomposition**

A time series is considered stationary when its mean, standard deviation, and autocovariance remain constant, and there is an absence of seasonality. In our case, the series exhibits the presence of a significant trend and multiple seasonality, leading us to conclude that it is non-stationary. We assume then that statistical models cannot be used to handle these time series since their basic hypothesis is not valid.

### 4.4. Data validation

In the vast panorama of literature devoted to wastewater data, it is striking to note a clear deficit in terms of formalization of pollution data validation, particularly in comparison with other fields such as air pollution [188]. This is the background of our work, whose first mission was to develop an expert validation methodology specifically dedicated to turbidity data. The major advantage lies in the possibility of exploiting physical redundancy, thus providing a basis for analyzing the reliability of information collected on site. Although the essence of this thesis lies in the application of AI to data validation, it remains crucial to establish a concrete baseline against which to assess the performance of the model thus developed.

#### 4.4.1. Manual data validation processus

Each interceptor is equipped with 7 sensors, which are interrogated remotely by the LERNE supervisor. The LERNE supervisor records the raw instantaneous values every 20 seconds and calculates various derived quantities. These include the 5-minute averages of each

measured parameter, as well as quantities combining information from several sensors, such as the discharged flow rate (see Figure 4-6). In our study, we focus on four derived quantities: the 5-minute averages of turbidity measured by each sensor, conductivity, and spill flow. At the same time, data is collected from 6 rain gauges, although these were not used in the validation process. In fact, conductivity and overflow are direct indicators of the impact of rainfall events on network operation and were used to validate turbidity data.

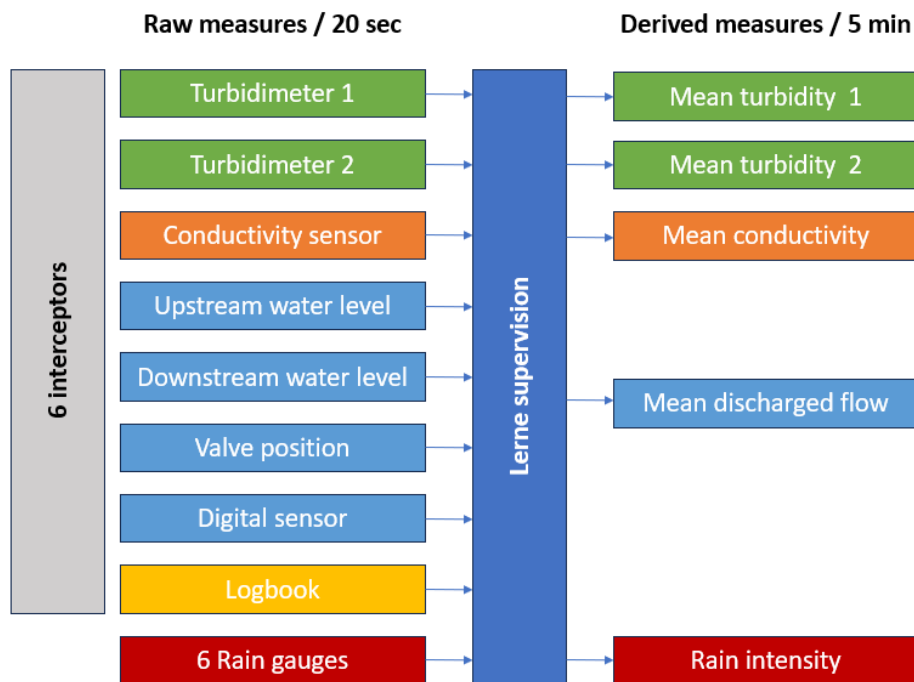


Figure 4-6: Data acquisition process for variables of interest

Manual validation is carried out systematically every month, covering all data recorded the previous month. This procedure, which mobilizes an expert on average 2 hours per site each month<sup>7</sup>, focuses mainly on turbidity sensors. In fact, conductivity measures are generally very reliable, and their validation is limited to the identification of missing data, which is often consistent with turbidity data. Regarding overflow measurements, the calculation method has been validated by a specific study. Radar level sensors, which are a priori reliable, are subject to validation to identify missing data, which may be associated with spill periods. Parameterization errors, such as reverting to a previous configuration following a power failure, or the failure of a valve position sensor, can distort flow calculations, although the identification of spill periods remains intact. An overall check of the volumes discharged by each interceptor

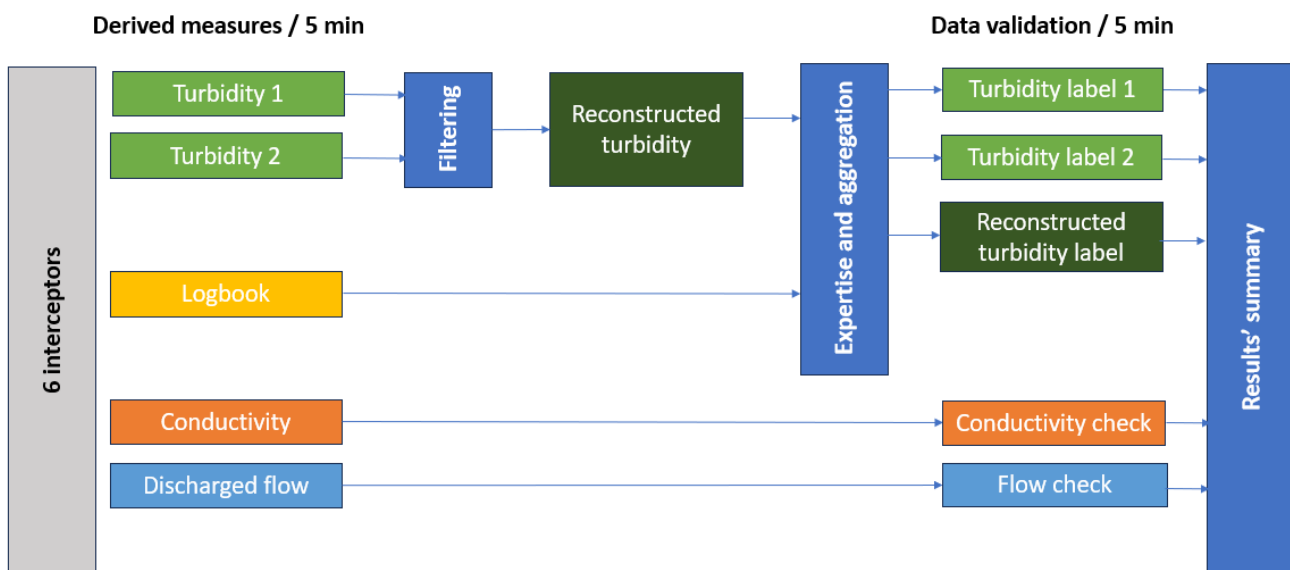
<sup>7</sup> This is an average for a process that takes between 1 and 4 hours, depending on the required degree of traceability and precision.

each month enables us to detect such anomalies, which are corrected by recalculating the data.

As turbidity data validation is concerned, three distinct stages are involved (see **Figure 4-7**):

- **Filtering:** consists of automatic validation of consistent data by redundancy, accompanied by the identification of periods requiring manual validation.
- **Expertise:** involves manual validation or invalidation of the lowest of the two since the majority of turbidity measurement disturbances are due to parasitic occultation of the light beam, resulting in increased turbidity
- **Automatic aggregation** of neighboring faults to obtain continuous periods of malfunction rather than a succession of multiple faults interspersed with short periods of apparent good working order

An Excel macro application has been developed to execute these three phases and synthesize the results. **In the remainder of this manuscript, we will refer to this process as manual validation, although there is an automated part based on hardware redundancy.**



**Figure 4-7: Organization of the validation process**

For the first phase, the consistency criterion used for automatic validation is defined as follows:

**Equation 3: Consistency criterion for turbidity validation based on redundancy**

$$| T1_{mean}(i - 6, i + 6) - T2_{mean}(i - 6, i + 6) | < \max(S_1 \times \min(T1_{mean}(i - 6, i + 6), T2_{mean}(i - 6, i + 6)), S_2)$$

## Chapter 4. IA's backbone: Introducing our model evaluation database

Where  $T1_{\text{mean}}(i-6,i+6)$  represents the sliding average calculated over a window of 13-time steps (i.e., 65 minutes) centered on the time step  $i$ , applied to the measurements taken by sensor 1. This inequality stipulates that the sliding averages from sensors 1 and 2 must satisfy the least restrictive of two criteria: a criterion expressed as a percentage of the smaller of the two measured values, defined by a threshold  $S_1$ , or a criterion expressed in absolute values, defined by a threshold  $S_2$ . By default,  $S_1$  is set to 10%, and  $S_2$  to 10 FNU, although these thresholds can be adjusted for each interceptor. If the inequality is satisfied, the central values of the 65-minute window  $T1(i)$  and  $T2(i)$  are validated, and the reconstructed turbidity value  $T(i)$  is equal to the mean of  $T1(i)$  and  $T2(i)$ . If the inequality is not satisfied, for example if  $T1_{\text{mean}}(i-6,i+6) - T2_{\text{mean}}(i-6,i+6) > S_2$ , the larger of the two measures, here  $T1(i)$  is invalidated, and the smaller, here  $T2(i)$ , is submitted to the expert for validation. The reconstructed value will be equal to the latter.

For phase 2, the expert visually examines monthly chronicles (Turbidity 1, Turbidity 2, Reconstructed turbidity, Conductivity, Discharged overflow), zooming in systematically on a weekly and daily basis. This examination can detect the following configurations:

- Zero or "blocked" values (= no variations over several hours)
- Non-consistent daily patterns during dry weather periods,
- Excessive noise (short-term variability) compared with usual behavior
- Unusual appearance and/or amplitude of rainy weather peaks
- A qualitatively homogeneous behavior of the two sensors, even if the numerical consistency criteria are not met
- A significant drop in turbidity, often following a rain event or a cleaning operation (identified through the logbook), which suggests that what precedes this drop is invalid

For the third phase, two aggregation criteria are used:

- For successive faults affecting the same sensor, a duration criterion is applied, set at 4 hours by default and adjustable for each interceptor and each period. Aggregation takes place simultaneously with detection, and it therefore aggregates potential faults that are submitted for assessment.

- For faults affecting the same sensor or not, post-processing of the results validated by expertise is based on a relative duration criterion: two faults are aggregated if the duration separating them is less than  $x\%$  of the duration of the longer fault. By default,  $x$  is set at 20%, but it is frequently adjusted according to the results obtained for a given interceptor and month chronicle.

At the end of this process, each turbidity value reconstructed at a 5-minute time step is assigned one of four attributes: Validated by redundancy (R) / Validated by expertise (V) / Invalid (I) / Missing (M). The label of the reconstructed turbidity depends on the labels of the raw turbidity values T1 and T2, with the potential combinations in [Table 6](#).

**Table 6: Labels of the reconstructed turbidity T, considering the labels of T1 and T2**

		T1			
		Redundancy	Validated	Invalidated	Missing
T2	Redundancy	Valid (R)			
	Validated			Valid (V)	Valid (V)
	Invalidated		Valid (V)	Invalid (I)	Invalid (I)
	Missing		Valid (V)	Invalid (I)	Invalid (M)

This validation process presents a harmonious approach, using automation while preserving the essential role of human expertise. The automated part, integrating physical redundancy, enables substantial time savings. On the other hand, expert validation, although it quickly excludes trivial anomalies such as zero values, saturation, or out-of-limits, maintains a human dimension necessary to apprehend the diversity of the most frequent faults. Although certain steps could potentially be automated, the decision to submit them to the expert reflects a desire to benefit from his or her discernment in order to obtain a clear view of the spectrum of defects. However, it is important to note that the nature of defects such as bias, drift and lack of precision unquestionably requires human intervention and expertise. This phase, while essential, also exposes the process to potential human bias and error, particularly in the manual splitting of defect periods and the manual modification of their beginnings and/or ends.

#### 4.4.2. Anomalies qualification

[Figure 4-8](#) provides a visual representation of several crucial aspects of data quality in our database. Firstly, it highlights the proportion of consistent data, ranging from 38% to 72%, opposed to invalid data, ranging from 7-11%, to 20-25%, and up to 45%: the “Antilles” and “Goutte” localities show a higher percentage of invalid data. This trend can be attributed to recurring failures, as mentioned in [Section 4.3](#), underlining the importance of expert validation to improve data reliability in these specific contexts. Secondly, [Figure 4-8](#) shows the proportion of data recovered through manual validation, ranging from 17% to 33%, which is far from negligible.

A crucial step in preparing the database for use by artificial intelligence models is to convert the various labels into a binary classification. In this way, each observation is categorized into

## Chapter 4. IA's backbone: Introducing our model evaluation database

two distinct classes. Class 0 groups together valid data, without distinguishing between those validated by redundancy or expertise. Class 1, on the other hand, includes data that is invalid, either as an output of the validation process or due to its absence.

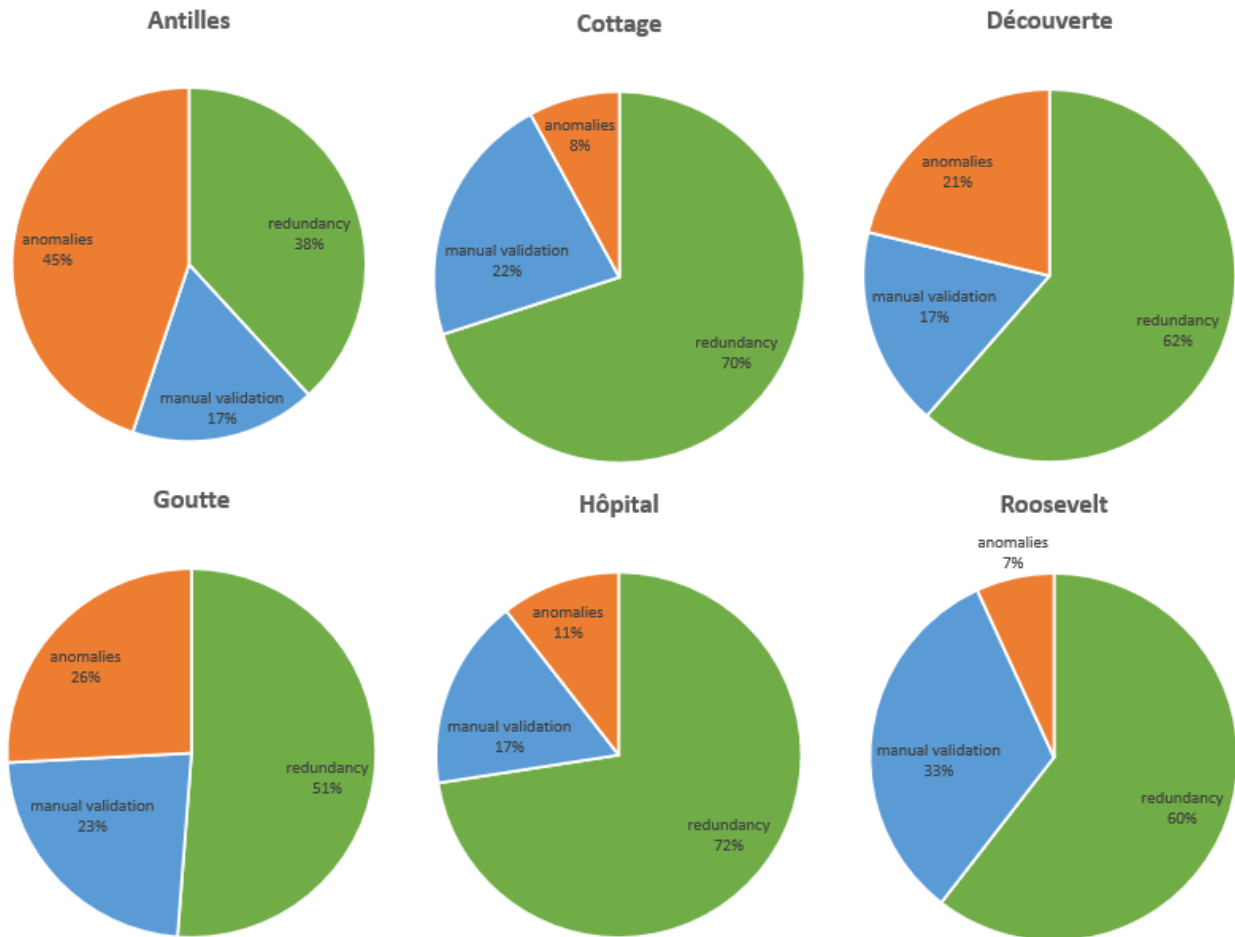
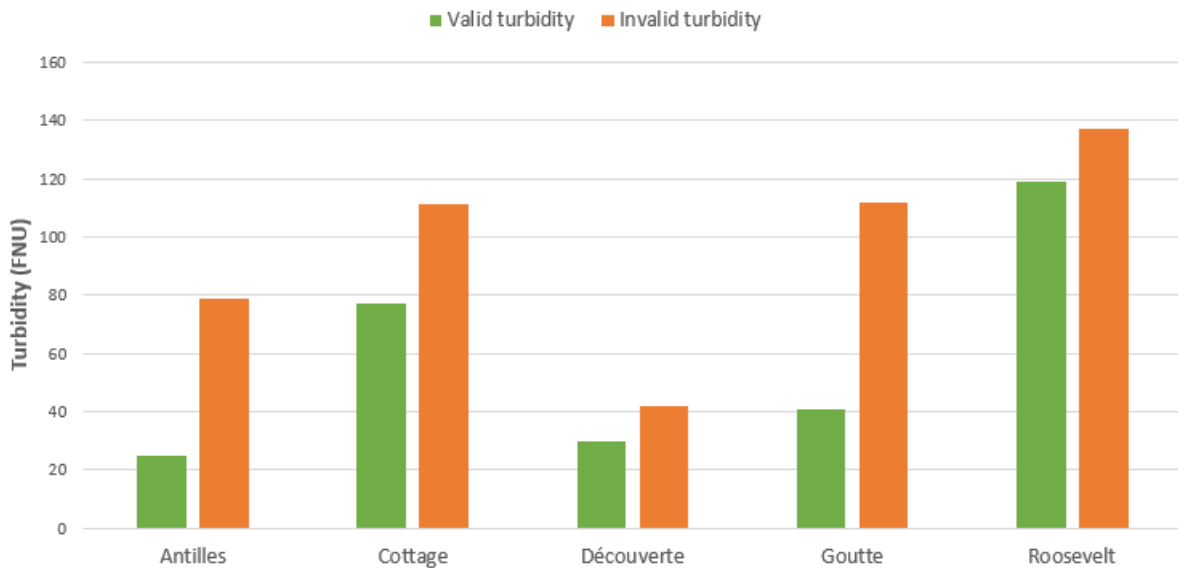


Figure 4-8: Results of manual validation.

When we examine the mean turbidity for both valid and invalid data (see Figure 4-9), we find a significant discrepancy between the two categories, with higher values associated with invalid data. This observation could be expected, as a main cause of failure is clogging, which results in less light collected by the receiving cell and an apparent increase of turbidity, and often saturation of the sensor. The discrepancy is particularly pronounced at the “Antilles” and “Goutte” sites, where there is a prevalence of outliers with large amplitudes. For the other sites, on the other hand, the deviation is more moderate, indicating that the anomalies are more structural in nature than in amplitude. These anomalies are therefore more subtle. In the ongoing study, **the “Cottage” site was chosen as the reference site for the evaluation of the various models.** This selection is explained by the fact that the site's hydraulics are typical, with a turbidity range in line with the average for sewer systems. In addition, the presence of various faults requires in-depth expertise to identify them, making Cottage a representative



and relevant choice for model analysis. Within this site, fault occurrences range from 10 minutes to 7 days. **The average fault duration is 17 hours, with a median of 4 hours.**



**Figure 4-9: Average turbidity at different sites, differentiating valid and invalid data.**

### 4.4.3. Mitigating subjectivity by organizing a validation pool

Recognizing the inherent risks associated with subjectivity and human error, it has been imperative for us to quantify this bias in order to gauge its impact on the evaluation of AI models. This problem is not specific to wastewater network data. Indeed, annotators are rarely in complete agreement when expressing their opinion, and this disagreement can be characterized as bias, the tendency of an annotator to prefer one decision to another, and variance, the natural variation between one annotator and another (or themselves at a later date) [189]. In view of the cost and complexity of obtaining a gold standard (GT), **it is commonly accepted that the opinion of one (or some) annotator(s) is close to this reference (GT)** [190]. Nevertheless, it is important to ensure that this assumption is valid in our case.

To address this, we have established a **validation pool** dedicated to validating turbidity data. This approach aims to enhance the transparency and reliability of our assessments, ensuring a robust understanding of the interplay between expert's baseline validation and the performance of AI models.

The validation pool relies on a diverse team of **four experts**, including a professional in turbidity data validation who contributed to the development of the reference guide [37], as well as two final-year interns from an engineering school specializing in water and environment.

The latter underwent a month's training, raising their awareness of turbidity data dynamics and potential faults. I was also part of the team, bringing my experience in data validation from my previous work on the Saint Malo project and my thesis. Due to time constraints, the multi-validation was carried out at **four different sites over a six-month period**, covering the whole year and its various seasons. The months selected were carefully chosen, avoiding the first months of installation to eliminate any bias linked to sensor optimization. All the experts work on the basis of the output of the Excel macro file filtering phase, intervening specifically during Phase 2 to bring their expertise to the validation of doubtful sequences.

In setting up this validation pool, we assume that annotators are not malicious in producing their annotations, that they don't produce annotations randomly, and that they don't simply follow low-level cues, but are able to mobilize higher-level knowledge. This enables them to distinguish sequences belonging to the positive class (anomalies) [191]. The different approaches for comparing experts and their agreement rates will be studied later in this manuscript (see [Chapter 7](#)).

### 4.5. Synthesis of Chapter 4

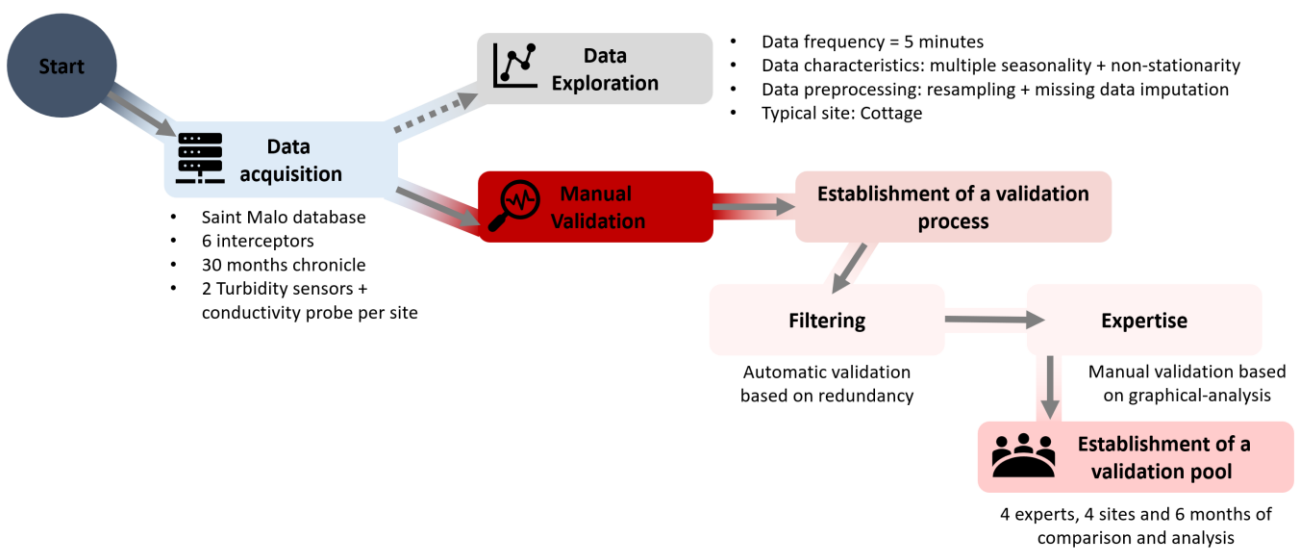
In the absence of a reliable public database, this chapter focuses on the challenges and methodological approaches associated with data collection and validation in the field of urban wastewater, based here on instrumentation from Saint-Malo. The study database comprises six interception sites, each equipped with two turbidity sensors and one conductivity sensor. Instrumentation was initiated simultaneously with the launch of the thesis, enabling data to be collected gradually as the project progressed. Considerable effort has gone into the operational monitoring of these sites, including regular maintenance, both curative and preventive, to ensure the reliability of the data collected. The measurement period runs from February 1<sup>st</sup>, 2021, to July 31<sup>st</sup>, 2022, with a frequency of 5 minutes. However, examination of the frequency reveals artifacts such as duplicates and irregular frequencies. To compensate for these anomalies, a **data pre-processing step is set up to resample the data and fill in missing data with zeros** so that it can be identified later. Statistical analysis of the data reveals the **non-stationarity of the time series**, characterized by an irregular trend and multiple seasonality.

Secondly, to evaluate the performance of AI models effectively, it is crucial to have a baseline against which to compare the results of different models. Therefore, we begin by developing an expert methodology specifically designed for validating turbidity data. This data validation process consists of three distinct stages, from automatic filtering to expert validation and automatic aggregation of anomalies. This approach leverages the inherent physical

## Chapter 4. IA's backbone: Introducing our model evaluation database

redundancy, and a consistency criterion is defined for automatic validation. Manual expertise focuses primarily on human intervention and expertise. **This phase exposes the process to potential human bias and error.**

Aiming to evaluate subjectivity and human error risks in data validation, a quantitative assessment was deemed essential to evaluate the potential variability among different annotators. Acknowledging the complexity and cost of obtaining a gold standard, **a validation pool was established, comprising four diverse experts.** This approach was taken to enhance transparency and reliability, ensuring a robust understanding of the interplay between expert baseline validation. Multi-validation was conducted at four different sites over six months, covering various seasons, and experts worked based on the output of the filtering phase. Assumptions included annotators' non-malicious intent, non-random annotations, and the ability to mobilize higher-level knowledge for distinguishing between sequences belonging to the positive class. Future chapters will delve into different approaches for comparing experts and their agreement rates.



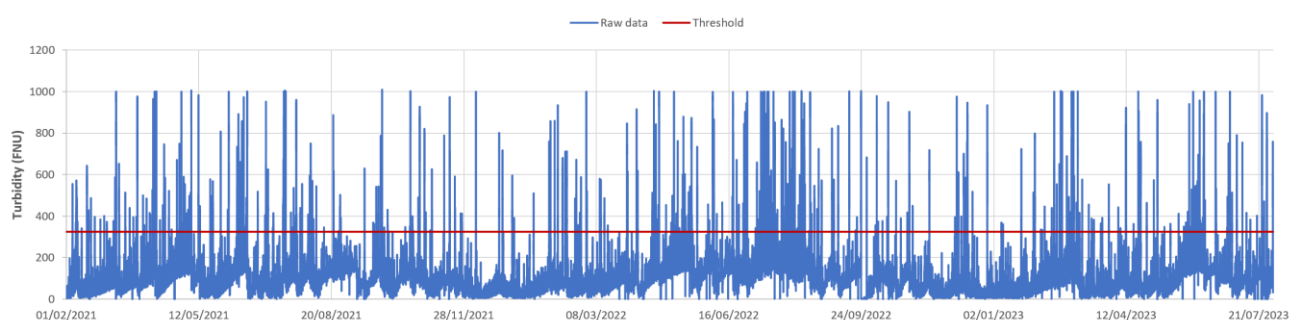
**Figure 4-10: Overview of database preparation for models' evaluation**

# Chapter 5. Benchmarking Models for Data Validation and Anomaly Detection

The main objective of this thesis is to leverage AI models for the validation of data from wastewater networks, focusing on turbidity chronicles at SMA interceptors. In this chapter, we delve into the conceptual fundamentals of the promising models which emerge from the literature review (see [Chapter 3](#)) and outline a testing methodology. This section serves to introduce these models, explain their core concepts, and articulate how they can be deployed to address the challenges inherent in our specific research problem. By providing a thorough understanding of the models and their potential applications, we pave the way for an exploration of their performance in subsequent sections.

## 5.1 Does our data justify the use of AI approaches?

In a context where AI is gaining ground and can be used for almost anything and everything, it becomes imperative, in the frame of this research, to question whether the deployment of AI tools is justified by the complexity of our test database. [192] has highlighted one of the flaws in public datasets, which is the simplicity of the task. Hence, it is crucial for us to examine whether our testing baseline (database + task) warrants the sophistication of AI tools. In this context, testing basic approaches such as 3-sigma allows to gauge how effectively we can address the intricacies of our problem (see [Figure 5-1](#)). This approach ensures that we do not succumb to the allure of complexity when simplicity might be the key to understanding and solving our challenges.



**Figure 5-1: Application of the 3-sigma rule to turbidity data in Cottage**

**Table 7** summarizes the results obtained by invalidating all data exceeding the mean plus 3 times the standard deviation for each respective site. This methodology eliminates a small number of outliers. A comparison between the first column, representing the actual ratio of invalid data, and the second column, illustrating the ratio of invalid data according to the 3-

sigma rule, reveals a clear disparity. This could be expected, as the 3 sigma rule is designed to discard only 0.3% of data for a centered gaussian distribution, and 11% at maximum in any case (Bienaymé-Tchebychev inequality). What's more, in some cases, such as Cottage, Hôpital and Roosevelt, a significant proportion of invalidated data (issued from Column 2) corresponds to false alarms, meaning intrinsically valid data (Column 3). This observation is explained by the significant presence of drift faults and noisy data for these sites. It may be noted that our Cottage type-site is home to a substantial number of subtle defects which do not manifest themselves in an extreme manner. By contrast, Antilles, for example, displays a predominance of saturation faults (non-gaussian distribution with many extremely high values).

**Table 7: Results of the 3-sigma rule for anomaly detection**

Site	Real anomaly ratio in the database	Anomaly ratio detected using the 3-sigma rule	Rate of false anomalies detected using the 3-sigma rule
Cottage	7.9 %	1.1 %	56.7%
Antilles	44.8%	5.6%	0.6%
Découverte	21.2%	2.2%	5.6%
Goutte	25.8%	4.1%	4.2%
Hôpital	10.6%	1.0%	29.8%
Roosevelt	6.9%	1.0%	68.1%

It is therefore concluded that in our evaluation data set, the presence of defects is far from the trivial anomalies easily detected by a statistic based on the 3-sigma rule. This finding justifies the use of this database and the deployment of sophisticated tools, for instance AI models.

## 5.2 Benchmark of models and tests

The literature review (see [Chapter 3](#)) revealed several potentially interesting approaches for anomaly detection in wastewater network data. Due to time constraints, we selected few models we consider most promising beforehand, in order to explore them in detail. This selection is strategic, encompassing a variety of approaches, combining both Machine Learning, Deep Learning, supervised and unsupervised techniques.

Supervised ML models, often sensitive to input data with learning limitations, have been largely replaced by their unsupervised counterparts according to the literature. Hence we will not investigate these models. However, we consider that they could be relevant in settings where dynamics are controlled (laboratory data, error simulation, ...). So, within the framework of

## Chapter 5. Benchmarking models for data validation and anomaly detection

supervised models, we are focusing on exploring neural networks, with particular attention paid to CNNs. According to a recent study [134], **ResNet** outperforms other CNN architectures, justifying our choice to explore it for our case study.

However, in the wastewater domain, obtaining labels for training models is arduous, often limited and potentially biased by human subjectivity. Therefore, exploring unsupervised AI models is essential. Within the framework of traditional ML methods, we found that state-of-the-art approaches proved unsuitable for our specific context [17]. This inadequacy stems from the absence of a clear majority class in our data and the complexity associated with defining a density in the presence of temporal correlations. However, among these methods, one model caught our attention, namely the **Matrix Profile**, recognized as a state-of-the-art approach to time series analysis and anomaly detection. Significantly, to the best of our knowledge, this model has never been applied to our specific field of study, which raises particular interest in its evaluation in our context.

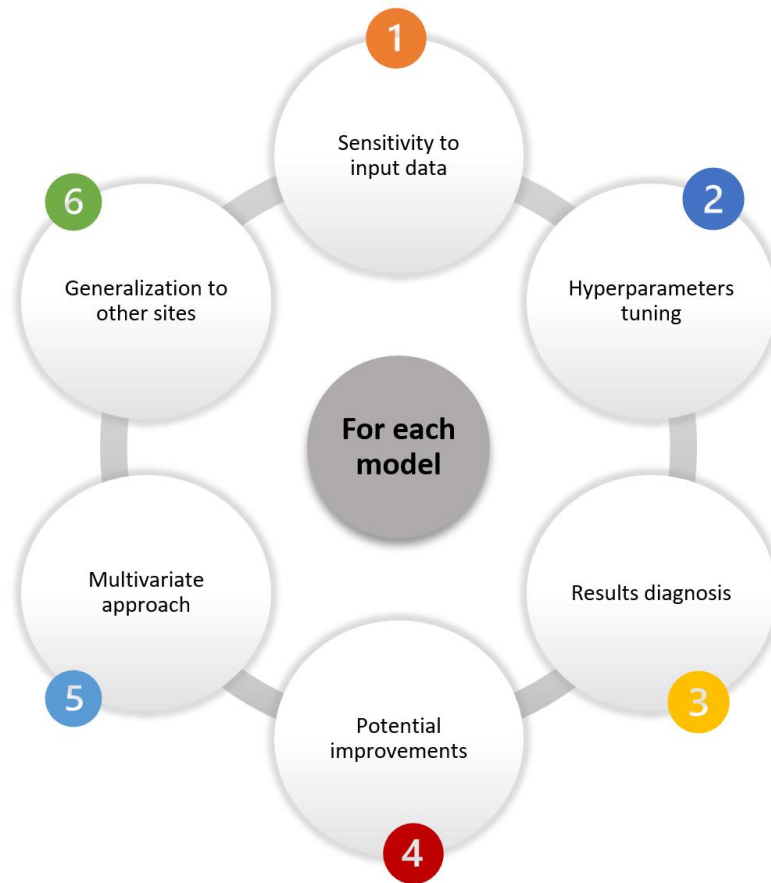
On the other hand, in the field of unsupervised neural networks for anomaly detection in time series, two main categories have emerged: recurrent networks, with their many variants, and autoencoders. We have chosen to focus on **Autoencoders**, which have shown promising results in the literature. While we recognize the potential interest of recurrent networks, their complex implementation and the significant risk of divergence led to their exclusion from our initial research due to time constraints. However, we consider that recurrent networks could be explored in future work, given their likely relevance to our field of study.

In summary, our choice of models to test for data validation and anomaly detection in wastewater network time-series aims to ensure a comprehensive evaluation tailored to our specific context, while exploring different AI approaches (see [Table 8](#)).

**Table 8: Benchmark of the tested AI models**

	Machine Learning	Deep Learning
Supervised	X	ResNet
Unsupervised	Matrix Profile	Autoencoder

The experiments will be conducted using turbidity data from the **Cottage site** as a typical site. The various models are subjected to a pre-defined 6-step process, as illustrated in [Figure 5-2](#).



**Figure 5-2: Diagram of the different tests established for each model from our benchmark.**

In the initial phase, tests are conducted by evaluating the **sensitivity to input data**. This involves determining whether it is preferable to use the raw measured data from T1 and T2 or to consider the reconstructed turbidity output from the filtering stage. Moreover, time series data require some groundwork before it can be mapped by ML algorithms. The crucial issues are temporal regularity and filling in missing data. These challenges have already been leveraged in [Section 4.3.1](#).

However, one last preprocessing issue remains and concerns the scaling of time series. Many ML algorithms achieve better performance if the time series data has a consistent scale or distribution [193]. The two common techniques that can be used to consistently rescale time series data are normalization and standardization. These two terms are used quite loosely in different fields, but they are still different.

Normalization is a rescaling of the data from the original range so that all values are within the range of 0 and 1, we can refer to also as the min-max scaling (see [Equation 4](#)).

**Equation 4: Normalization formula**

$$x_{norm}^{(i)} = \frac{x^{(i)} - x_{min}}{x_{max} - x_{min}}$$

where  $x^{(i)}$  is a measured value,  $x_{min}$  is the smallest value in the dataset and  $x_{max}$  is the largest.

Standardizing a dataset involves rescaling the distribution of values so that the mean of observed values is 0 and the standard deviation is 1. Standardization assumes that the data fits a Gaussian distribution (see [Equation 5](#)).

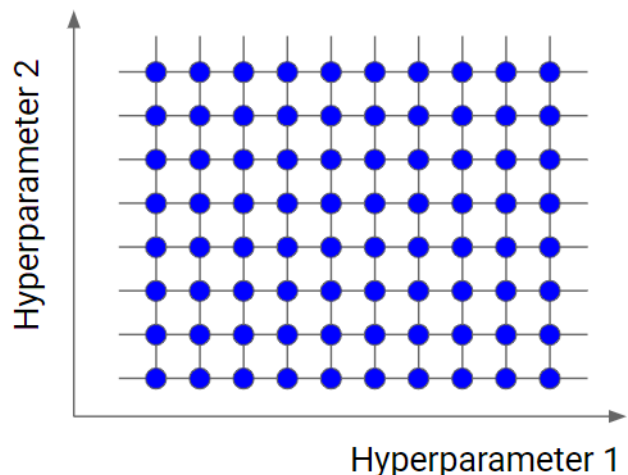
**Equation 5: Standardization formula**

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

where  $x^{(i)}$  is a measured value,  $\mu_x$  is the mean of the dataset and  $\sigma_x$  the corresponding standard deviation.

Therefore, tests will be conducted to determine the optimal strategy in this regard. Additionally, sensitivity tests will be performed on the input database, exploring whether to use the entire dataset, employ a pre-selection, or seek additional data to enhance the learning process.

**Hyperparameter tuning** is required to identify the set of hyperparameters that results in the best performance of each model. In the case of our benchmark models and objective of data validation, input data consists of sequences, making the sequence size and stride the initial parameters to be calibrated. Following this, each model has specific hyperparameters that govern its operation, such as the number of layers and neurons per layer for neural networks. Therefore, prior to testing each model, we will define



**Figure 5-3: Grid Search for hyperparameters tuning**

which hyperparameters will be fixed and which ones will undergo tuning. Different optimization algorithms can be operated, here we use a grid search [194]. It is a brute-force exhaustive search paradigm where we specify a list of values for different hyperparameters (see [Figure 5-3](#)). For each combination, the model is tested and evaluated in order to identify the optimal combination. Grid search for hyperparameter tuning offers several significant advantages. Firstly, the implementation of this approach is relatively simple, with the results being



## Chapter 5. Benchmarking models for data validation and anomaly detection

reproducible (unlike a random search). Moreover, the ability to parallelize the process speeds up evaluations.

Once the best model is identified, the aim of the **results diagnosis** phase is to analyze the results, compare them with the expert's findings and identify the strengths and limitations of each model. By using different visualization approaches, we aim to gain an in-depth understanding of how each model processes data and to distinguish the aspects in which it excels from those that require improvement. Further **improvements** will then be explored in order to leverage the identified limits. These will be specific to each model and will depend on the results of the diagnosis phase. This process will then allow us to assess the model's final performance on the complete database, so that models can be compared with each other.

Furthermore, a **multivariate approach** will be adopted, taking advantage of the different on-site sensors. Our aim is to provide the model, for each timestamp, with additional available data. **Table 9** provides a nomenclature of the different multivariate configurations that will be evaluated.

**Table 9: Nomenclature for the evaluated multivariable approaches and their input data**

	Input data			
Index	Raw T1	Raw T2	Reconstructed T	Conductivity C
2T	X	X		
3T	X	X	X	
2TC	X	X		X
3TC	X	X	X	X

Finally, the **generalization to other sites** will consider the effectiveness of each model within its respective architecture when applied to new data sourced from various other sites. This evaluation aims to determine the adaptability and generalizability of the models, facilitating the understanding of their performance across diverse contexts and contributing valuable insights for broader applicability.

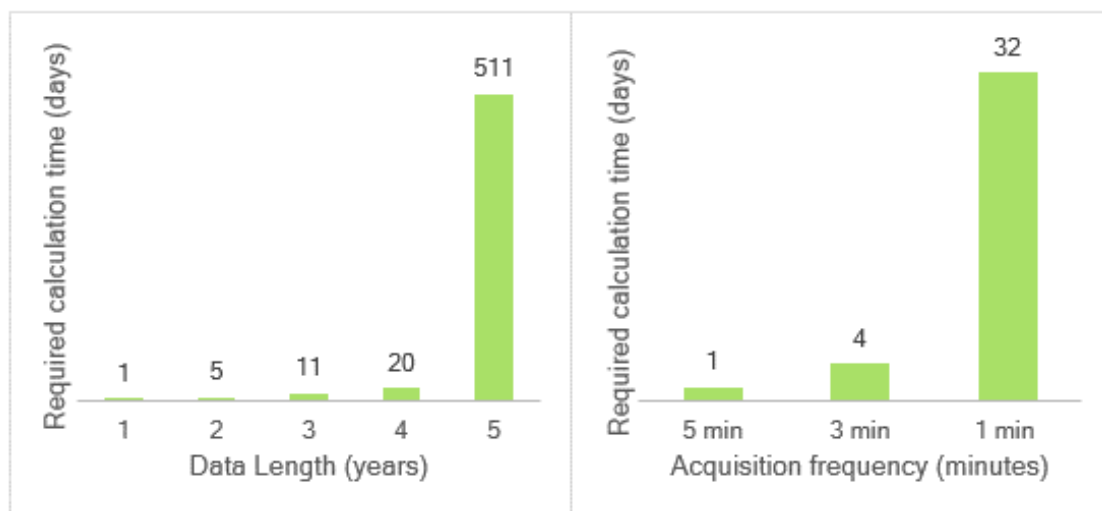
### 5.3 Matrix Profile

#### 5.3.1 Introduction and Background

**Matrix profile** is an algorithm for time series analysis, introduced in 2016 by Eamonn Keogh (University of California Riverside) and Abdullah Mueen (University of New Mexico). Its principle is to perform **similarity join** on time series. The basic problem statement for similarity join is: *Given a collection of data objects, retrieve the nearest neighbor for every object* [195].

## Chapter 5. Benchmarking models for data validation and anomaly detection

This approach has been largely deployed in the text domain [196]. However, despite the analogies between text and time series processing algorithms [197], there has been little progress on time series similarity join. In fact, the application of this approach consists in comparing snippets of the time series against itself by computing the distance between each pair of snippets. The principle is easy to implement using a brute force algorithm with loops, however it may take months or years to receive an answer for a temperately sized time series. Considering a classic turbidity sensor in the wastewater network with an acquisition frequency of 5 minutes, after a year, the dataset length is 105120. An operator may wish to perform a similarity self-join on this data with a day-long subsequences (288). The nested loop algorithm requires 10 989 853 056 Euclidean distance computations. With an assumption that one calculation takes  $10^{-5}$  seconds (using numpy<sup>8</sup>), the task will take approximatively 30 hours. Hence, the main advantage of matrix profile is to provide “an *ultra-fast similarity search algorithm*” and drastically reduce the computation time using an off-the-shelf desktop computer.



**Figure 5-4: Required calculation time depending on (left) Data Length (with an acquisition frequency of 5 min) and (right) Acquisition frequency (with data length of one year)**

Moreover, the Euclidean distance value is difficult to interpret even for a domain expert. Thus, the similarity between time series is evaluated within a user-supplied threshold. However, it is difficult to set an appropriate threshold without domain knowledge. Moreover, in some cases, the threshold must be precise to the 3<sup>rd</sup> decimal place to achieve satisfactory results, such is the case for [199] where they had to set the threshold to 0.818. The strength of MP is that it does not require a threshold. Indeed, once the join is computed, the user can define his own

---

<sup>8</sup> We did not conduct these tests ourselves, but we relied on the estimates of [198]

rules and filters in post processing; for example, having the ten most obvious anomalies. In addition to the features stated above, MP is a domain agnostic model and can leverage hardware by being parallelizable, both on CPUs and GPUs [200].

### 5.3.2 Definitions and Notation

**Definition 1:** A *time series*  $T$  is a sequence of real-valued numbers  $t_i$ :  $T = t_1, t_2, \dots, t_n$  where  $n$  is the *length* of  $T$ .

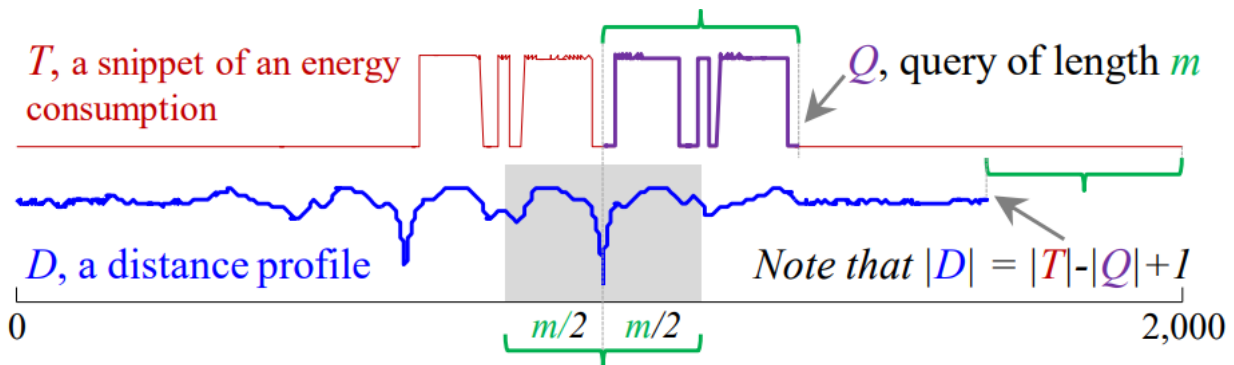
In order to identify anomalies, we are interested in the similarity between local subsequences rather than the global properties of a time series.

**Definition 2:** A *subsequence*  $T_{i,m}$  of  $T$  is a continuous subset of values from  $T$  of length  $m$  starting from the  $i^{\text{th}}$  position:  $T_{i,m} = t_i, t_{i+1}, \dots, t_{i+m-1}$  where  $1 \leq i \leq n - m + 1$

**Definition 3:** A *distance profile*  $D_i$  of a subsequence  $T_{i,m}$  and a time series  $T$  is an ordered array that stores the distance between the query ( $T_{i,m}$ ) and all the other subsequences from the same time series  $T$ :  $D_i = \text{dist}(T_{i,m}, T_{j,m}) \forall j \in [1, 2, \dots, n-m+1]$ .

Distance is measured using the Euclidean distance between the normalized subsequences. By definition, the  $i^{\text{th}}$  location of the distance profile  $D_i$  is zero, and close to zero just before and after this location. Such matches are called *trivial matches* [197]. We avoid such matches by identifying an *exclusion zone* of  $m/2$  before and after the location of  $T_{i,m}$ .

The definitions presented above are illustrated in **Figure 5-5**.



**Figure 5-5:** A subsequence  $Q$  extracted from a time series  $T$  is used as a query.  $D$  is the distance profile of  $Q$ . The grey area is the exclusion zone – © [179]

In order to identify the anomalies, we are interested in finding the nearest neighbors of all subsequences in  $T$  by computing the matrix profile.

**Definition 4:** A *matrix profile*  $P$  of a time series  $T$  is a vector of the Euclidian distances between each subsequence and its nearest neighbor. Formally,  $P = [\min(D_1), \min(D_2), \dots, \min(D_{n-m+1})]$  where  $D_i$  ( $1 \leq i \leq n-m+1$ ) is the distance profile of time series  $T$ .

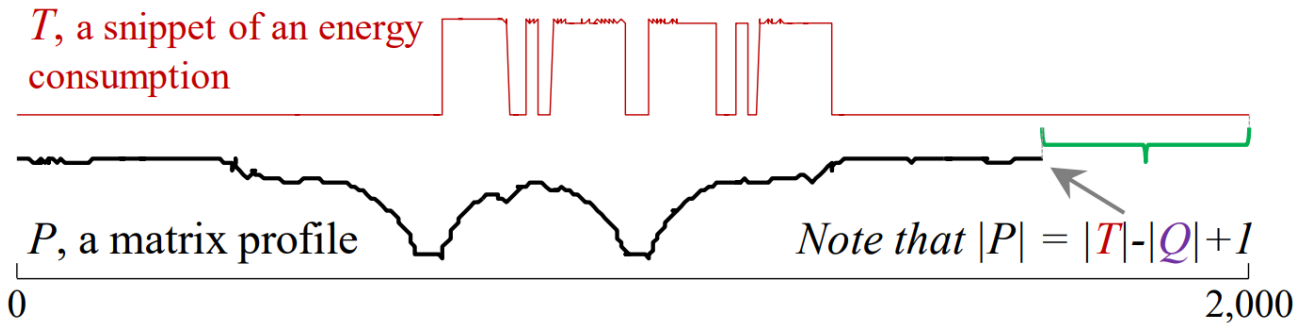


Figure 5-6: A time series  $T$ , and its self-join matrix profile  $P$  - © [179]

This vector is called matrix profile because one naïve way to compute it would be to compute the full distance matrix of all pairs of subsequences in a time series  $T$  and then evaluate the minimum value of each column. However, this brute force approach is not feasible considering both the computational complexity as well as the storage complexity. The process used to simplify the calculation is explained in Appendix G.

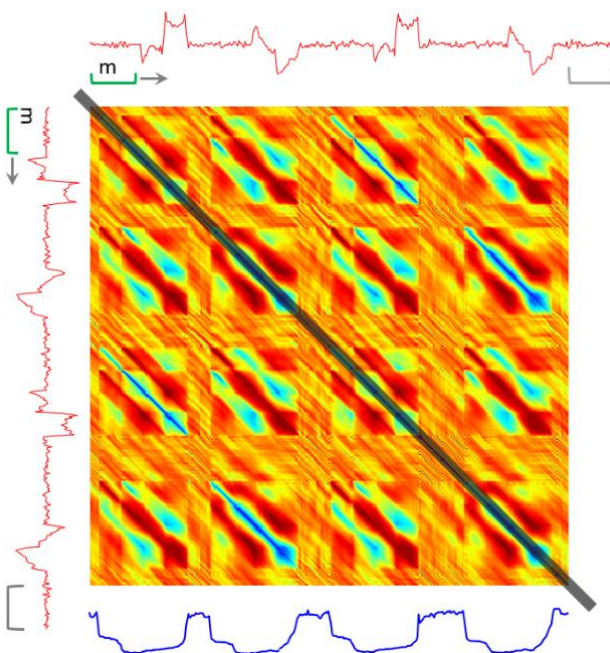
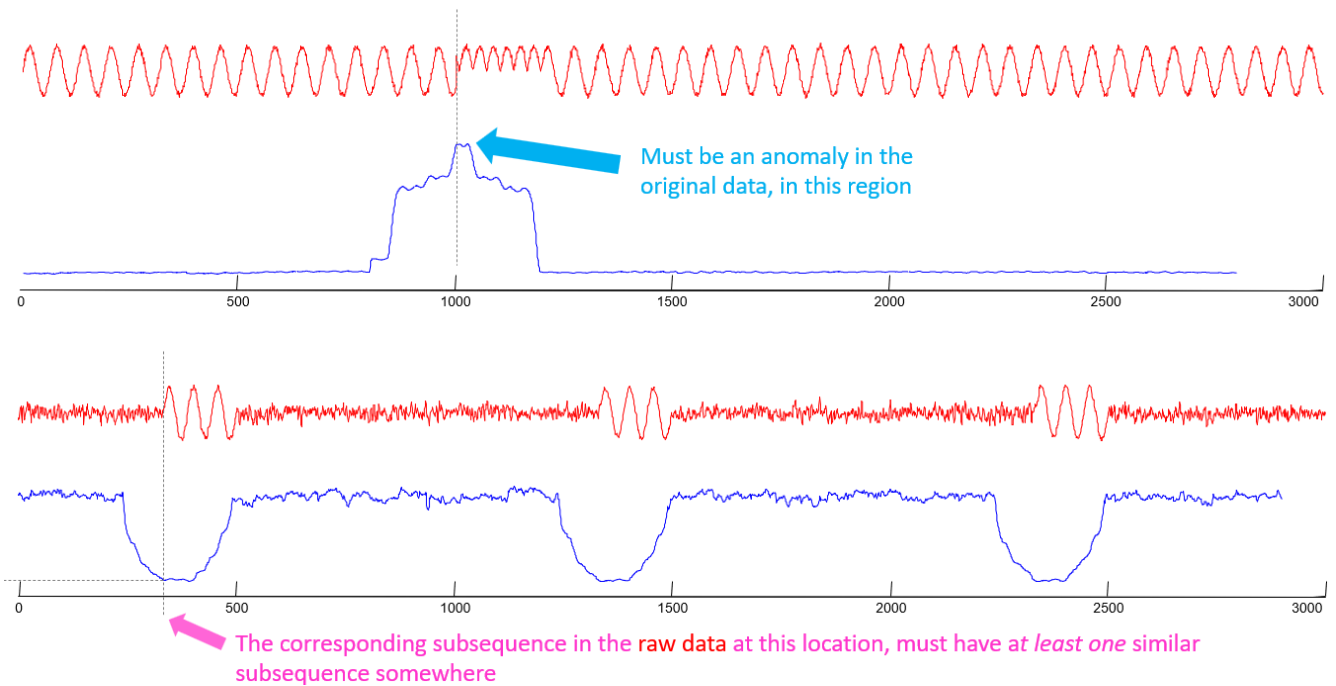


Figure 5-7: Brute Force Matrix Profile: *in red* time series with a subsequence length of  $m$ , *in blue* the matrix profile as defined below, *the heat map* is the distance matrix (Small distances are blue and large distances are red, dark stripe is the exclusion zone) – © UCR

MP model has interesting properties that can be used to identify anomalies. In fact, the highest point on the profile corresponds to the time series discord [201]; meaning that even the nearest neighbor is far away compared to the other subsequences similarity join. On the other hand, the lowest points correspond to the locations of the best time series motif pair [197], but this is not of interest in this research work.



**Figure 5-8: Examples of matrix profile interpretation: *in red* raw synthetic time series and *in blue* matrix profile. *Top*: anomaly detection & *Bottom*: Motif detection – © UCR**

However, the matrix profile does not reveal where the nearest neighbor is located. This information is stored in the matrix profile index.

**Definition 5:** A matrix profile index  $I$  of time series  $T$  is a vector of integers:  $I = [I_1, I_2, \dots, I_{n-m+1}]$  where  $I_i = j$  if  $d_{i,j} = \min(D_i)$ .

In addition to the special case of a single time series, matrix profile generalizes and extends to multidimensional time series.

**Definition 6:** A multidimensional time series  $T$  is a set of co-evolving time series  $T^{(i)}$  of length  $n$ :  $T = [T^{(1)}, T^{(2)}, \dots, T^{(d)}]^T$  where  $d$  is the dimensionality of  $T$  and  $n$  is the length of  $T$ .

**Definition 7:** A multidimensional subsequence  $T_{i,m}$  of a multidimensional time series  $T$  is a set of univariate subsequences from  $T$  of length  $m$  starting from the  $i^{\text{th}}$  position. Formally,  $T_{i,m} = [T_{i,m}^{(1)}, T_{i,m}^{(2)}, \dots, T_{i,m}^{(d)}]^T$ .

Multidimensional time series may have some irrelevant dimensions. Hence, in order to have meaningful results, we might select only a subset of all dimensions. We are talking about subdimensional subsequences.

## Chapter 5. Benchmarking models for data validation and anomaly detection

**Definition 8:** A *subdimensional subsequence*  $T_{i,m}(X)$  is a multidimensional subsequence for which only a subset of dimension is selected.  $X$  is an indicator vector that shows which dimension is included and  $k$  is the number of dimensions included (i.e.,  $\|X\| = k$ ).

The calculation of the distance between two multidimensional subsequences is done by looking at each of its subdimensional subsequences.

**Definition 9:** The *k-dimensional distance* function  $\text{dist}^{(k)}$  computes the distance between two dimensional subsequences by using only the “best”  $k$  out of  $d$  dimensions. Formally,  $\text{dist}^{(k)}(T_{i,m}, T_{j,m}) = \min \text{dist}(T_{i,m}(X), T_{j,m}(X))$  where  $\|X\| = k$ .

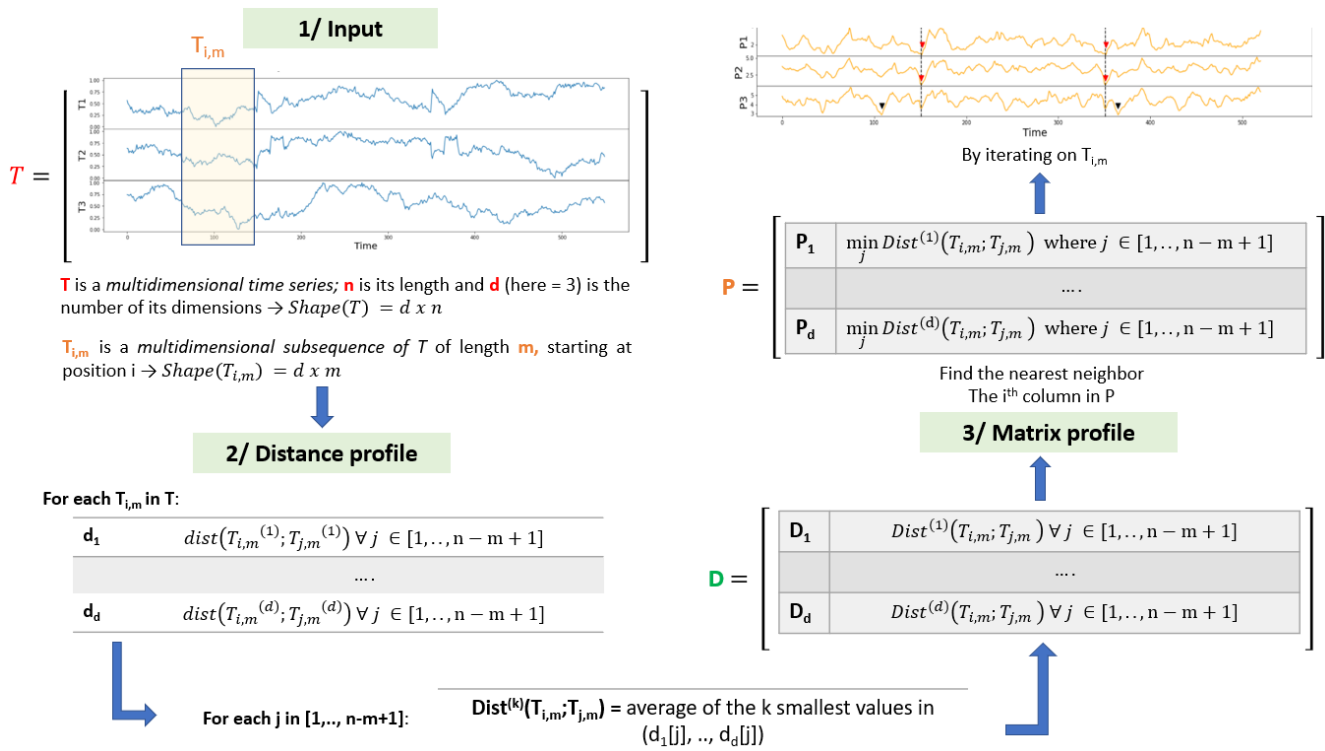
**Definition 10:** The *k-dimensional distance profile*  $D$  of a multidimensional time series  $T$  and a subsequence  $T_{i,m}$  is a vector that stores  $\text{dist}^{(k)}(T_{i,m}, T_{j,m}) \forall j \in [1, 2, \dots, n-m+1]$ .

**Definition 11:** The *k-dimensional matrix profile*  $P$  of a multidimensional time series  $T$  is a meta time series that stores the z-normalized Euclidean distance between each subsequence and its nearest neighbor (using  $k$ -dimensional distance). Formally, the  $i^{\text{th}}$  position in  $P$  stores  $\min(\text{dist}^{(k)}(T_{i,m}, T_{j,m})) \forall j \in [1, 2, \dots, n-m+1]$ , where  $i \neq j$ .

The definitions above are difficult to understand for multidimensional time series. Thus, for better understanding, [Figure 5-9](#) illustrates the full process.

However, the  $k$ -dimensional matrix profile only reveals the location of the anomaly / motif but does not reveal which  $k$  dimensions are involved. To store this information, another meta time series is built.

**Definition 12:** A *k-dimensional matrix profile subspace*  $S$  stores the selected  $k$  dimensions for each subsequence when computing the distance with others.



**Figure 5-9: Matrix Profile for multidimensional time series:  $T$  a time series,  $T_{i,m}$  a multidimensional subsequence,  $D$  the  $k$ -dimensional distance profile and  $P$  the  $k$ -dimensional matrix profile**

### 5.3.3 Principle and algorithms

There are handful of algorithms and different implementations for Matrix Profile (see **Appendix G**). The main time complexity of the MP approach comes from the calculation of the distance profile. In 2011, [202] introduced “The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance”, namely Mueen's Algorithm for similarity Search (MASS). The principle is to create the distance profile of a query to a long time series, exploiting the overlap between subsequences using the classic Fast Fourier Transform (FFT) algorithm. This algorithm represents the basis for the calculation of the matrix profile. Once the distance profile is calculated for each subsequence of a time series  $T$ , the matrix profile becomes a simple loop that extracts the minimum of each row. The algorithm that we used in this research is PYSCAMP<sup>9</sup>, which is a python implementation of MP [203], with high computational optimizations that are beyond the scope of this work [204]. For the multidimensional approach, we used the python implementation, publicly available in [205].

<sup>9</sup> Source code accessible online <https://scamp-docs.readthedocs.io/en/latest/pyscamp/intro.html>

### 5.3.4 Anomaly detection using Matrix Profile

Once the matrix profile is calculated, additional algorithms must be used to extract information from it. In our study, we are only interested in the identification of anomalies and discords. For this, we use the **discover.discords** function developed by matrix profile foundation [206]. This algorithm finds the top K number of discords given a matrix profile.

#### 5.3.4.1 Sensitivity to input data

The main tests carried out in this context concern sensitivity to the input data, by comparing raw and reconstructed turbidity. For the MP model, which uses normalized Euclidean distance, the data are already scaled, eliminating the need for any additional scaling step. However, we take advantage of the model's ability to operate even in the presence of missing data to validate our approach of imputing missing data with zeros. Subsampling and data smoothing tests will also be carried out to validate our acquisition and sampling strategy.

#### 5.3.4.2 Hyperparameters tuning

According to [179], MP is a parameter free algorithm. In fact, [200] assumes that the window size  $w$  is not a real hyperparameter, but a user choice reflecting a prior knowledge of the domain. It corresponds to a typical duration of a motif or discord. Likewise,  $K$  in the mining algorithm introduced in [Section 5.3.4](#), is the number of anomalies that the user wants to identify. However, in our use case, the objective of using MP is to identify **all anomalies** that are present in the time series without having a prior knowledge of the adequate window length, nor the number of anomalies. Hence, we consider two hyperparameters to tune: **the window size  $w$  and the number of anomalies  $K$** . For this, we vary the window size  $w$  between 2 and 72 hours with a time step of 2 hours and the anomaly ratio  $k^{10}$  is set between 5% and 20% with a step of 0.5%, using a grid search.

#### 5.3.4.3 Results diagnosis

Analysis of Matrix Profile results begins with an examination of the model output in the form of a graph (Curve C in [Figure 5-10](#)) and heatmap (Graph B in [Figure 5-10](#)) describing the evolution of the matrix profile over the input dataset (Curve A in [Figure 5-10](#)). The matrix profile plot identifies the peaks associated with the anomalies, to which are added red stars representing the K top anomalies identified (here  $K=3$ ). Each star indicates the starting index of the anomalous sequence. A second representation of the matrix profile takes the form of a heatmap, offering a broader view of defects and a less sharp delineation of their extent. This

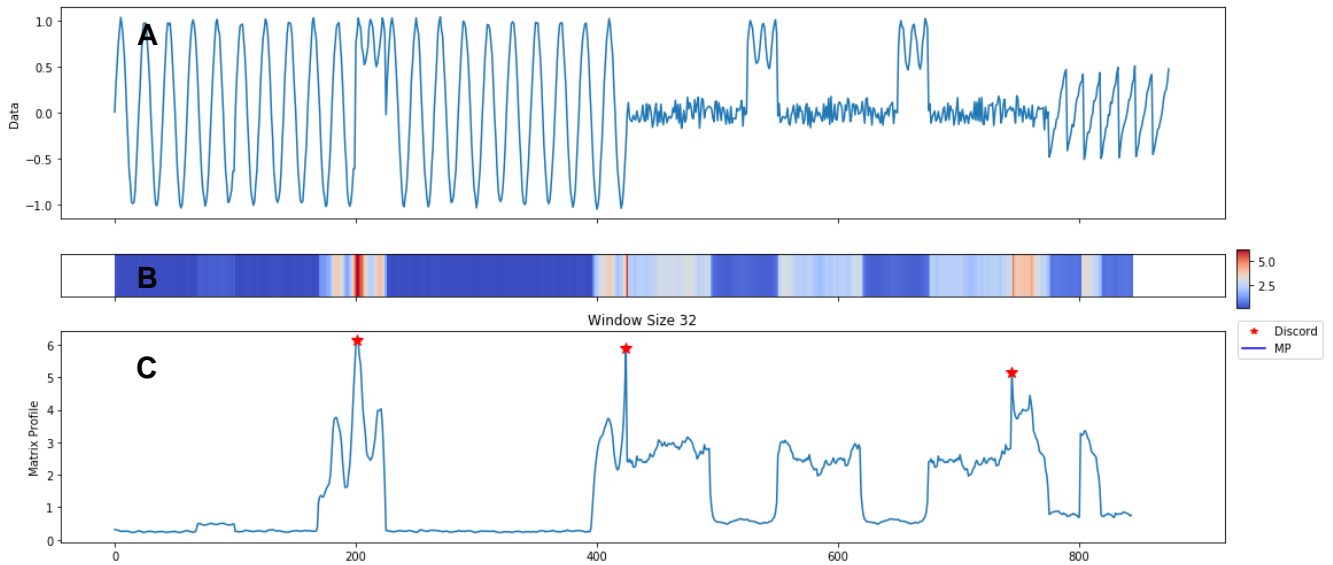
---

<sup>10</sup> *The number of anomalies  $K = \text{the anomaly ratio } k \times \text{the dataset length} / \text{the window size}$*



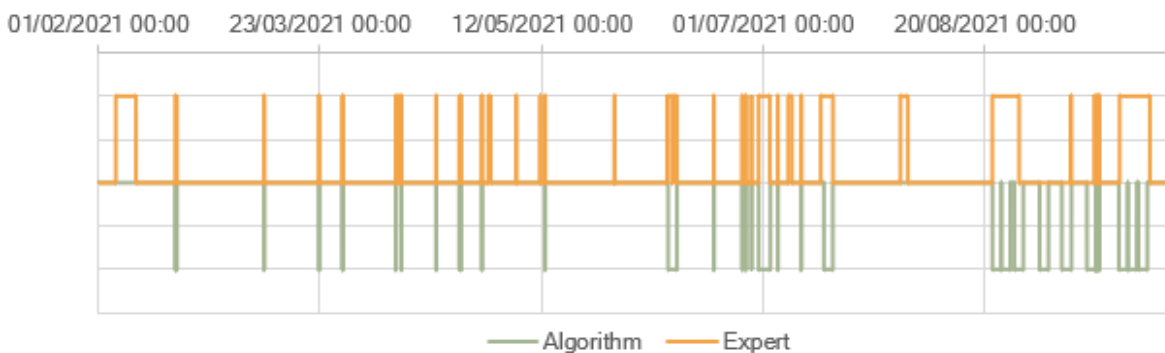
## Chapter 5. Benchmarking models for data validation and anomaly detection

analysis is carried out by examining the color range around the invalid sequence, where red hues signal the presence of anomalies, while shades of blue correspond to normality.



**Figure 5-10: Example of output of the "Matrix Profile" model - © [206]**

Next, the analysis of the results involves a comparison between the model output and that of the manual validation process (filtering + expertise + aggregation). The two approaches are carried out at two levels: at sequence level and at 5-minute interval level. For expert validation, whose output is based on 5-minute intervals, a sequence is considered invalid if more than half of its timestamps are invalid. Furthermore, to adjust Matrix Profile to the 5-minute interval scale, if the latter invalidates a sequence, all its time intervals are considered invalid. The comparison takes the form of a Boolean graph comparing the two sets of results (see [Figure 5-11](#)). A value of 1 or -1 indicates the presence of an anomaly, while 0 indicates normality. The subsequent aim is to analyze the convergences and divergences in order to identify the model's errors and successes.



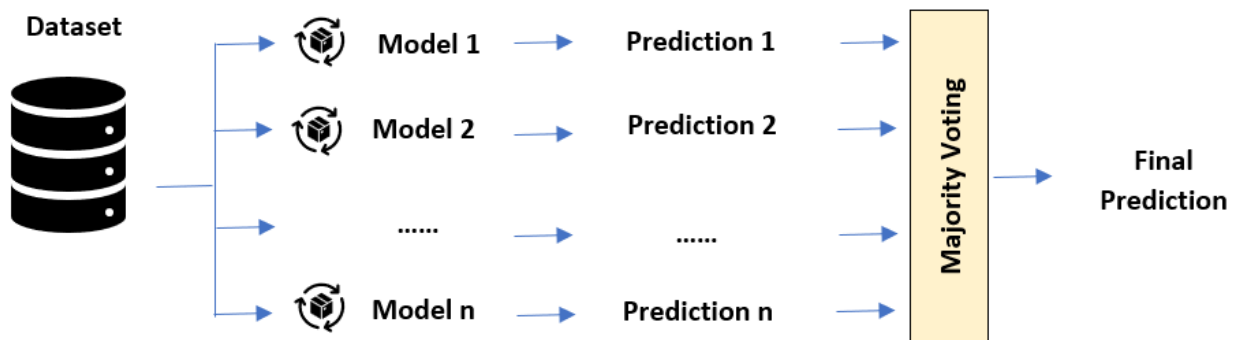
**Figure 5-11: Comparison of the model results and those issued from the manual validation. A value of 1 indicates the presence of an anomaly**

#### 5.3.4.4 Potential improvements

The main problem using MP is the fixed length of anomalies using a window size  $w$ . However, in real-life scenarios, anomalies can be long-lasting (for example a sensor malfunction that requires on-site intervention) or temporary (such as deposits on the sensor which are naturally cleaned by the flow). Hence to tackle this heterogeneity in the discords, **ensemble models** have been tested.

The goal of ensemble methods is to combine multiple models into a meta-model that has a better generalization performance than each individual model [193]. The outputs from the submodels are combined using different techniques to produce the output of the entire system [207]. One of the most popular techniques for ensemble learning in the context of classification / clustering is majority voting.

Majority voting (or plurality voting for multiclass classification) selects the class label that has been predicted by more than 50% of the votes as the final output of the meta-model. **Figure 5-12** illustrates the concept of using majority voting.



**Figure 5-12: Ensemble model using majority vote.**

Hence, to consider the different anomaly sizes and to potentially improve the performance of MP, we tested an ensemble model with MP using different window sizes  $w$ . We have limited the number of sub-models to three. The choice of the different subsequence lengths was made by considering both the optimization of the hyperparameters as explained in **Section 5.3.4.2** and domain knowledge. The majority voting technique was considered. It allows us to identify “confirmed” anomalies. Moreover, we have adapted this approach to a minority vote to identify a maximum of anomalies.

### 5.3.4.5 Generalization to other sites

Matrix Profile stands out as an unsupervised model with no traditional learning process. This technique for exploratory analysis of temporal data involves a systematic comparison of each sequence window with all other possible windows to measure similarity. Unlike a learning model, MP does not retain knowledge between different time series, meaning that the process has to be reinitialized when applied to a new site. Despite this limitation, in tests at other sites such as “Goutte” and “Roosevelt”, we are assessing Matrix Profile's adaptability to different anomaly rates. These tests also aim to determine the sensitivity of the pre-calibrated hyperparameters to these specific data characteristics.

### 5.3.5 Conclusion

Matrix profile is an algorithm based on the principle of similarity join, which makes it well suited to detect anomalies in time series. After calculating the matrix profile, specific functions can be used to extract the main discordances. The particularity of the MP model lies in its ability to operate without the need for a prior learning process. The predominant advantage of this algorithm is the ease of use and the interpretability of the results. The use of Matrix Profile for anomaly detection often involves visual analysis to identify peaks associated with anomalies.

The tests applied to the MP model encompass several aspects crucial to assessing its effectiveness in detecting anomalies. Firstly, sensitivity to input data is assessed by comparing raw and reconstructed data, while exploring the model's ability to operate in the presence of missing data. Next, hyperparameter settings are addressed, notably window size ( $w$ ) and number of anomalies ( $K$ ). These tests are carried out via an exhaustive grid search, varying the window size between 2 and 72 hours and adjusting the anomaly ratio between 5% and 20%. Results are analyzed using matrix profile graphs, heatmaps and a comparison with manual validation. In addition, the model is tested against anomaly scenarios of varying durations, using ensemble models with different window sizes. Finally, the generalization of the model to other sites is explored to assess its adaptability to different anomaly rates over varied time series. These collective tests provide a comprehensive assessment of the performance and capabilities of the Matrix Profile model in anomaly detection for wastewater network data.

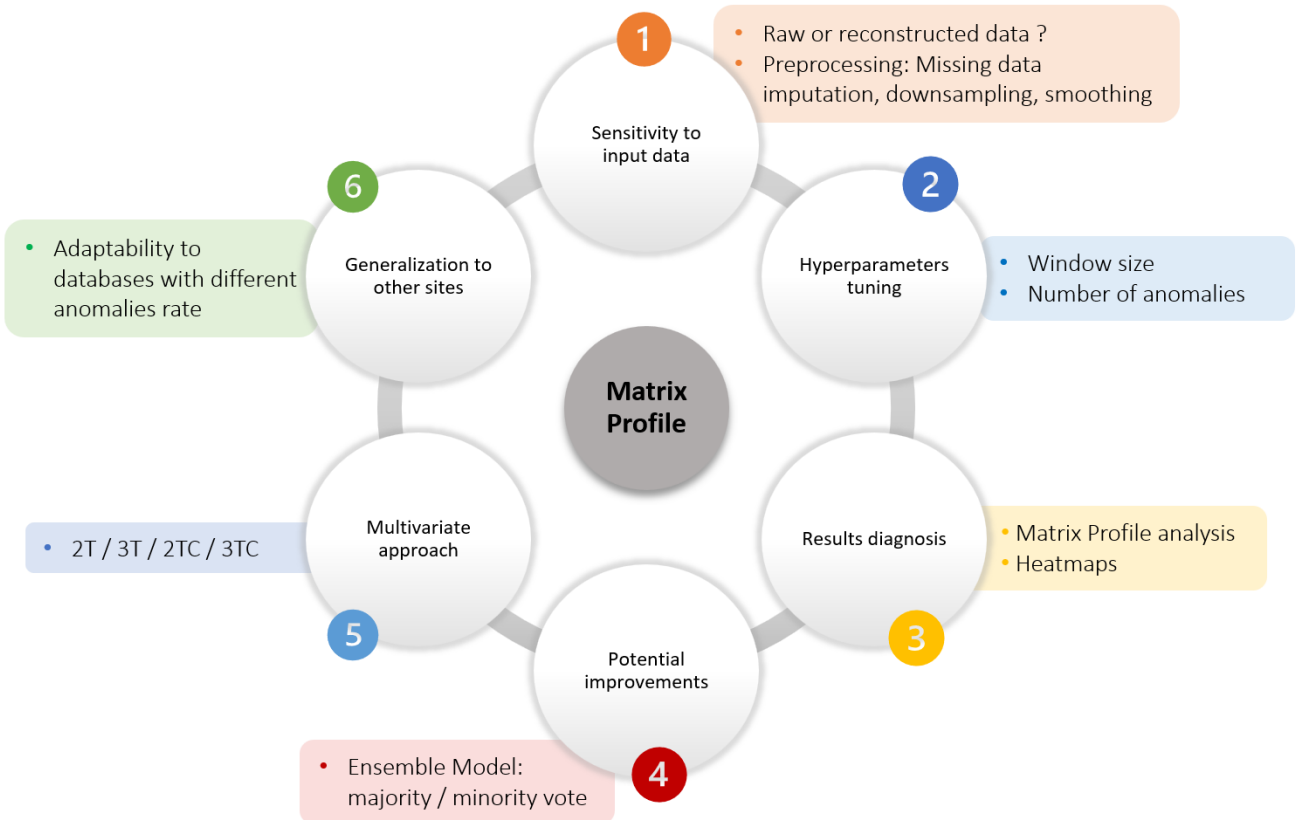


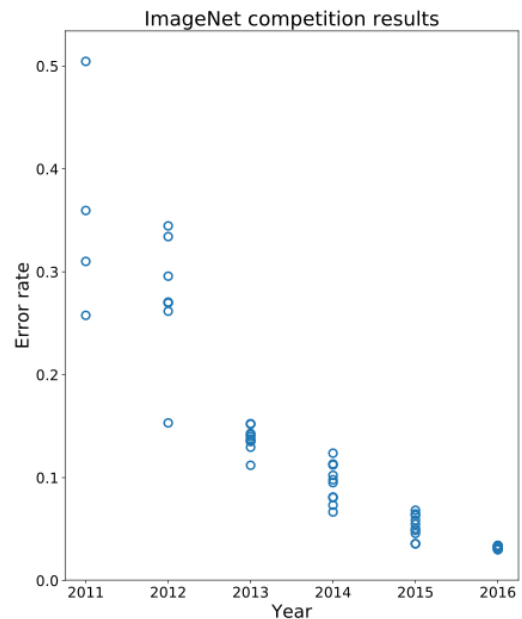
Figure 5-13: Overview of tests related to Matrix Profile model

## 5.4 ResNet

### 5.4.1 Introduction and background

The ResNet model, developed by [208], played a significant role in the ImageNet project. ImageNet is an organization that created a massive database of annotated images for visual object recognition software research [209]. Between 2010 and 2017, the ImageNet project hosted an annual competition known as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [210]. The competition focused on accurately detecting and classifying objects and scenes in images using AI algorithms. The advancements in image processing during the 2010s were remarkable (see [Figure 5-14](#)). In 2011, the lowest classification error rates in the ILSVRC competition stood at approximately 25%. However, in 2012, the advent of deep learning revolutionized the field, reducing the error rate to 16%. Subsequently, over the next years, the error rate dropped significantly to just a few percent. In 2015, ResNet emerged as the winner of the competition [211].

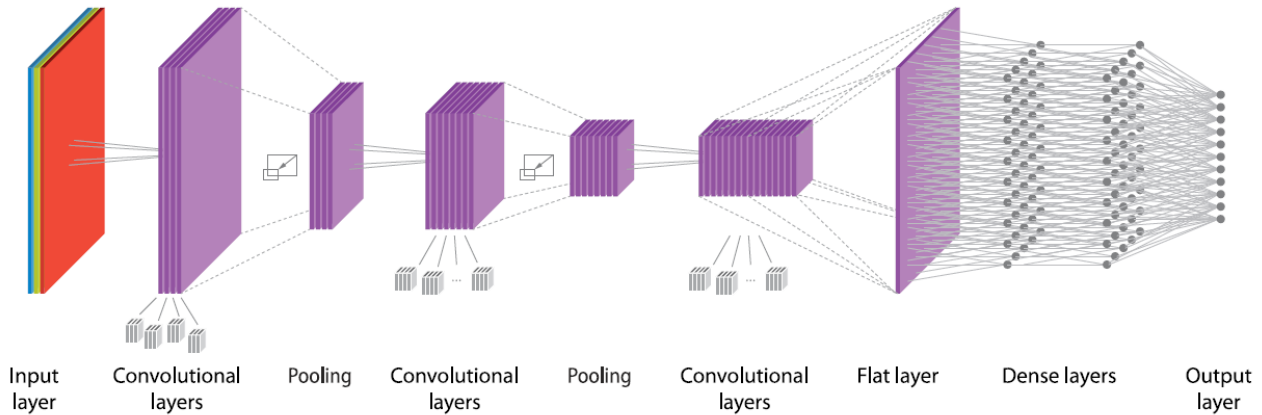
Following the victory of AlexNet, a CNN-based architecture in the ImageNet 2012 competition, subsequent winning architectures have consistently employed deeper neural networks with an increased number of layers to decrease the error rate. While this approach proves effective with a limited number of layers, it introduces a common challenge in deep architectures known as the Vanishing / Exploding gradient problem when the number of layers is increased. This problem manifests itself when the gradients diminish as they flow backward through the network. Indeed, repeated multiplication may make the gradient infinitely small. As a result, the deeper the network goes, the more its performance becomes saturated or even starts rapidly degrading. ResNet, which was proposed in 2015 by researchers at Microsoft Research, remedies this problem by introducing a new architecture called Residual Network.



**Figure 5-14: Error rate history on ImageNet**

### 5.4.2 Definitions and notation

The ResNet model draws its essence from the fundamental structure of CNNs. [Figure 5-15](#) illustrates a typical CNN structure, highlighting the convolution and pooling layers that make up its core.



**Figure 5-15: Convolutional Neural Network Architecture**

#### Definition 1: Convolutional layers

The convolutional layer serves as the fundamental building block within the CNN architecture. This layer executes a dot product operation between two matrices: one matrix represents the input of shape (input height  $\times$  input width  $\times$  input channels), while the other matrix is the **kernel**. Notably, the kernel is spatially smaller than the input. In the context of an RGB image with three channels, the kernel's height and width are compact, but its depth spans across all three channels. Throughout the forward pass, the kernel systematically traverses the height and width dimensions of the input, performing convolution operations.

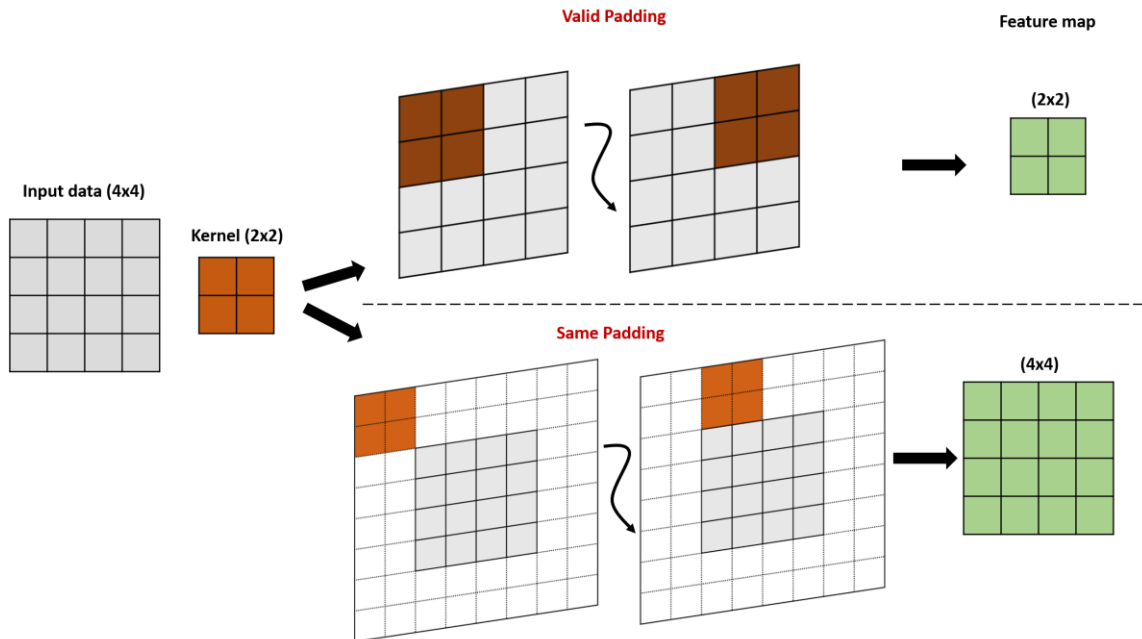
Following the passage through a convolutional layer, the input undergoes a transformation, resulting in the creation of a **feature map**, also referred to as an activation map. The derivation of this feature map relies on three crucial hyperparameters, which necessitate configuration prior to the neural network training. The first element is obviously the kernel size, while the other two hyperparameters are as follows (see [Figure 5-16](#)):

- **Stride**, which is the step size of the kernel as it traverses the input image. Typically set to 1 by default, while adjusting the stride to a higher value can be employed for downsampling an image.
- **Zero-padding** which dictates how the border of a sample is handled. Two main distinct types of padding exist:
  - Valid padding: Also known as no padding, this approach results in the exclusion of the last convolution if dimensions fail to align

## Chapter 5. Benchmarking models for data validation and anomaly detection

- Same padding: This padding technique ensures that the output layer maintains the same size as the input layer by incorporating zeros.

Post each convolution operation, a Rectified Linear Unit (ReLU) transformation is applied to the feature map within the CNN. This introduces nonlinearity to the model, enhancing its capacity to capture intricate patterns and relationships within the data.



**Figure 5-16: Different padding approaches (here the stride = 2)**

### Definition 2: Pooling layers

Pooling layers, also recognized as downsampling, execute dimensionality reduction, thereby diminishing the number of parameters in the input. Analogous to the convolutional layer, the pooling operation employs a filter that traverses the entire input yet diverges in that this filter lacks any weight. Instead, the kernel applies an aggregation function to the values within the receptive field, generating the output array. Two predominant types of pooling exist (see [Figure 5-17](#)):

- **Max pooling:** During the filter's traversal across the input, it selects the pixel with the maximum value to transmit to the output array.
- **Average pooling:** As the filter moves across the input, it computes the average value within the receptive field, which is then forwarded to the output array.

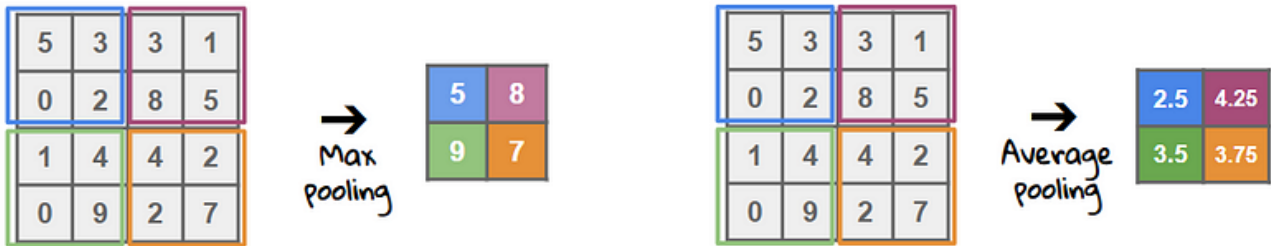


Figure 5-17: Pooling approaches

While the pooling layer inevitably results in information loss, it confers several advantages to CNNs. It contributes to complexity reduction, enhances computational efficiency, and mitigates the risk of overfitting by fostering a more generalized representation of the input data.

**Definition 3:** Residual connections

The key idea behind ResNet is the introduction of residual connections, also known as skip connections or shortcut connections. In traditional deep neural networks, each layer learns a mapping from its input to its output. However, as the network becomes deeper, it becomes challenging for the network to learn and propagate gradients effectively. This can lead to the vanishing gradient problem. The **skip connections** enable ResNet to learn residual mappings instead of directly learning the desired underlying mappings. Residual connections allow information to bypass certain layers in the network, making it easier for the gradients to flow during backpropagation. This helps alleviate the vanishing gradient problem and enables the training of very deep networks with hundreds or even thousands of layers.

**Definition 4:** Residual Blocks

The ResNet architecture consists of a series of **residual blocks**. Each residual block typically contains multiple convolutional layers with batch normalization and non-linear activation functions, such as ReLU (Rectified Linear Unit). The residual connection skips one or more convolutional layers within the block, and the input is added to the output of the block using element-wise addition (see [Figure 5-18](#)). This residual connection ensures that the information from earlier layers can flow more easily to the deeper layers, facilitating gradient flow and improving the network's ability to learn.



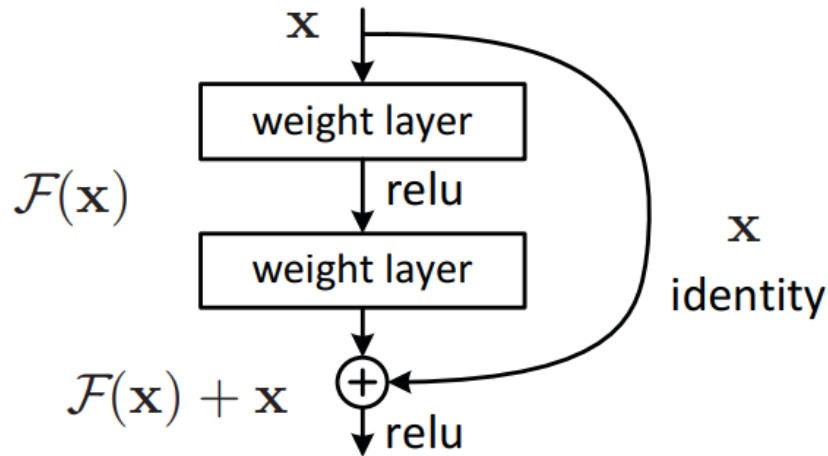


Figure 5-18: Residual learning: a building block - © [208]

The approach behind this network is instead of layers learning the underlying mapping  $H(x)$ , the network will fit the residual mapping  $F(x)$  according to Equation 6:

Equation 6: Residual mapping approach

$$F(x) = output - input = H(x) - x \rightarrow H(x) = F(x) + x$$

Definition 5: Batch normalization

Batch normalization is a fundamental technique in DL aimed at stabilizing and accelerating the training of neural networks. This method consists in normalizing the activations of a layer by adjusting their mean and standard deviation on a mini lot of data during training. Specifically, for each activation channel, batch normalization centers and resizes the values using the mean and standard deviation of the mini batch (see Figure 5-19). The introduction of batch normalization reduces covariance mismatch problems, stabilizing the optimization process. In addition to promoting faster model convergence, batch normalization acts as a regulator, reducing the risk of overfitting and enabling the use of higher learning rates.

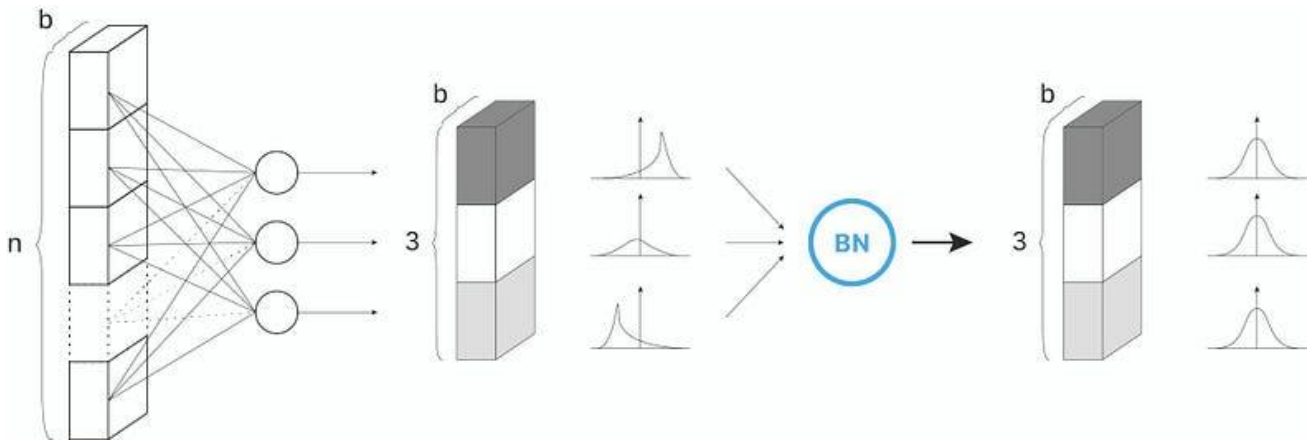


Figure 5-19: Batch Normalization process - © [212]

### 5.4.3 Model architecture

ResNet architectures come in different variations, such as ResNet-18, ResNet-50 and ResNet-152, which indicate the number of layers in the network. These architectures typically consist of several blocks, where each block contains multiple convolutional layers along with batch normalization and ReLU activation functions. The skip connections are added within each block, allowing the network to learn residual mappings at different depths. The model architecture utilized in this study follows the design proposed by [133] and illustrated in Figure 5-20.

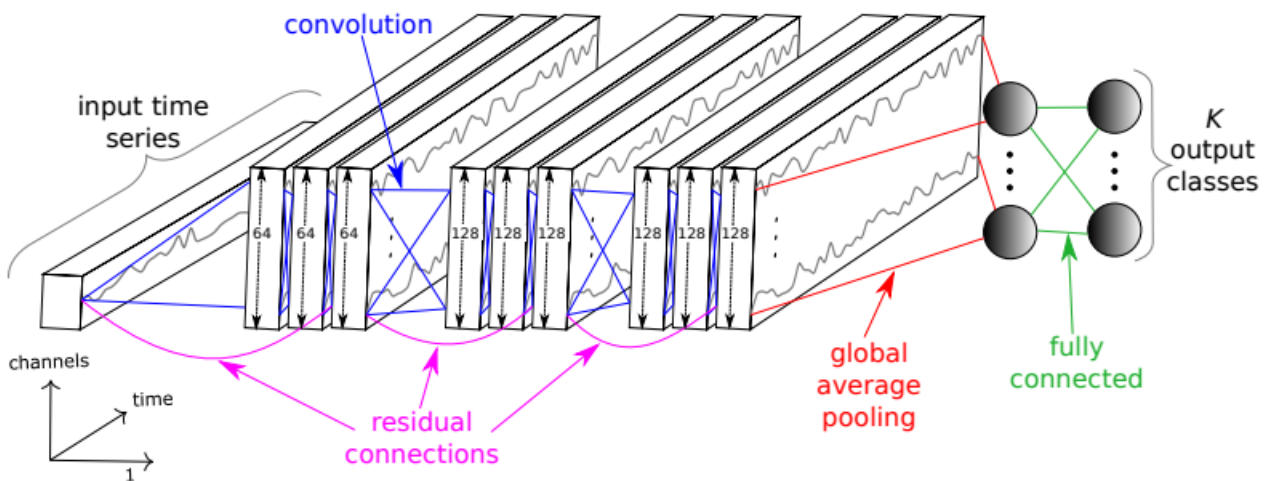
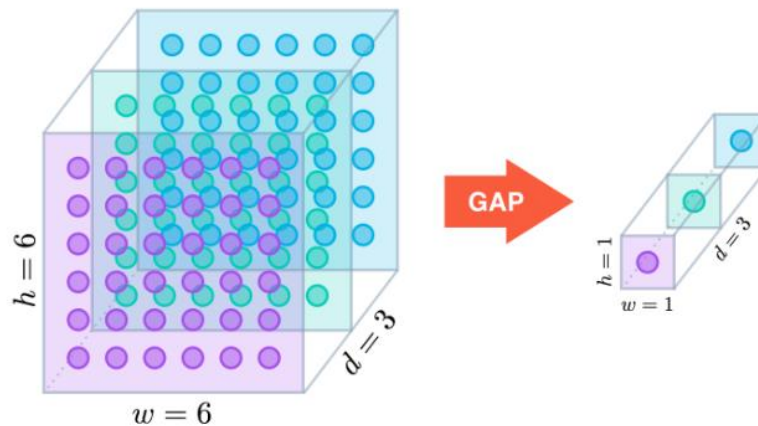


Figure 5-20: Architecture of the used ResNet model - © [134]

The network consists of three residual blocks, which are sequentially connected. Each **residual block** comprises three convolutions. The output of these **convolutions** is added to the input of the **residual block** and then passed to the subsequent block. The number of filters for convolutions of each block is {64, 128, 128}, and the ReLU activation function is applied

after a batch normalization operation. In each residual block, the lengths of the filters are set to 8, 5, and 3, respectively, for the first, second, and third convolutions.

Following these residual blocks, there is a **Global Average Pooling (GAP) layer**, which is a pooling operation that computes the average value of each feature map across spatial dimensions. It reduces the spatial dimensions of the feature maps to a single value per channel. This process helps capture the most important information from each feature map and aggregates it into a compact representation (see [Figure 5-21](#)).



**Figure 5-21: Global Average Pooling principle**

The final layer of the model is a **SoftMax classifier**. SoftMax is a mathematical function that converts the output of the previous layers into a probability distribution over the classes in the dataset (see [Appendix A](#)). It assigns a probability value to each class, indicating the likelihood of the input belonging to that class. The class with the highest probability is selected as the predicted class label.

### 5.4.4 Anomaly detection using ResNet

Detecting anomalies in time series using ResNet models relies on the ability of these networks to learn deep and meaningful representations of temporal data. To implement this process, the architecture of the ResNet model is adapted to take as input a temporal sequence of predefined duration and have as output the label of the sequence: valid or invalid.

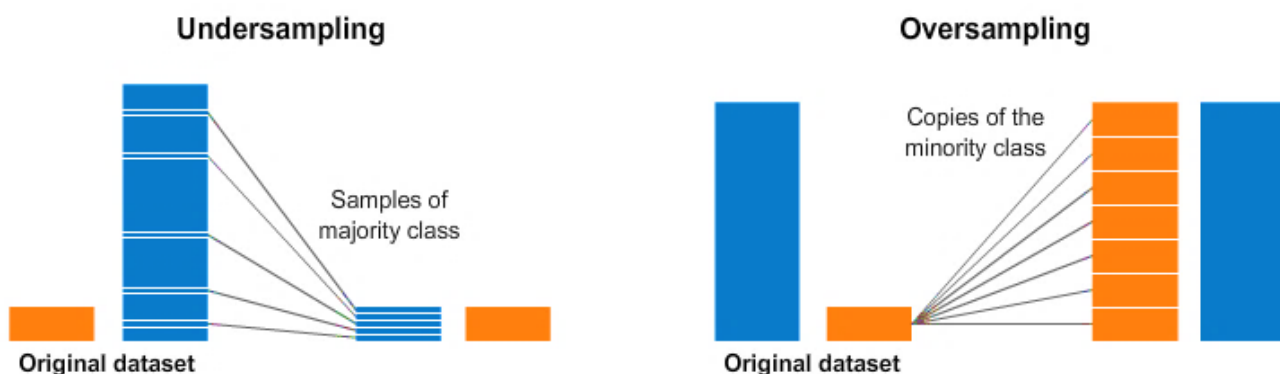
#### 5.4.4.1 Sensitivity to input data

In the preprocessing step for input time series data in the ResNet model, a sub sequencing approach aims to divide the original time series into smaller subsequences of fixed length, allowing the model to capture temporal dependencies within a specific timeframe. Since the sequence itself contains sufficient temporal information, shuffling the subsequences is generally acceptable. Randomly shuffling the order of subsequences helps prevent any

inherent biases that may exist in the original dataset from affecting the model's learning process.

Similar to the Matrix Profile, **sensitivity tests to input** data will be conducted for the ResNet model, comparing raw and reconstructed turbidity. However, being a deep and supervised neural network, it is crucial to examine the representativeness of the sequences. The primary issue that arises is class imbalance. Anomalies, by their definition, constitute a minority, and thus, it is essential to explore how this affects the learning process. Upon examining the Cottage turbidity database throughout the entire year, an anomaly rate of 8% is observed at the daily scale. In other words, the number of valid sequences is 11 times greater than the number of invalid sequences. Therefore, the utilization of data enhancement, with the aim of balancing the two classes, becomes pertinent in addressing this issue.

**Data enhancement** can be achieved by downsampling or up-sampling (see [Figure 5-22](#)). Downsampling involves reducing the number of samples in the majority class (valid sequences), while up-sampling aims to increase the number of samples in the minority class (invalid sequences). Downsampling has the disadvantage of potentially losing useful information. Hence we favor up-sampling, by creating duplicates of the same sequences. For oversampling, we must set the sampling strategy in the enhanced database. When float, the value corresponds to the desired ratio between the number of samples in the minority class over the number of samples in the majority class after resampling. Otherwise, we can use 'minority' to resample the minority class in order to equalize the number of samples between classes.



**Figure 5-22: Classic data enhancement strategies**

A second enhancement approach is to inject white noise into sequences already identified as invalid. The underlying assumption is that adding noise to an invalid sequence will keep it invalid. This method aims to introduce additional samples and variability into the training data.

## Chapter 5. Benchmarking models for data validation and anomaly detection

In practice, we add noise using random samples from a normal Gaussian distribution with a mean equal to 0 and using a variable scale / width.

Finally, a third approach is to adopt the concept of cost-sensitive learning. This approach takes into account the cost associated with each class when training the model (as a dictionary {class 0 weight: class 1 weight}), which is particularly useful in cases of class imbalance. It allows classification errors to be given different weights depending on the class, thus favoring the accuracy of the minority class, in our case, invalid sequences (class 1).

### 5.4.4.2 Hyperparameters tuning

In this study, the focus will not be on optimizing the structure of the model, such as the number of hidden layers or other architectural aspects. The main modifications will be made to the input and output of the model. By keeping the underlying model architecture consistent, it will be possible to isolate and evaluate the effects of specific changes, providing insights into their individual contributions to the overall performance of the anomaly detection system.

Sensitivity tests will be conducted to determine the **optimal input window size**, ensuring that the selected window adequately captures the relevant patterns and characteristics in the data. While the default window size is 24-hours, tests ranging from 2 hours to 72 hours will be conducted using a stride of 2.

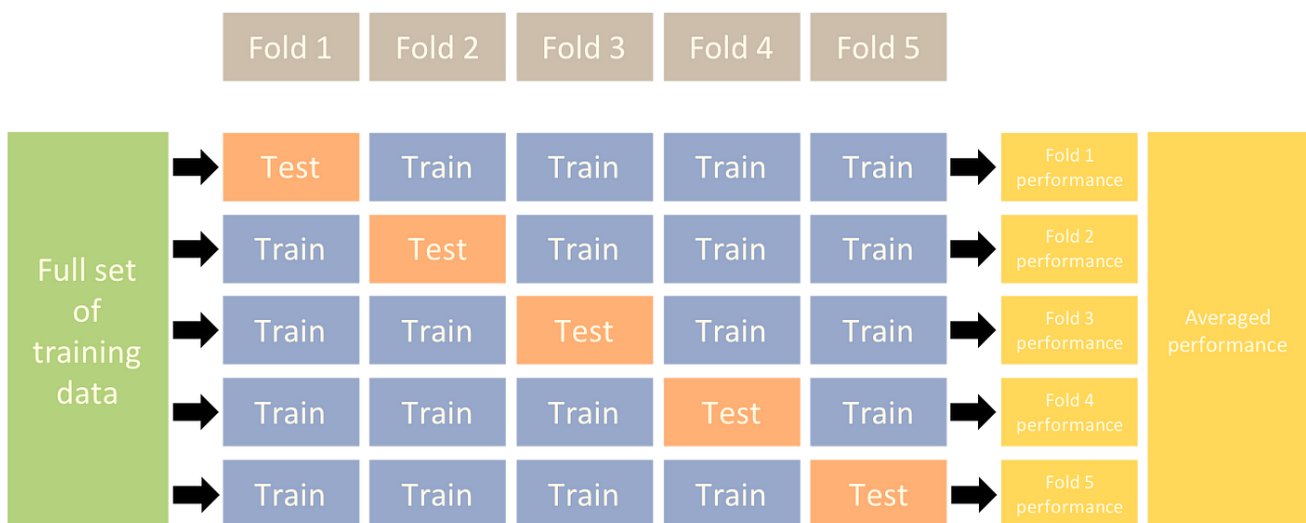
Regarding the **output target**, a transformation is performed to convert the labels from a time step-by-time step basis to a label per sequence basis. If more than 50% of a sequence is considered invalid, the entire sequence is labelled as invalid (indicating the presence of an anomaly). This simplification allows the model to focus on classifying the entire sequence rather than individual time steps. Finally, to represent the target classes, a one-hot encoding scheme is applied. This converts the two classes, valid (0) and invalid (1), into binary vectors, where each class is represented by a unique combination of zeros and ones. Hence one-hot encoding transforms the "valid" label into [1, 0] and the "invalid" label into [0, 1]. The main advantage of this approach lies in its ability to provide a clear, binary representation of classes, while eliminating any notion of order between classes.

According to its definition, the final layer of the ResNet model comprises two neurons representing the valid and invalid classes, indicating the probability of belonging to each class. Consequently, the class with the highest probability is assigned to the analyzed sequence. Nevertheless, examinations of the appropriateness of the **classification threshold** will be undertaken. The question addressed is whether a probability exceeding 50% is the most effective way to determine the model output. To refine the results, different strategies will be

implemented. This involves assigning greater weight to invalid sequences on one hand and adjusting the classification threshold on the other. The aim is to explore variations in the model's output and assess the impact of different threshold levels on the overall performance of the ResNet model.

### 5.4.4.3 Model training

The ResNet model is trained using a **K-fold cross-validation strategy** to assess the model's generalization ability by training and evaluating it on different combinations of training and validation data. The principle of K-fold cross validation is to divide the data set into K folds (or subsets) of equal size. The model is then trained K times, each time using K-1 folds as the training set and the remaining fold as the validation set. This procedure is repeated K times, each fold being used once as a validation set. At each iteration, model performance is evaluated. K-fold cross-validation mitigates variations due to the arbitrary selection of training and validation sets, offering a more reliable assessment of model performance. Ultimately, average performance over K iterations is often used as the final measure of model performance, offering a more stable and representative estimate of its ability to generalize to new data. In our case study we used a 5-fold cross validation, hence ensuring an 80% ratio for training and 20% for validation (see [Figure 5-23](#)).



**Figure 5-23: K-fold cross validation strategy**

During training, several useful **callbacks** are employed to enhance the training process. Callbacks enable dynamic and real-time monitoring. **TensorBoard**, for example, offers a graphical visualization of the learning progress, enabling key metrics to be monitored over epochs. Initially, the number of **epochs** per fold can be set at a high value, such as 1000 in our case. However, by following the learning curves generated by TensorBoard, this number

## Chapter 5. Benchmarking models for data validation and anomaly detection

can be adjusted according to model convergence, thus reconciling computation time and performance.



**Figure 5-24: TensorBoard: (in orange) the accuracy of training data, (in blue) the accuracy of validation data along different epochs during the training process**

On the other hand, the **ModelCheckpoint** callback allows saving the model with the best performance observed during training. This is essential to prevent overfitting and to ensure that the model version generalizes well to the validation data. If results deteriorate, or if signs of overfitting are detected, we can then revert to the model saved at best performance. This model presents an optimal generalization to the validation data and can be used directly for the test phase or for subsequent deployment.

To diagnose ResNet model results, the use of **Class Activation Maps** (CAMs) makes it possible to visualize which specific parts of the time series have contributed most to the model's prediction (see **Appendix H**). In the context of anomaly detection, this provides essential transparency as to the temporal features taken into account by the neural network. By applying CAMs to the ResNet model, we can identify the particular temporal segments that led to the classification of a sequence as normal or anomalous. **Figure 5-25** illustrates the application of CAM to the results of a 24-hour time sequence. The term "True label" represents the class assigned by the expert to this sequence, while "likelihood of label" indicates the probability, according to the model, that the sequence belongs to this true class. The blue curve corresponds to the raw time series, while the color palette, ranging from light yellow to dark red, illustrates the influence of each input point in the model's decision-making process.

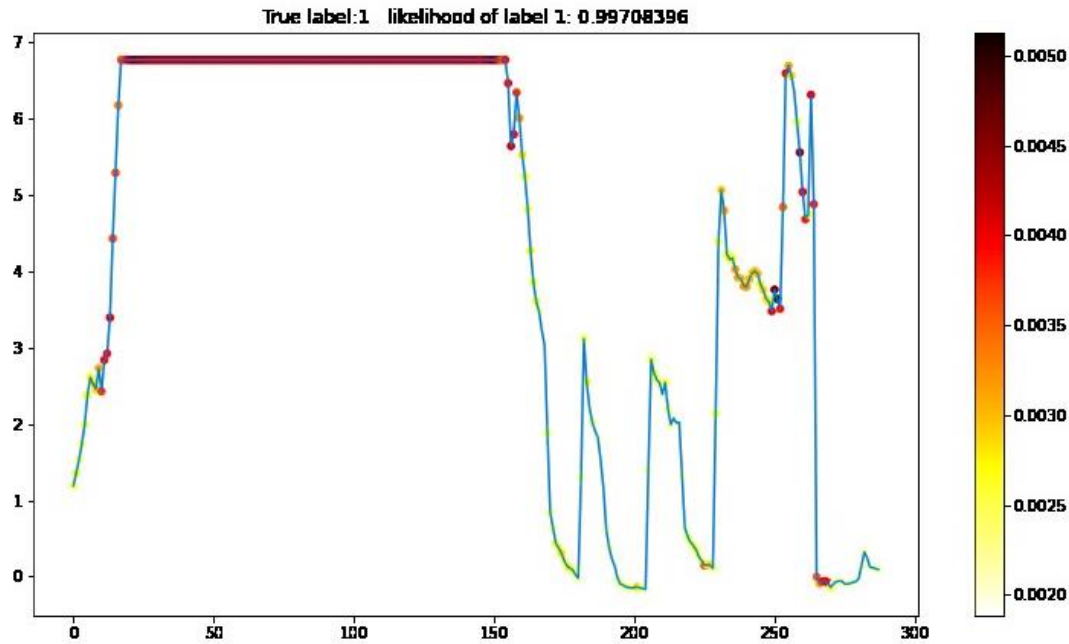


Figure 5-25: Class Activation Map of a sequence

#### 5.4.4.4 Potential improvements

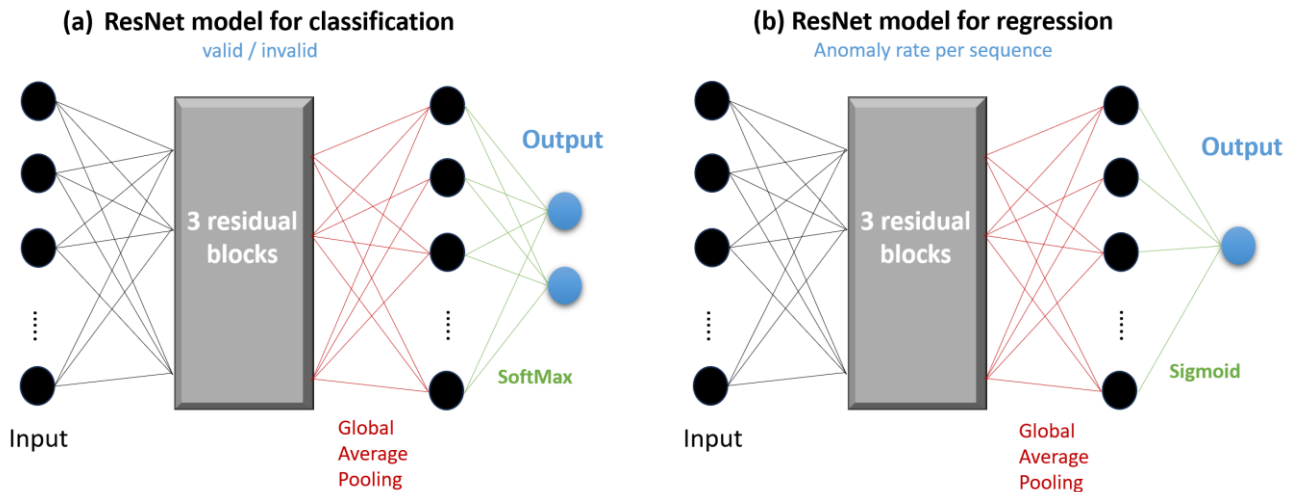
The ResNet model, originally designed for classification tasks, has been adapted for the detection of anomalies in time series, as described above. In its classic form, the model outputs two neurons with respective probabilities of belonging to a valid or invalid class, resulting from a supervised binary classification. This process involves the results of the manual validation elaborated by an expert, who assigns labels at each time step.

During the data pre-processing phase, we converted this labeling into a binary classification, using a threshold of 50% to declare a sequence valid or invalid. However, this condition seemed rather restrictive, as it considered a sequence to be valid even if 49% of its points were drifting. This led us to explore a **multiclass classification** approach with the aim of establishing less strict categories; valid, dubious and invalid. In this way, we will have 3 neurons instead of two at the output of the model. The aim was for the intermediate class to encompass all sequences that gravitate around the 50% limit. However, even with this approach, we need to define thresholds to separate the classes, which motivated sensitivity testing in this direction. This reflection also prompted us to consider bypassing the classification problem by adopting a direct prediction approach of the anomaly rate in the sequence, using a **regression approach**.

To implement this modification, the last layer of the model was adjusted. Rather than using a SoftMax function with two (or three) neurons as output, the model will be configured with a Sigmoid function and a single neuron as output. This transformation enables the model to



generate continuous predictions representing the level of anomaly in the sequence, offering a more nuanced and continuous approach than binary classification. These three approaches (see [Figure 5-26](#)) will be analyzed and compared as part of this research project.



**Figure 5-26: ResNet architectures used in this study**

### 5.4.4.5 Generalization to other sites

Unlike Matrix Profile, ResNet operates on a learning principle that can be deployed directly in the evaluation phase. In this way, the generalization of the model to other sites can be assessed by directly testing the saved pre-trained model on their respective data. However, due to the specificities of the tests, this approach may sometimes prove insufficient, requiring a "re-tuning" of the model with data from new sites. This operation can be carried out in the traditional way, or using Transfer Learning, a technique which aims to exploit the knowledge acquired by a model on one task to improve its performance on a similar task, without starting from scratch. Globally, there are three main Transfer Learning strategies (see [Figure 5-27](#)):

- **Total fine-tuning:** This approach involves taking a pre-trained model and completely re-tuning it for the new task, replacing the output layer with a new one adapted to the target task, if necessary. Then, the whole model is trained on the new data, allowing full adaptation of the model to the specifics of the new task.
- **Feature extraction:** In this strategy, the pre-trained model is used as a kind of feature extractor. We remove the output layer from the pre-trained model and freeze the weights of the other layers. In this way, the representations learned by the model on the first task can be used as inputs for the final layer that will be specifically trained for the target task.

## Chapter 5. Benchmarking models for data validation and anomaly detection

- Partial fine-tuning: This strategy combines the two previous approaches. However, instead of freezing all layers, some layers of the pre-trained model are left to thaw, allowing partial adaptation of the model to the new data while preserving certain features learned during pre-training. This strategy is useful when the new task is similar to the initial task but has unique features.

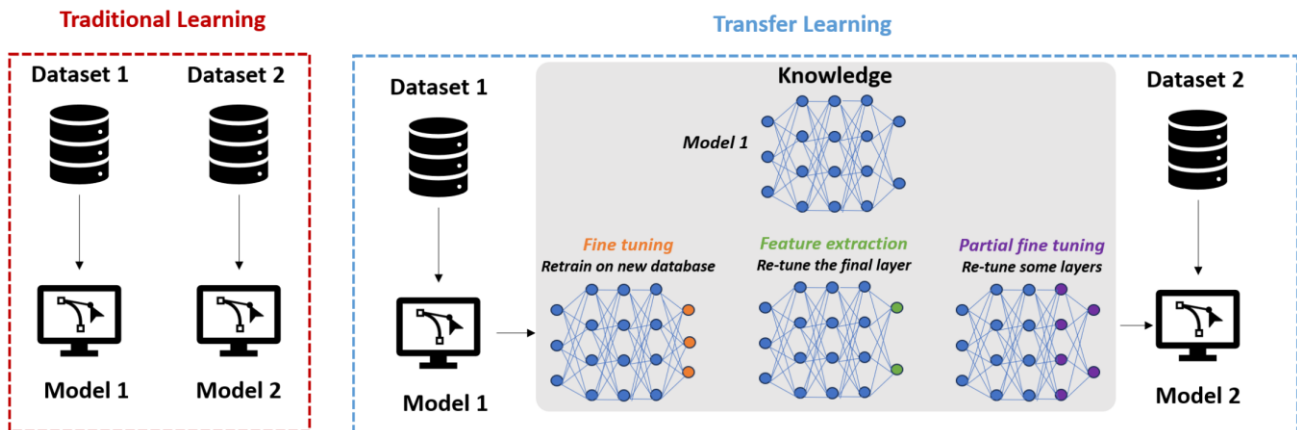


Figure 5-27: Transfer Learning strategies

### 5.4.5 Conclusion

The ResNet model was designed to overcome the problem of the gradient that disappears or explodes in deep architectures. It introduces residual connections, allowing the model to learn residual mappings instead of directly learning the underlying mappings. With regard to anomaly detection with ResNet, the model, originally developed for image processing, has been adapted to process time sequences as input.

The tests conducted involve several key aspects. In terms of sensitivity to input data, a sub-sequencing approach is applied during the preprocessing of time series data, dividing the original series into smaller fixed-length subsequences. Data enhancement techniques, such as up sampling, are employed to address class imbalance. Hyperparameter tuning focuses on optimizing the input window size and transforming labels from a probability to a label. Model training utilizes K-fold cross-validation for robust performance evaluation, while CAMs aid in interpreting results, visualizing which temporal segments contribute to the model's predictions. Potential improvements include exploring multiclass classification and a direct prediction approach of anomaly rates using regression. The ResNet model's adaptability to new sites is assessed, considering Transfer Learning strategies. These strategies allow the model to leverage previously acquired knowledge for improved performance on new tasks.

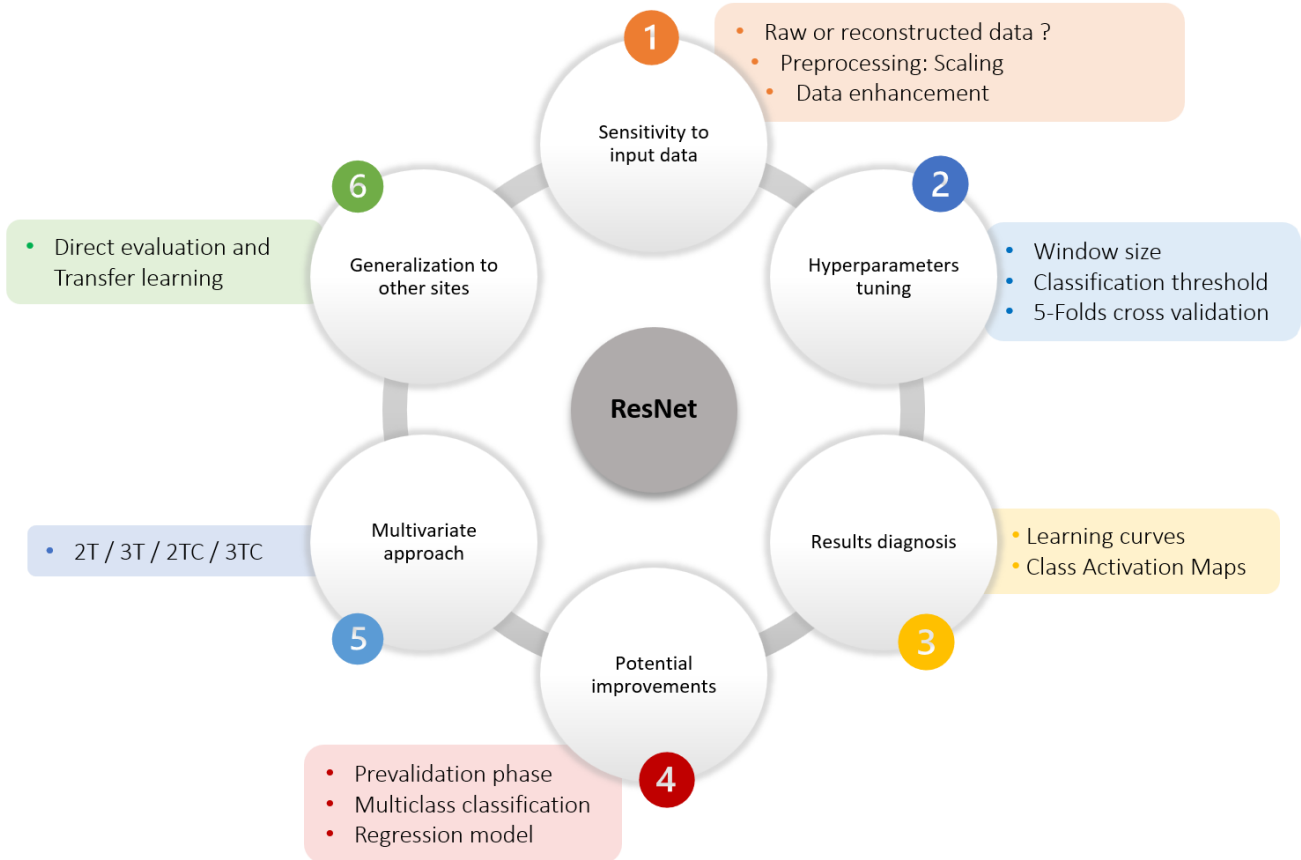
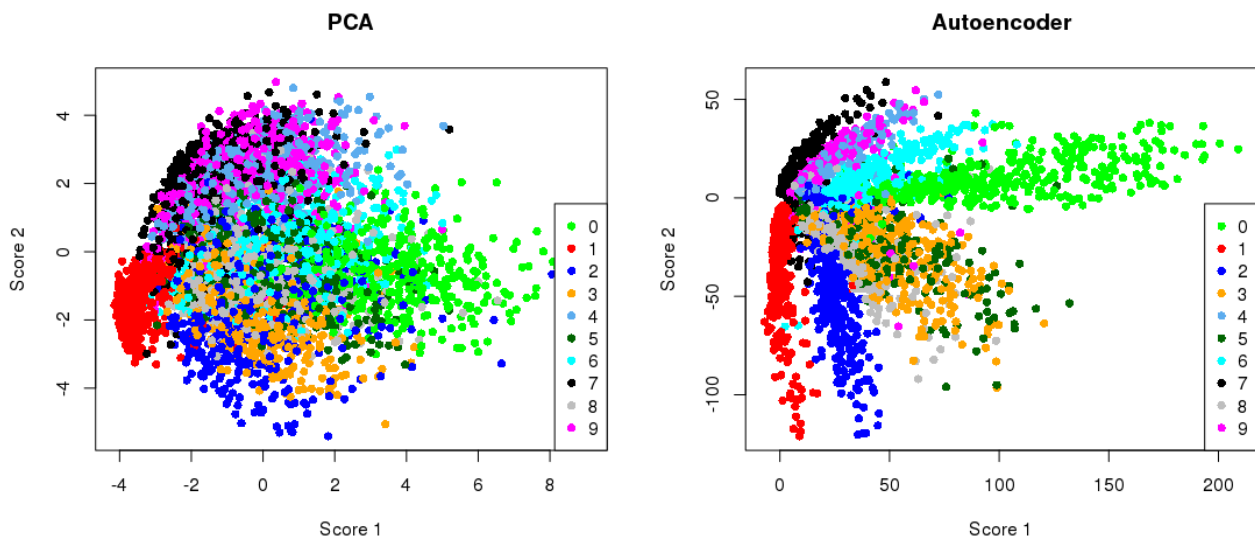


Figure 5-28: Overview of tests related to ResNet model

## 5.5 Autoencoder

### 5.5.1 Introduction and background

Autoencoders, originally introduced by Hinton in the 1980s [213], are well-known networks designed to reproduce their own inputs with minimal distortion. Addressing the challenge of "backpropagation without a teacher," autoencoders employ the input data as a form of self-guided teacher. This conceptually simple yet powerful approach leverages Hebbian learning rules [214], providing a foundational framework for unsupervised learning [215].

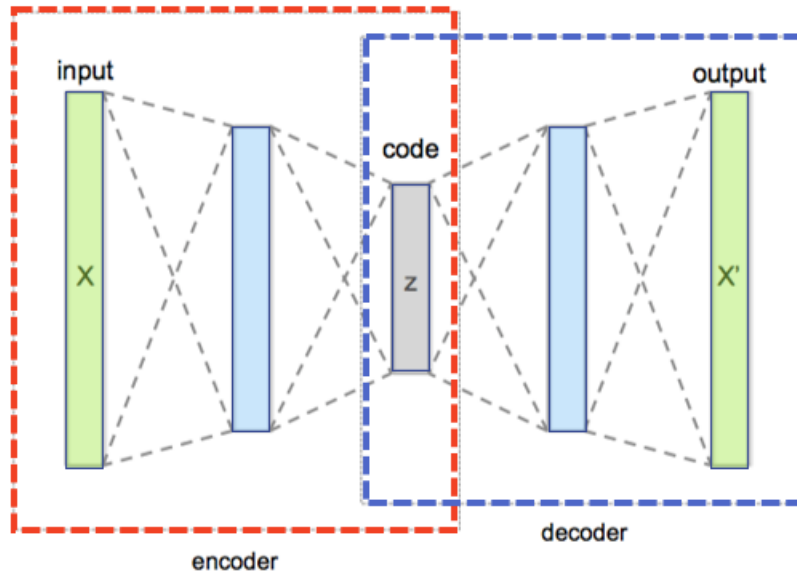


**Figure 5-29: Comparison of the results of dimensionality reduction using PCA and Autoencoder - © [216]**

Originally proposed as a nonlinear generalization of Principal Component Analysis (PCA) [217], autoencoders have since evolved to fulfil various purposes (see [Figure 5-29](#)). The key idea behind autoencoders lies in their ability to encode information in a compressed manner and faithfully reconstruct the original data from this latent representation. This autoencoding process not only facilitates the exploration of underlying data structures but also enables the detection of significant patterns and features. Their versatility is demonstrated in applications such as noise reduction [218], data instance generation [219], and notably, anomaly detection [162]. The adaptability of autoencoders to anomaly detection stems from their capacity to capture normal data structures while remaining sensitive to unusual variations. Furthermore, the concept of autoencoders has become widely utilized in the realm of generative model learning, extending their applicability to diverse fields [220]. The continuous development and application of autoencoders underscore their significance in DL, marking a substantial contribution to the evolution of unsupervised learning paradigms.

### 5.5.2 Definitions and notation

Autoencoders, at the heart of unsupervised learning, are part of the quest to optimally represent complex data. Basically, an autoencoder seeks to generate an output as similar as possible to the input presented to it. This task is performed by a neural network which basic architecture is presented in [Figure 5-30](#).



**Figure 5-30: Baseline of an autoencoder model**

**Definition 1:** Input and output

The main objective of an AE is to minimize the divergence between its output and its input, thus seeking to accurately reconstruct the original data. In concrete terms, this translates into a goal where the output target is aligned with the **input**, meaning that the number of neurons in the input layer is equivalent to the number of neurons in the **output layer**. This neural matching is intended to ensure that the autoencoder learns an internal representation that best preserves the essential features of the input during compression and reconstruction.

**Definition 2:** “Bottleneck” hidden layers

Between input and output, **hidden layers** follow, usually smaller in size than the input layers. This particular design is known as a bottleneck architecture. The idea behind this architecture is to force the model to learn a condensed, informative representation of the input data. By reducing the dimensionality of the hidden layers, the autoencoder is forced to capture the most salient and significant aspects of the data, as it must represent the information in a lower-dimensional space. However, this is not the unique possible architecture. There are different variations of autoencoders, including complete and sub-complete autoencoders. The former

maintains the same dimension for the hidden layer as for the input layer, while the latter reduces it. These layers can take different forms, ranging from fully connected layers to more complex architectures such as convolutional neural networks (CNN) or recurrent neural networks (LSTM).

### Definition 3: encoder

The **encoder** is the first half of the autoencoder. Its function is to transform the original input into a compressed representation, also known as latent space. This transformation is achieved by a series of neural layers that reduce the size of the input. The encoder thus captures the most important features of the input data, forming a condensed representation.

### Definition 4: latent space (code)

**Latent space**, also known as code, represents the hidden layer at the heart of the autoencoder. It is a representation of lower dimensionality than the input and is designed to encapsulate crucial information while reducing redundancy. The quality of this latent representation determines the autoencoder's ability to extract meaningful features from the data.

### Definition 5: Decoder

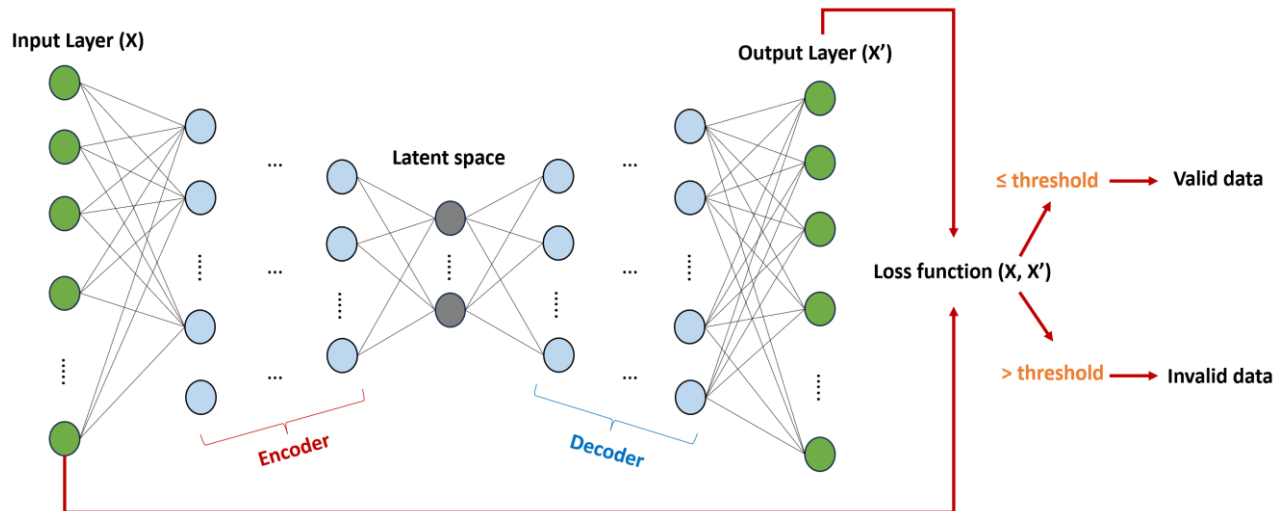
The **decoder** is the second half of the autoencoder. It takes the latent representation generated by the encoder and attempts to reconstruct it into an output that should be as close as possible to the original input. Like the encoder, the decoder consists of a series of neural layers, but this time it performs an inverse operation of increasing dimension.

During the learning process, all these components work in tandem in order to preserve as much information as possible between the input and the latent space. This preservation is crucial to enable the decoder to accurately reconstruct the original input.

### 5.5.3 Model architecture

Unlike tests on residual neural networks (ResNet), the use of autoencoders in our context has no typical framework that stands out when it comes to its use for anomaly detection in time series. The literature offers a variety of architectures for autoencoders, ranging from simple autoencoders (AE) to combined variants such as convolutional (CNN-AE) or recurrent (LSTM-AE) autoencoders. In our project, due to time constraints and the need to ensure a variety of heterogeneous models, we have decided to exploit deep autoencoders (deep-AE) and exclude structures based on CNNs that may be close to ResNets in terms of their baseline, as well as

architectures using LSTMs in view of their complexity. Preliminary tests were carried out in this regard, validating our deliberate choice. By opting for deep-AEs, we aim to maximize the model's ability to learn complex, hierarchical data representations, while simplifying architectural complexity for reasons of practicality and efficiency (see [Figure 5-31](#)).



**Figure 5-31: Deep autoencoder architecture for data validation**

The loss function measures the difference between the decoder output and the original input. It quantifies the reconstruction error and serves as a signal for adjusting the neural network weights during training. The aim is to minimize this loss function, enabling the AE to generate outputs close to the original input. The use of Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) as a loss function for autoencoders in the context of anomaly detection is a commonly adopted approach. These measures quantify the mean or quadratic deviation between the output generated by the autoencoder and the original input. In the case of anomaly detection, a low value indicates that the reconstructed output is similar to the input, suggesting that the given example conforms to the learned model. Anomalies, being rare and unusual instances, often generate significantly higher reconstruction errors. So, when a new input produces an MSE or RMSE significantly higher than normal, this may indicate the presence of an anomaly in the data. The threshold for declaring an input as abnormal may need to be adjusted according to the specific characteristics of the dataset. In our case study, MSE has been chosen as the main loss function of the model, while the RMSE is used as a tracking metric.

**Equation 7: Mean Squared Error (MSE)**

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

### Equation 8: Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Where:  $n$  is the number of samples,  $y_i$  is the input value and  $\hat{y}_i$  is the predicted output.

#### 5.5.4 Anomaly detection using AE

##### 5.5.4.1 Sensitivity to input data

Similar to the other models, sensitivity tests to input data will be conducted for the AE model, comparing raw and reconstructed turbidity.

In the data pre-processing stage, a sub-sequencing approach is applied, breaking down the input time series into small 24-hour subsequences. In addition, tests for **normalization or standardization** techniques are applied to the subsequences to scale the data to a common range and ensure comparable magnitudes between features. As with ResNet, given that the sequences contain sufficient temporal information, random mixing of subsequences is generally considered appropriate. This random permutation helps to avoid any inherent bias in the original dataset that might influence the learning process of the autoencoder model dedicated to detecting anomalies in time series.

Nevertheless, since AE is designed to reconstruct normal patterns, providing it with the entire set of input sequences might be inappropriate. Hence, we will assess its performance by comparing two scenarios: one involving **the full dataset and another where only fully valid data sequences are used for training**. The evaluation, on the other hand, will encompass all sequences to ascertain the model's ability to accurately identify anomalies.

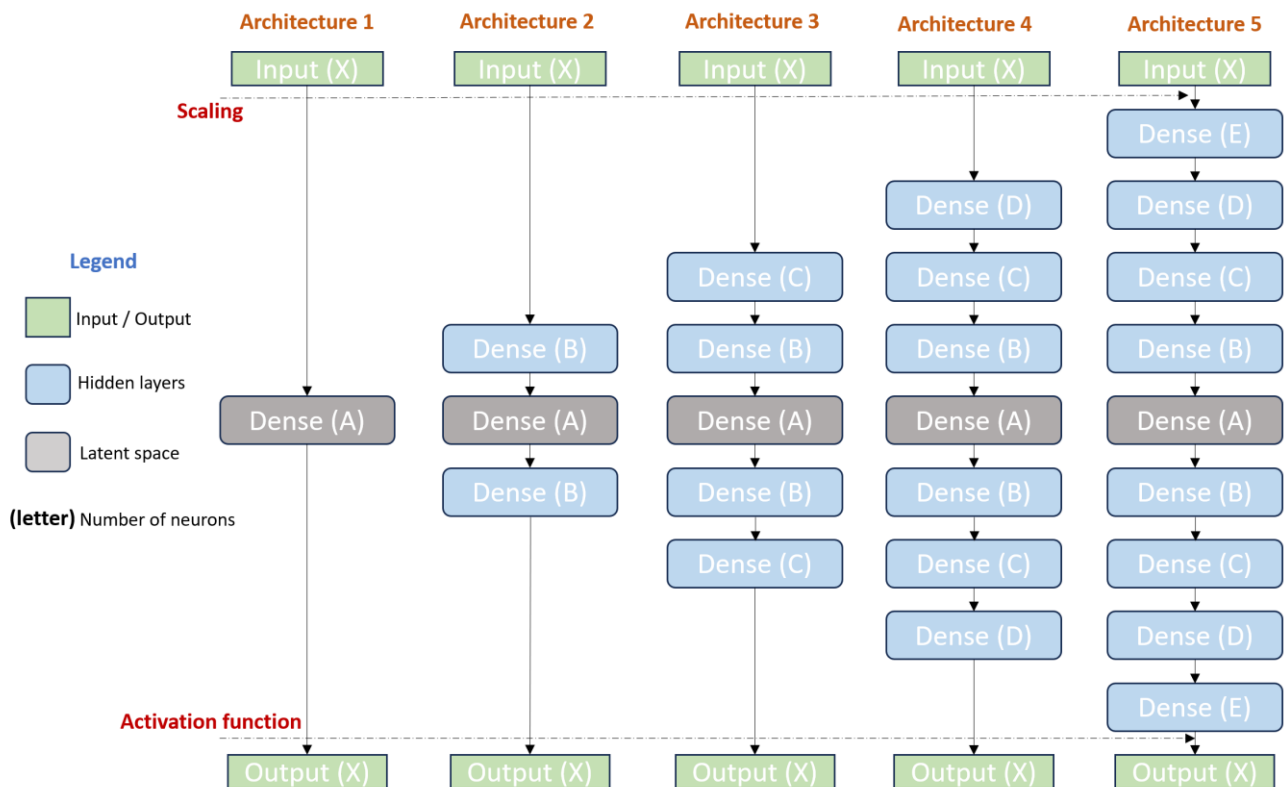
However, given the assumption that normality in sewer networks remains relatively stable in dry weather and that recurring patterns may be present in the input data, we will assess the sensitivity of the model to the size of the input base by reducing the number of samples used during training. This approach also aims to determine the number of additional sequences, if any, needed to improve learning. It is important to note that this approach is biased, as no pre-processing related to the separation between dry and rainy weather was applied. Thus, **sensitivity to database size** is assessed by randomly removing a set percentage of data.



### 5.5.4.2 Hyperparameters tuning

First of all, the segmentation of the input data into subsequences makes the window size a parameter to be tuned. Sensitivity tests will then be carried out to determine the **optimal window size**, guaranteeing adequate capture of significant patterns in the data.

Moreover, in the absence of a defined typical structure, **different architectures** were explored in our approach. A varied set of 5 architectures was tested to determine which best suits the specific nature of our data and the objectives of our project (see [Figure 5-32](#)).



**Figure 5-32: Different architectures of AE tested in this study**

For each architecture, various models were tested with different characteristics, such as the number of layers and neurons per layer. Apart from the tests conducted on input data and normalization approaches, as well as those examining the activation function of the final layer, we trained and evaluated 21 different AE models (see [Table 10](#)). However, certain elements were kept constant, namely the optimizer for parameter updates, specifically Adam, and the activation functions for intermediate layers, which were set as Rectified Linear Units (ReLU).

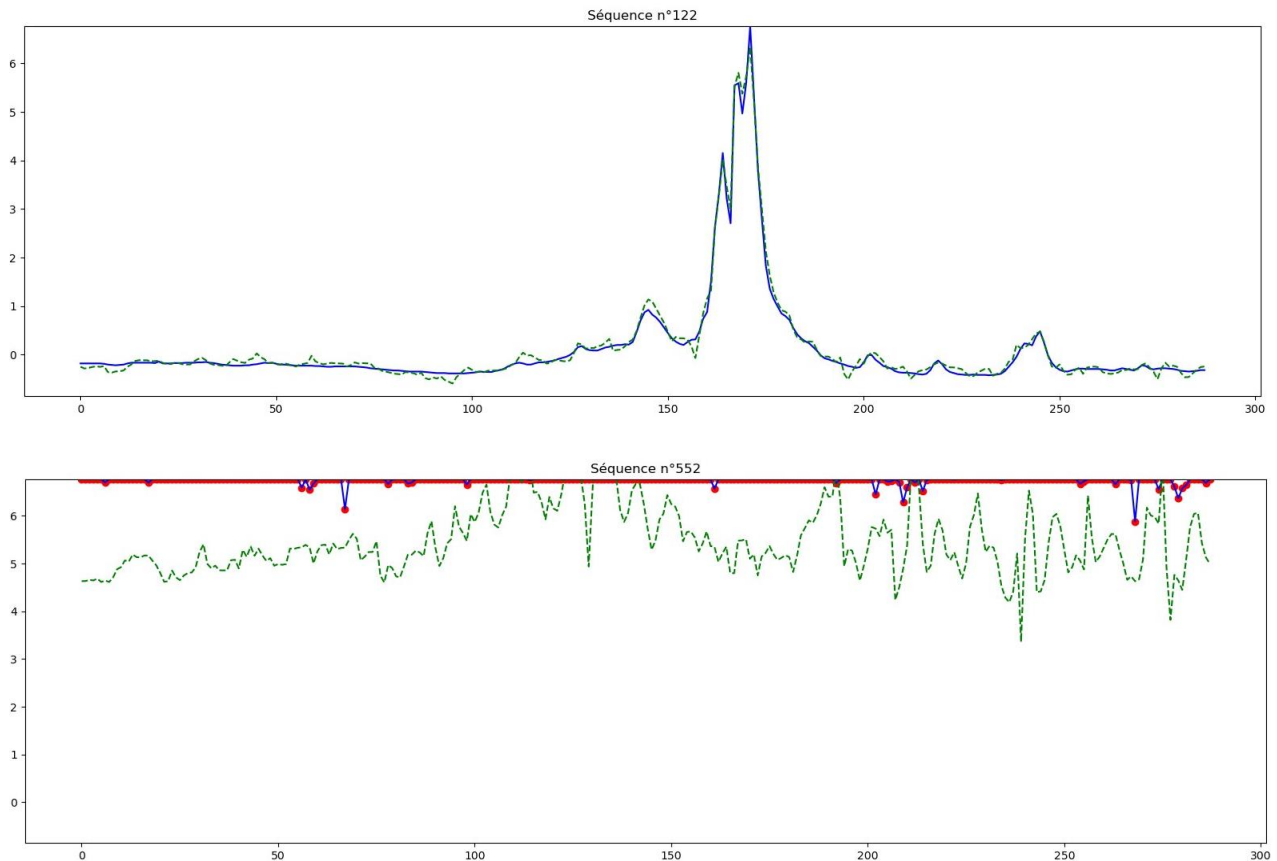
Table 10: List of AE models evaluated in this study

Architecture	N°	Model
1	1	Input → 4 → Output
	2	Input → 8 → Output
	3	Input → 16 → Output
	4	Input → 32 → Output
	5	Input → 64 → Output
	6	Input → 128 → Output
	7	Input → [96 x 32 x 96] → Output
	8	Input → [128 x 64 x 128] → Output
	9	Input → [144 x 72 x 144] → Output
	10	Input → [192 x 128 x 192] → Output
	11	Input → [80 x 64 x 80] → Output
2	12	Input → [96 x 64 x 96] → Output
	13	Input → [160 x 64 x 160] → Output
	14	Input → [192 x 64 x 192] → Output
	15	Input → [128 x 16 x 128] → Output
	16	Input → [128 x 32 x 128] → Output
	17	Input → [128 x 96 x 128] → Output
	18	Input → [128 x 112 x 128] → Output
3	19	Input → [192 x 128 x 64 x 128 x 192] → Output
4	20	Input → [192 x 128 x 92 x 64 x 92 x 128 x 192] → Output
5	21	Input → [216 x 168 x 128 x 72 x 64 x 72 x 128 x 168 x 216] → Output

### 5.5.4.3 Model training

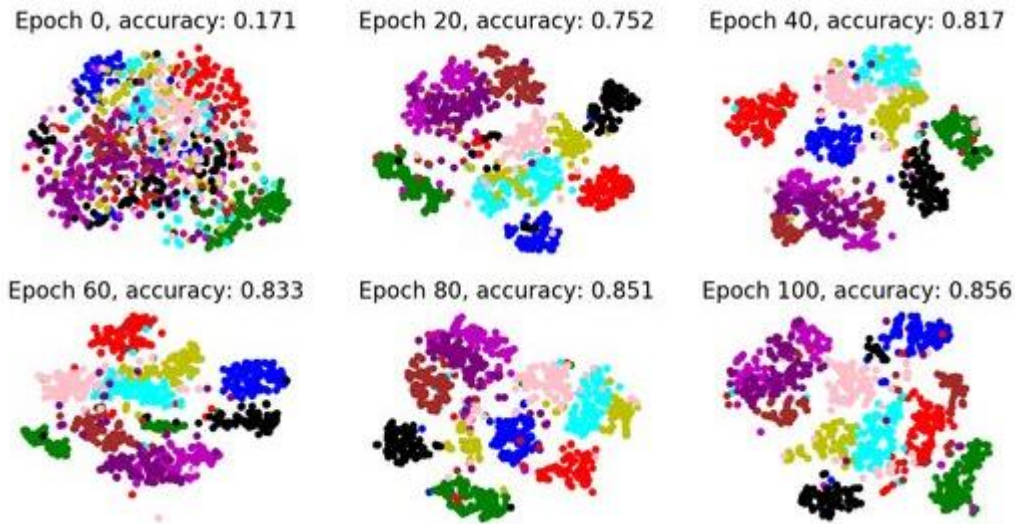
The AE model training process follows a similar approach to that of the ResNet model, with the consistent application of cross-validation and callbacks to optimize model performance.

In the context of anomaly detection, the comparison between input and output of the autoencoder plays a central role. The autoencoder aims to reconstruct input data in such a way as to minimize information loss. By comparing the original input with its reconstructed output, we can assess the discrepancy between the expected representation and that generated by the model. Anomalies, being unusual or divergent occurrences, can manifest themselves as significant differences between the input and output of the autoencoder. **Figure 5-33** shows an example of a normal sequence and an invalid sequence, as well as their reconstruction using AE.



**Figure 5-33: Example of output of the AE model. (Top) Valid sequence - (Bottom) Invalid saturation sequence. (Colors) in blue: raw data, in green: AE output, in red: anomalies**

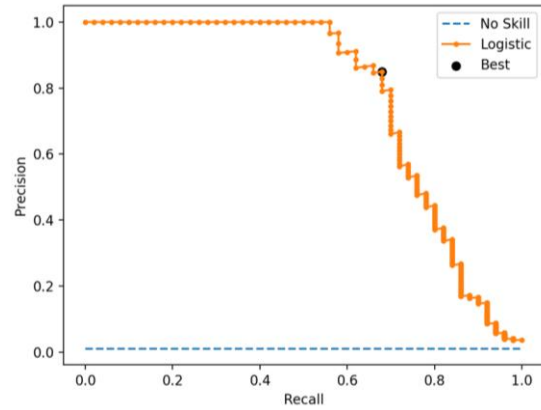
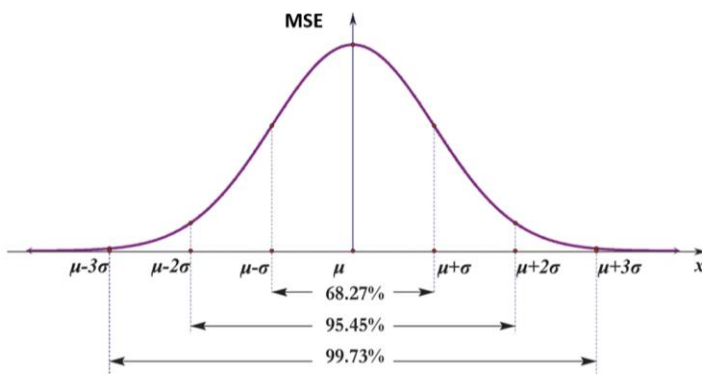
In order to compare different AE models, we can evaluate their performance in terms of sequence reconstruction or classification to identify the best performing model. However, in some situations, two models may perform equally well. It then becomes crucial to determine whether they are performing the same task. One approach is to compare their output sequences, or to use a reduced-space representation, facilitating interpretation by projecting their code into a space of reduced dimensions. To this end, we use a dimension reduction method called t-Distributed Stochastic Neighbor Embedding (t-SNE) [221]. This dimension reduction technique is used to visualize complex, high-dimensional data in a space of reduced dimensions, facilitating the observation of underlying patterns or structures. Compared with standard PCA, t-SNE offers significant advantages by preserving non-linear structures and local relationships between data (see [Appendix I](#)). If the models are similar in their reconstruction task, we expect their t-SNE projections to show similarities in reduced space. On the other hand, significant discrepancies in the distribution of points may indicate differences in the way the models learn and represent the underlying structures of the data. This provides a visual and interpretable perspective on the similarity or divergence of autoencoding models (see [Figure 5-34](#)).



**Figure 5-34: Latent space visualization of the same model at different epochs - [222]**

Unlike ResNet, where the task consists of classification, the autoencoder is intrinsically linked to the detection of anomalies in time series by evaluating the reconstruction error. Thus, the threshold plays a central role in decision-making. Different methods can be explored to establish an optimal threshold (see [Figure 5-35](#)).

The first approach is rooted in the statistics of the reconstruction error, utilizing a 3-sigma rule: any sequence with an MSE greater than the mean MSE plus 3 times the standard deviation is deemed anomalous. Alternatively, we adopted an approach based on maximizing the classification score, with a specific focus in our case on the F1 score (refer to [Section 6.1.1](#)). The approach involves testing different decision thresholds and calculating the F1 score corresponding to each threshold. The optimal threshold is the one that produces the highest F1 score. Once this threshold has been determined, it is used to classify examples as normal or abnormal in the test phase. Consequently, the autoencoder training process goes beyond the simple optimization of network weights and involves a thorough understanding of the characteristics of reconstruction errors. This makes it possible to define a relevant threshold, thus making the test phase crucial for validating the effectiveness of the autoencoder approach in this specific context.



**Figure 5-35: Classification approaches: (Left): 3-sigma rule - (Right): PR Curve approach**

#### 5.5.4.4 Potential improvements

Once the best autoencoder models have been identified, the possibilities for improvement focus mainly on exploiting their advantages by combining their outputs using an **ensemble model**, similar to the approach adopted for Matrix Profile. Two approaches are possible: firstly, to exploit the respective mean square errors (MSE) directly, calculating an average MSE of the different models to classify sequences as valid or invalid according to an adjusted threshold. Secondly, merge the models after classification using majority voting or consensus, which would mean invalidating a sequence only if all models invalidate it equally.

Another area for improvement is to optimize the classifier output of the AE to ensure that the **classification rules** are appropriate. For example, the relevance of the 3-sigma threshold in this context may be questioned. In addition, the comparison between the model and the manual validation implicitly assigning a label to the sequence, generally the majority label with respect to the points that make it up, could be improved. Sensitivity tests to the threshold applied to the reference will therefore be carried out.

A last technique for improving model results concerns the **acquisition of new data**. Although this is not possible in our case, if such acquisition were feasible, it would be interesting to assess whether we can estimate the amount of data needed to achieve optimal convergence of the autoencoder model. All these questions will be examined through in-depth sensitivity tests.

#### 5.5.4.5 Generalization to other sites

A straightforward approach is to evaluate the model on data from other sites, thus measuring its performance in a variety of contexts. However, due to the specificities of each site, this approach can prove insufficient. To address this limitation, a complementary strategy would

be to re-train the model using all sites data. This process could allow the model to have a wider vision of normality, thus improving its adaptability. With this in mind, it might be worth considering the possibility of tuning site-specific models, taking into account the unique characteristics of their respective data.

### 5.5.5 Conclusion

The autoencoder (AE) is a type of unsupervised deep learning model that can be used to detect anomalies in time series. Its principle is based on learning a compressed representation of the input data, called a "latent code", and reconstructing the data from this code. The more typical the input data are, the better the reconstruction, so a bad reconstruction denotes atypical input data.

Evaluation of anomaly detection using an AE involved various tests. Firstly, sensitivity analyses to input data will be assessed by comparing raw and reconstructed turbidity, considering two scenarios: one using all data and the other limited to sequences fully valid for training. Next, hyperparameter sensitivity tests will be carried out to determine the optimal size of the input window, guaranteeing adequate capture of significant patterns. Regarding the AE architecture, different configurations are explored, and 21 distinct AE models were evaluated. The AE model training process followed a similar approach to that of the ResNet model, with the consistent application of cross-validation and callbacks to optimize model performance. Finally, prospects for improvement will be explored, including the exploitation of AE model ensembles and the optimization of the classification threshold, raising questions about the relevance of the 3-sigma threshold. The adaptability of the model to other sites will be assessed, with a view to direct evaluation and potential retraining with data from other sites to take account of their specific normality.

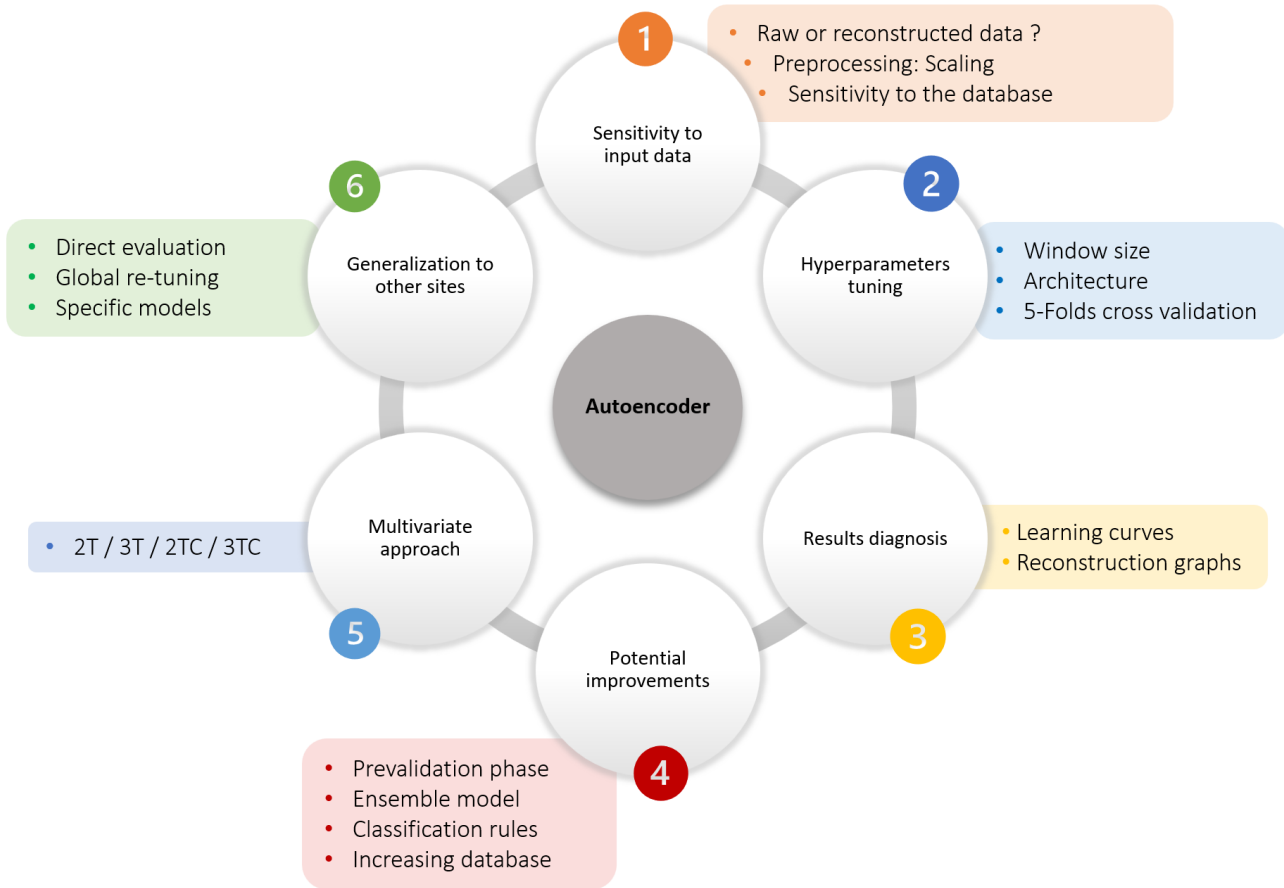


Figure 5-36: Overview of tests related to Autoencoder model

### 5.6 Synthesis of Chapter 5

The benchmark section aims to introduce our three models to be tested, namely Matrix Profile, ResNet, and Autoencoders, for data validation and time series anomaly detection. Nevertheless, a first question merged, necessitating to explore the usefulness of traditional statistical approaches (3-sigma rule). This investigation calls into question the relevance of classical approaches to the complexity of temporal data. The results justified the use of more sophisticated approaches based on AI.

Hence, the objective of this work is to look at the effectiveness of our selected models. The latter have demonstrated their usefulness in various fields, but their direct applicability to the detection of anomalies in wastewater networks time series requires in-depth evaluation. In this research work, a wide-ranging strategy was adopted to explore a spectrum of models using different approaches (see Figure 5-37). Given the complexity of obtaining a reliable reference for model training, particularly in terms of time, cost and subjectivity, the use of unsupervised approaches proved obvious. This is why we chose to test both a "simple" model, namely Matrix Profile, and a more sophisticated model based on a deep autoencoder. However, given that evaluating model performance inevitably requires a reliable reference, a fair investment has been made to obtain one. Consequently, it seems appropriate to take advantage of this reference by using a supervised anomaly detection model, in this case ResNet.

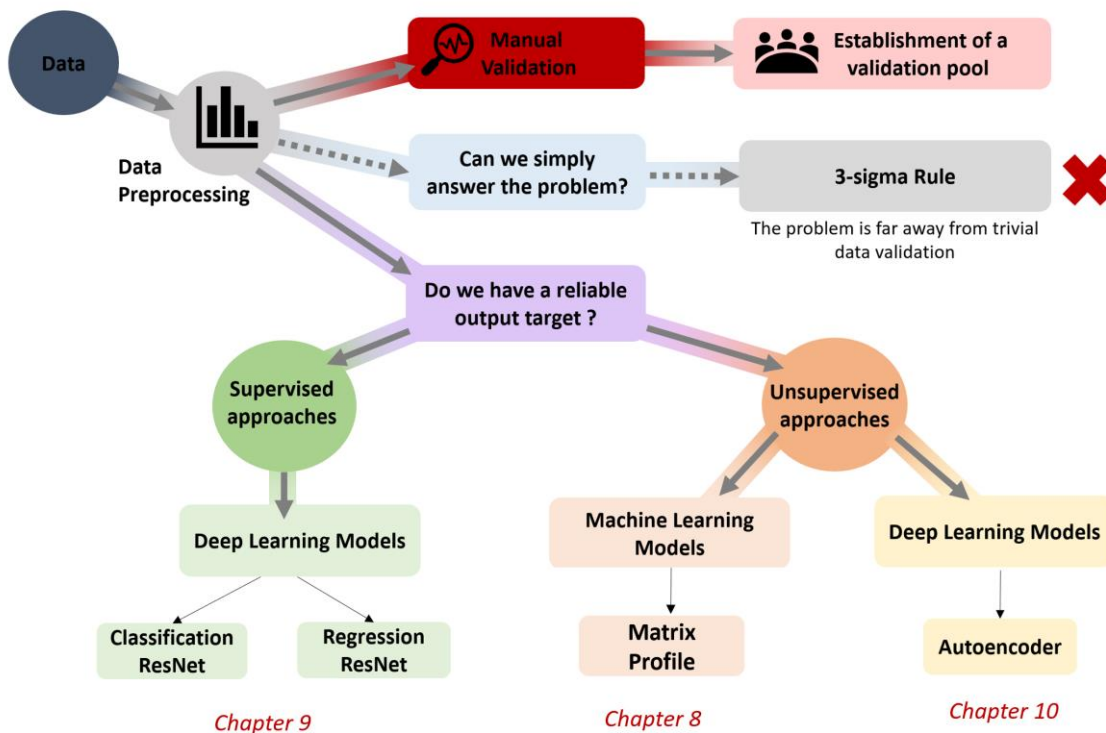


Figure 5-37: Benchmark of the model tested for anomaly detection in turbidity data



## Chapter 5. Benchmarking models for data validation and anomaly detection

Experiments will use turbidity data from the Cottage as a typical site. Sensitivity assessments are conducted to determine the optimal use of input data, considering raw measurements or reconstructed turbidity. Challenges related to temporal regularity, missing data, and time series scaling are addressed. Normalization and standardization techniques are explored, and sensitivity tests on the input database are performed, considering the entire dataset, pre-selection, or acquiring additional data for improved learning. Hyperparameter tuning, crucial for optimal model performance, involves calibrating parameters such as sequence size and stride, along with model-specific parameters like the number of layers and neurons per layer. The optimization procedure, utilizing grid search, systematically explores the configuration space. Following model identification, the results diagnosis phase involves a thorough analysis, comparing model outcomes with expert findings to identify strengths and limitations. Visualization approaches are employed for a detailed understanding, leading to model-specific improvements. A multivariate approach, leveraging on-site sensors, is adopted to provide additional data for each timestamp. The generalization to other sites is systematically assessed to gauge adaptability and performance across diverse contexts, offering valuable insights for broader applicability.

**Table 11: Overview of different tests**

	Matrix Profile	ResNet	Autoencoder
<b>Sensitivity to input data</b>	<ul style="list-style-type: none"> <li>Raw or reconstructed               <ul style="list-style-type: none"> <li>No scaling</li> </ul> </li> <li>Preprocessing: missing data imputation, downsampling, smoothing</li> </ul>	<ul style="list-style-type: none"> <li>Raw or reconstructed</li> <li>Preprocessing: scaling               <ul style="list-style-type: none"> <li>All or preselection</li> </ul> </li> <li>Data enhancement</li> </ul>	<ul style="list-style-type: none"> <li>Raw or reconstructed</li> <li>Preprocessing: scaling               <ul style="list-style-type: none"> <li>All or preselection</li> </ul> </li> <li>Sensitivity to database size</li> </ul>
<b>Hyperparameters tuning</b>	<ul style="list-style-type: none"> <li>Window size</li> <li>Number of anomalies</li> </ul>	<ul style="list-style-type: none"> <li>Window size</li> <li>Classification threshold</li> </ul>	<ul style="list-style-type: none"> <li>Window size</li> <li>Model architecture</li> </ul>
<b>Potential improvements</b>	<ul style="list-style-type: none"> <li>Prevalidation</li> <li>Ensemble model</li> </ul>	<ul style="list-style-type: none"> <li>Prevalidation</li> <li>Multiclass classification               <ul style="list-style-type: none"> <li>Regression</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Prevalidation</li> <li>Ensemble model</li> <li>Classification rules</li> <li>Increasing database</li> </ul>
<b>Multivariable</b>		<ul style="list-style-type: none"> <li>2T / 3T / 2TC / 3TC</li> </ul>	
<b>Generalization to other sites</b>	<ul style="list-style-type: none"> <li>Direct evaluation</li> </ul>	<ul style="list-style-type: none"> <li>Direct + transfer learning</li> </ul>	<ul style="list-style-type: none"> <li>Direct + specific models</li> </ul>

# Chapter 6. Beyond data and models: Performance Metrics and Hardware-Software Configuration

This chapter delves into the realm beyond the mere analysis of data and the construction of models. In this section, our focus expands to encompass performance metrics and hardware-software configuration. Understanding the efficacy of data-driven models requires a nuanced exploration of the metrics that quantify their success.

## 6.1 Model's performance metrics

The evaluation of AI models dedicated to anomaly detection allows measuring their performance and reliability in various contexts, whether supervised or unsupervised. Whatever the approach adopted, it is essential to compare the model's results with a reference, in this case obtained by manual validation (filtering + expertise + aggregation). This comparison enables us to assess the effectiveness of the model by distinguishing correctly identified anomaly cases from those that are neglected or incorrectly classified. In this evaluation context, several metrics may be considered.

### 6.1.1 Confusion Matrix

Even if it's not a metric per se, the confusion matrix is one of the key concepts for binary classification. With a tabular visualization, it faces the model predictions to the ground truth labels. The latter is the result of the expert validation. The template for any binary confusion matrix is a 2x2 matrix with four kinds of results (see [Table 12](#)).

**Table 12: Confusion Matrix**

		Predicted Label	
		Positive (PP)	Negative (PN)
Actual Label	Total population = P + N		
	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

The positive term means the feature that we are trying to extract from the data. Here, it corresponds to an anomaly / invalid data. The diagonal elements of this matrix denote the correct prediction for different classes, while the off-diagonal elements denote the samples

which are mis-classified. Once the confusion matrix is established, different metrics can be calculated.

### 6.1.1.1 Accuracy

Accuracy is the simplest metric for binary classification. It represents the proportion of correct predictions (both true positives and true negatives) among the total population (see [Equation 9](#)). However, for anomaly detection problems, we assume that discords are minority. Hence, even if the model does not detect any anomaly, accuracy would remain important. Thus, this metric was not used in this work due to the unbalanced nature of the problem.

#### Equation 9: Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

### 6.1.1.2 Precision and recall

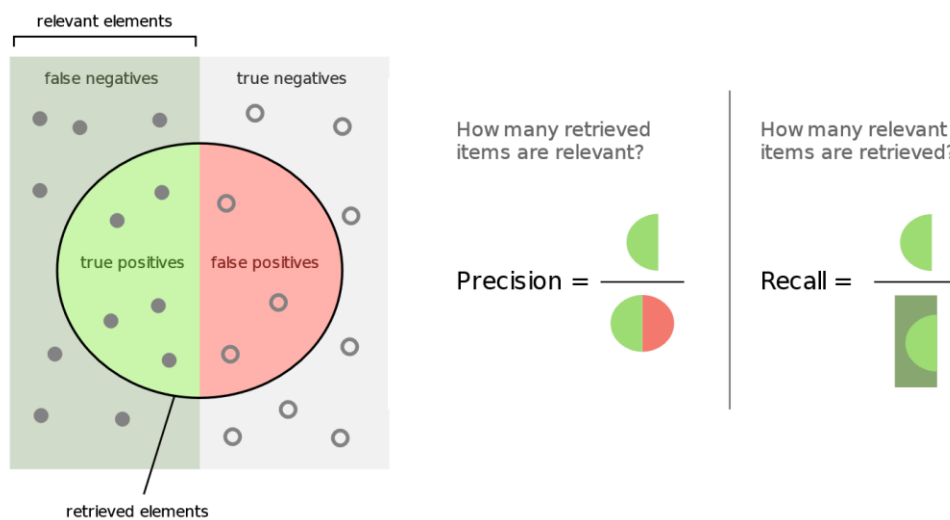


Figure 6-1: Diagram representation of precision and recall - © [223]

Precision is the fraction of true positive instances from all the positive instances that were retrieved by the model (see [Equation 10](#)). It evaluates, by complementarity, the rate of false alarms.

#### Equation 10: Precision

$$Precision = \frac{TP}{TP + FP}$$

On the other hand, the recall (also called sensitivity) represents the fraction of positive instances identified by the model from all the real positive instances (see [Equation 11](#)). It evaluates, by complementarity, the rate of missed anomalies.

**Equation 11: Recall**

$$Recall = \frac{TP}{TP + FN}$$

A perfect model will provide answers with precision and recall equal to 1 (the model finds all relevant instances and makes no errors). In practice, classification algorithms have varying degrees of precision and recall. In borderline cases, a model that identify the whole dataset as abnormal will have a recall of 1 but poor precision, while a model that identify a unique discord period will have a precision of 1 for a very low recall. The value of a classifier is therefore not reduced to a good score in precision or recall, but both.

**6.1.1.3 F score**

The F1 score is a harmonic mean of the precision and the recall equally weighted (see [Equation 12](#)). In the case where false alarms or missed anomalies are not of equal interest, this score can be pondered using a predefined ratio  $\beta$ , such that recall is considered  $\beta$  times as important as precision. In our context, false positives (false alerts) and false negatives (misfires) are equally important. Thus, we mainly use the F1 score giving the same weight to these two variables. Tests were nevertheless carried out by varying this coefficient (see [Section 9.2.2.1](#)).

**Equation 12:  $F_\beta$  Score**

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

**6.1.2 Matthew's Correlation Coefficient**

The F1 score, despite its popularity, may produce optimistic results, particularly in the context of datasets with a positive class imbalance. Firstly, F1 exhibits variability upon interchanging class labels, such that the positive class being relabeled as negative and vice versa alters the score. Moreover, the F1 score remains unaffected by the accurate classification of samples as negative, since it does not account for the true negative (TN) in the confusion matrix. Recent investigations by various researchers have underscored limitations associated with the F1 measure [224]. [225] asserts that alternative metrics should be employed due to fundamental conceptual flaws in the F1 score.

Offering an alternative resilient to imbalanced datasets, the Matthews Correlation Coefficient (MCC) exploits the confusion matrix to calculate the Pearson product-moment correlation coefficient [226] between actual and predicted values. The MCC is expressed as follows in [Equation 13](#).

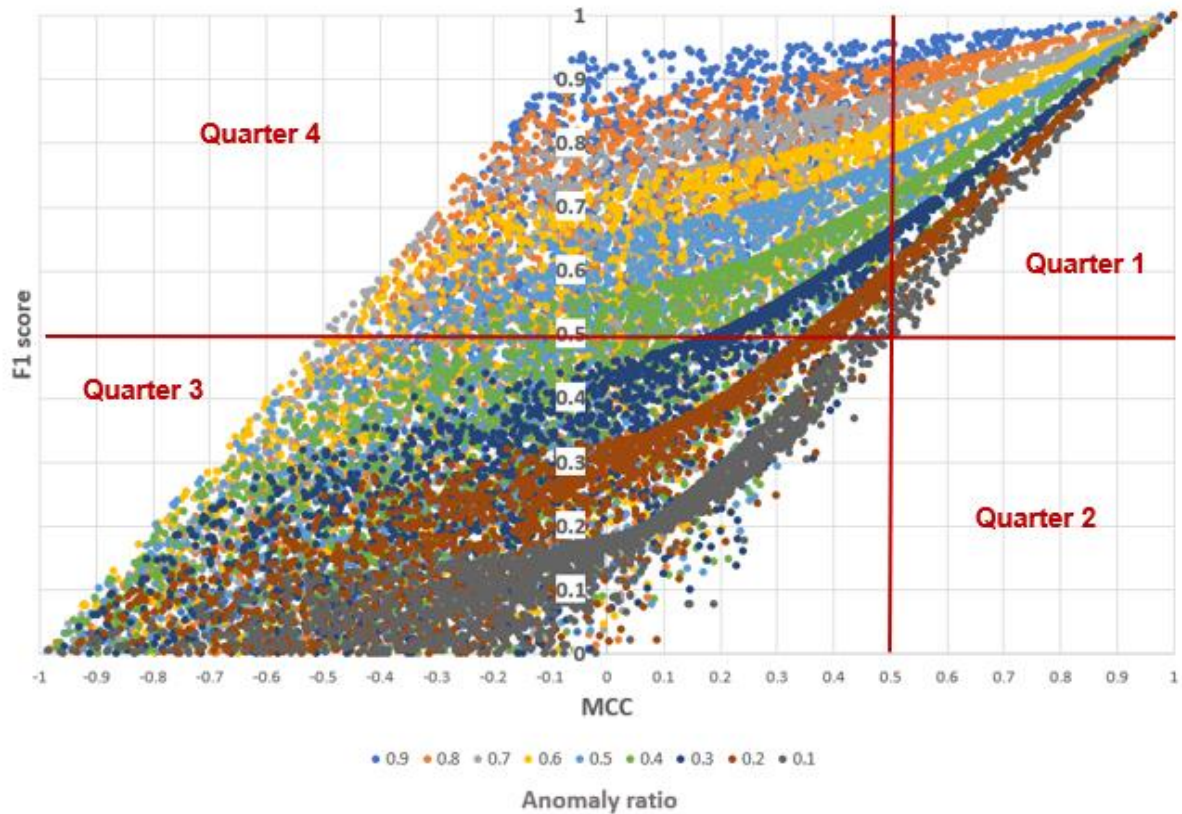
## Equation 13: Matthew's Correlation Coefficient

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

The MCC serves as a binary classification metric that yields a high score only when the model correctly predicts the majority of positive and negative instances. Its range spans from -1 to +1, with extreme values indicating perfect misclassification (-1) and perfect classification (+1), while MCC=0 represents the expected value for a classifier akin to random coin tossing [227].

Although the Matthews Correlation Coefficient (MCC) offers significant advantages, the F1 score remains the most widely used metric among researchers for classification tasks. However, a problem emerges, as there are many situations where the MCC and F1 score values diverge, making it difficult to draw correct conclusions about the behavior of the classifier under study.

Similar to [225], we present a scatterplot illustrating the Matthews correlation coefficients (MCCs) and F1 scores across 2000 potential confusion matrices generated from a synthetic dataset comprising 1000 samples with varying anomaly ratios (see [Figure 6-2](#)). Our observation reveals a reasonable concordance between the two metrics; however, the scatterplot cloud exhibits significant width. This suggests that for each F1 score value, there exists a corresponding range of MCC values, and vice versa, albeit with varying widths. According to [228], for any given F1 value ( $F1=x$ ), the MCC fluctuates within the interval  $[x-1, x]$ , indicating a fixed width of variability (1) irrespective of the specific value of  $x$ . Conversely, when considering a fixed MCC value ( $MCC=y$ ), the F1 score can span the range  $[0, y+1]$  if  $y \leq 0$  and  $[y, 1]$  if  $y > 0$ . In this case, the width of the range is determined by the expression  $1-|y|$ , signifying a dependence on the MCC value  $y$ .



**Figure 6-2: Relationship between MCC and F1 score, depending on the anomaly ratio.**

In general, the F1 score and the Matthews correlation coefficient (MCC) exhibit consistent agreement in their scores for predictions that accurately classify both positive and negative instances (Quarter 1), as well as for predictions that inaccurately classify both positive and negative instances (Quarter 3). However, these metrics display divergent behaviors when the prediction excels in only one of the two binary classes. Specifically, when a prediction yields numerous true positives but few true negatives, the F1 score can be misleading (Quarter 4), whereas the MCC consistently produces results that accurately reflect the overall issues with the prediction.

Indeed, [229] has claimed that the combined use of these two metrics "provides more realistic estimates of real-world model performance". Consequently, we calculated both metrics for all our tests: The F1 score allows us to focus on the task of interest, i.e. the identification of anomalies (positive instances), while the MCC allows us to assess the overall performance of the model, freeing us from any bias linked to the imbalance in class prediction.

### 6.1.3 Characteristic curves

#### 6.1.3.1 ROC Curve

Similarly to the confusion matrix, the receiver operating characteristic curve (ROC Curve) is not a metric per se. In fact, it is a plot that shows the performance of a binary classifier as a function of its cut-off threshold. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings (see [Equation 14](#) & [Equation 15](#)). This approach is only possible for probabilistic models where a score is calculated, indicating the probability to belong to a certain class. Then the model varies the threshold and computes the FP and TP according to the actual label. Each threshold is a point on the curve (see [Figure 6-3](#)).

#### Equation 14: True Positive Rate

$$TPR = recall = \frac{TP}{TP + FN}$$

#### Equation 15: False Positive Rate

$$FPR = \frac{FP}{FP + TN}$$

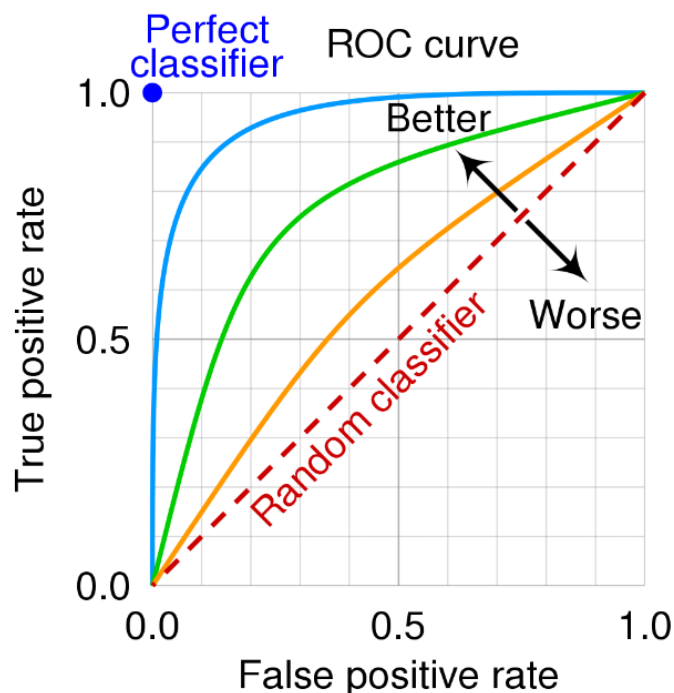


Figure 6-3: ROC Curve

- In (0, 0) the classifier classifies all negative: there are no false positives, but also no true positives.
- In (1, 1) the classifier classifies all positive: there is no true negative, but also no false negative.

- A random classifier will draw a line from (0, 0) to (1, 1).
- At (0, 1) the classifier has no false positives and no false negatives, and is therefore perfectly accurate, never being wrong.
- In (1, 0) the classifier has no true negative and no true positive, and is therefore perfectly inaccurate, always being wrong. It is enough to invert its prediction to make it a perfectly exact classifier.

### 6.1.3.2 PR Curve

Another way of analyzing precision and recall simultaneously is by means of Precision-Recall curves (PR curve). This curve visualizes the relationship between precision (positive predictive value) and recall (sensitivity) at different classification thresholds. By analyzing this curve (see [Figure 6-4](#)), we can see how the model balances precision and recall by adjusting the decision thresholds. An ideal model would present a curve heading towards the top right-hand corner of the graph, reflecting an optimal balance between precision and recall. Several key observations regarding the curve:

- Point 1 aligns with a threshold of 1.
- Point 3 aligns with the threshold of 0.
- Point 4 aligns with a threshold within the range of (0, 1).
- Point 2 aligns with an ideal model.

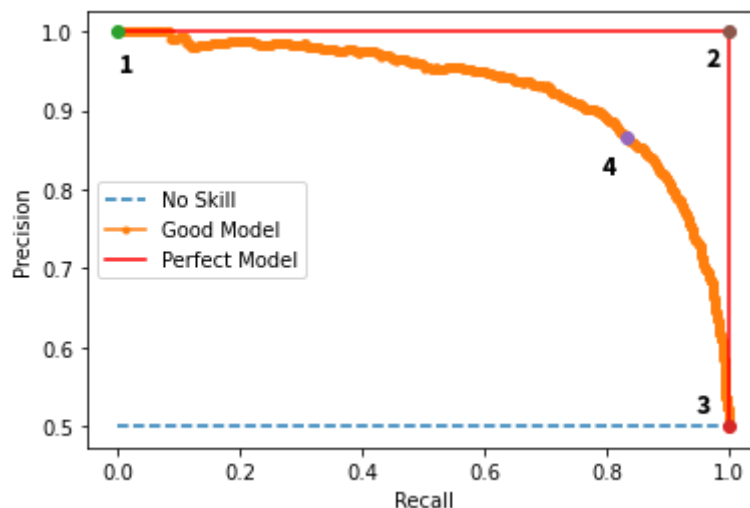


Figure 6-4: PR Curve

The baseline of a Precision-Recall (PR) Curve is subject to variations with changes in class imbalance, unlike the Receiver Operating Characteristic (ROC) Curve. This distinction arises from the direct impact of class imbalance on the precision of a No-Skill model. A "No-skill model" is a term used to denote a basic model that only predicts the majority class. The PR curve of a "No-skill model" is generally represented by a straight horizontal line at the level of



the precision of the majority class in the data set, although in reality the performance of such a model corresponds more to point 3. In this case, precision is equal to the number of true positives divided by the total number of positive predictions. As we always predict the majority class, precision will be equal to the proportion of true positives in the data set, which is simply the frequency of the majority class. On the other hand, recall will be equal to the number of true positives divided by the total number of true positives in the data set. Since we always predict the majority class, the number of true positives will be equal to the total number of true positives in the data set. Consequently, recall will be equal to 1, as all true positives will be captured.

Hence, the ROC curve is generally chosen (and indeed more commonly used) for an intuitive interpretation of the model's discriminative ability when compared to a random guessing model, while the PR curve is better suited to assessing performance on unbalanced datasets, focusing on the model's precision and recall.

### 6.1.3.3 Area under curve (AUC)

The area under the curve (AUC), is a metric that aggregates the measure of performance of a model on all possible threshold values. In practice, it calculates the area under the ROC curve or the PR Curve, where a higher value indicates a better ability of the model to discriminate between classes. The AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.

## 6.2. Annotator agreement metrics

The aim of this section is to assess the validation variance that exists among the multiple experts within the validation pool, along with an examination of their corresponding agreement rates. The evaluation of model performance often relies on the expertise and judgment of individuals within a validation pool, and understanding the degree of consensus or divergence among these experts is crucial for refining and optimizing the validation process. By quantifying the validation variance, we seek to uncover the extent to which interpretations and assessments may differ among experts, providing insights into the robustness and reliability of the validation outcomes. This analysis not only contributes to a comprehensive understanding of the validation dynamics but also lays the groundwork for interpretation of the overall process of the AI models, which takes the manual validation as a baseline. When evaluating annotators in the validation pool, different metrics can be employed, each offering a unique perspective on inter-annotator agreement.

### 6.2.1. Cohen's kappa coefficient

The most basic method for examining inter-annotator agreement involves calculating the observed proportion of instances where the raters concur. However, this approach is inherently flawed, as it does not account for the possibility that some level of agreement might occur purely by chance [230]. To address this limitation, [231] introduced a method for correcting the agreement between annotators by the likelihood of chance agreement. The resulting metric, known as Cohen's Kappa, is widely employed to assess agreement among evaluators in tasks involving subjective judgment. The formula for calculating this coefficient is as follows (Equation 16):

Equation 16: Cohen's kappa coefficient

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

where  $p_o$  is the relative observed agreement among raters, and  $p_e$  is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly seeing each category. Considering the confusion matrix for binary classifications, the Cohen's Kappa formula can be written as in Equation 17:

Equation 17: Cohen's kappa coefficient for binary classification

$$\kappa = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)}$$

However, there is considerable debate regarding the utility of kappa statistics for evaluating rater agreement. [232] assess that:

- kappa should not be considered the unequivocal standard for quantifying agreement,
- concerns arise from its controversial nature,
- alternatives should be explored to make an informed choice.

The application of Cohen's Kappa should be restricted to **testing if the observed agreement is significantly greater than what might occur by chance through random guessing**. When interpreting Cohen's Kappa values, general guidelines are often used to assess the level of agreement beyond chance [233].

Table 13: Interpretation of Cohen's Kappa coefficient

k	Interpretation
< 0	No agreement unless by chance
0.00 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 1.00	Almost perfect agreement

### 6.2.2. Pairwise F1 score

As a continuation of the **assessment of inter-annotator agreement**, the comparison of experts can be assessed using the pairwise F1 score between experts ( $F_{ij}$ ). This is calculated in the same way as described in **Section 6.1.1**, by pairing two experts at a time. To illustrate this calculation, let's take the example of the first matrix with the two experts, A and B. Depending on the reference considered, A or B, we have the following two confusion matrices:

If A is the reference				If B is the reference			
		Expert B				Expert B	
		Valid	Invalid			Valid	Invalid
Expert A	Valid	TN	FP	Expert A	Valid	TN	FN
	Invalid	FN	TP		Invalid	FP	TP

Figure 6-5: Confusion matrices based on the reference expert

Recalling the formulas for calculating precision and recall, we observe an inversion between precision and recall depending on the reference considered.

$$Precision = \frac{TP}{TP + FP} ; Recall = \frac{TP}{TP + FN}$$

However, as the F1 score is an unweighted harmonic mean of the two metrics, it remains constant. So, for each pair of experts, an F1 score is calculated, providing a balanced harmonic measure between precision and recall, irrespective of the reference used.

Furthermore, comparing experts using the pairwise F1 score involves **identifying outliers** to ensure the reliability of the annotations. For each annotator, the average difference in F1 score (calculated as  $1 - F_{ij}$ ) between this annotator and all other experts is determined [190]. Then, annotators whose mean difference exceeds the overall mean plus one standard deviation are identified as outliers. The use of  $1 - F_{ij}$  instead of the F1 score directly is explained by the need

to accentuate differences. By inverting the F1 score, we give more weight to weak performances, highlighting situations where one annotator stands out significantly in terms of disagreement compared to the others. This statistical method offers an objective approach to identifying annotators whose performance deviates substantially from the consensus established by the expert group. If such cases arise, it would be wiser to exclude such experts in the ground truth assessment.

### 6.2.3. Smyth's coefficient

Smyth's coefficient is a global metric used to assess agreement between a set of annotations. To do so, we compute the lower bound on the mean classification error rate relative to the 'true' labels for binary classification and  $N$  annotators according to [Equation 18](#) [234]:

**Equation 18: Error lower bound**

$$\bar{e} \geq \frac{1}{X \times N} \times \sum_{i=0}^X \min(N - A(i), A(i))$$

Where:

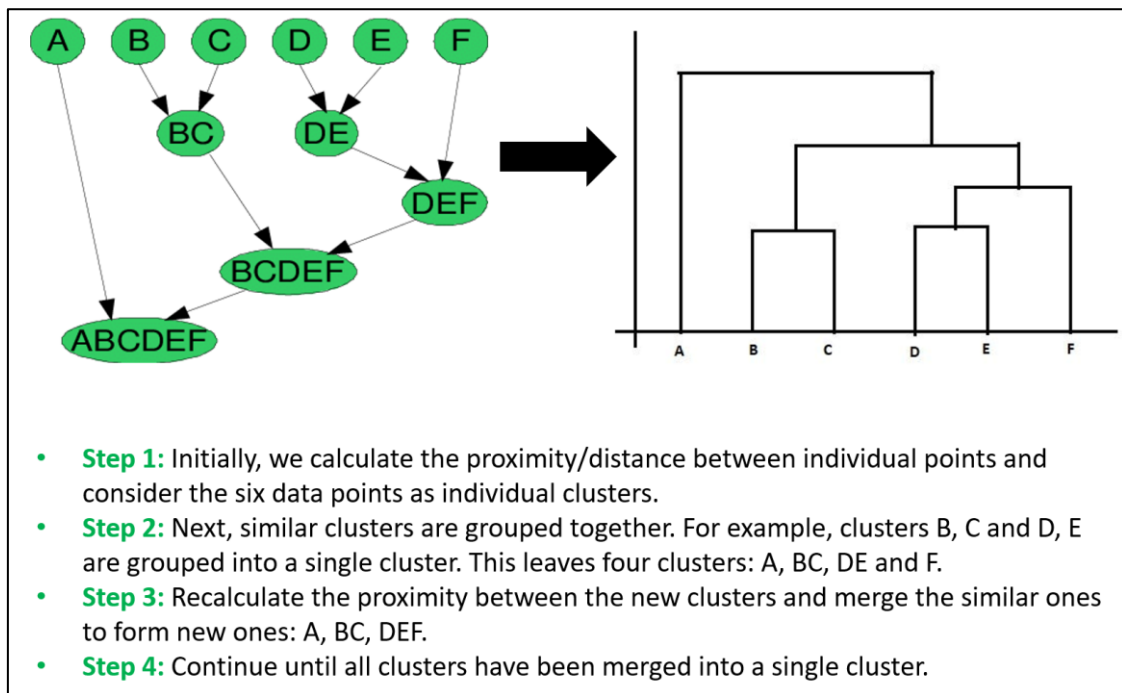
- $N$  is the total number of annotators
- $X$  is the number of samples
- $A(i)$  is the number of annotators that invalidated the data  $i$ .

The result of this calculation provides an approximation of the lower limit of error in annotations with respect to the unknown ground truth. In other words, it gives an idea of the disparity or disagreement that can be expected in the annotations compared to the real data. The concept of the method is closely linked to the entropy of annotators' decisions [190]. Entropy is a measure of uncertainty or randomness, and in this context reflects the degree of disagreement between annotators. More specifically, the entropy of annotators' decisions is related to the calculated error rate, providing an indicator of **the reliability of annotations**. If entropy is high, this suggests greater uncertainty and, consequently, greater discordance between annotators. Thus, Smyth's method offers an approach to assessing and interpreting annotation quality, taking into account the variability and consistency of annotators' decisions.

If annotators disagree on all items, their mean lower bound will be 0.5, while it will be 0 if they agree on all items. In line with the work of [234], if the lower bound of  $e$  is greater than 10%, the validation is considered inaccurate, and the quality of the expert annotation process needs to be reassessed.

### 6.2.4. Dendrogramm

The dendrogram represents a hierarchical clustering scheme that organizes data, in our case experts, into a tree-like structure based on their similarities [235]. The hierarchical clustering process begins by considering each object as its own group, then progressively combines similar groups to form larger ones, thus forming a hierarchy of clusters (see [Figure 6-6](#)). The dendrogram is a graphical representation of this hierarchy.



**Figure 6-6: Illustration of the principle of hierarchical clustering and the establishment of the corresponding dendrogram**

The fundamental aim of clustering the experts is to analyze the similarities between them, seeking to identify possible clusters linked to levels of expertise. In fact, the training of the experts took place in a cascading fashion, with expert A training expert C, who in turn trained experts B and D. This hierarchy gave rise to different levels of expertise, with A at senior level, C at confirmed level, and B and D at junior level. The central objective is to establish whether there is a distinction linked to level of expertise within the clusters formed. In addition, given the cascading nature of the training process, it is crucial to ensure that the trainer does not transmit his or her inherent bias and subjectivity to the learners, thus avoiding the creation of "clones". The aim is to have experts with their own subjectivity, while developing independent reasoning based on in-depth expertise and thorough analysis. This process ensures that data evaluation is based on critical thinking that goes beyond the obvious. In this way, dendrograms provide a graphic illustration of clusters, helping to answer these various questions and ensure diversity of perspective within the team of experts.

In the present study, hierarchical clustering is performed based on pairwise F1-score and using Ward's minimum variance, which is a hierarchical clustering technique that seeks to minimize intra-cluster variance when forming groups. Ward's approach considers each object as an individual cluster at the outset, and iteratively merges the most similar clusters to form larger groups. The decision to merge two clusters is guided by the criterion of minimizing intra-cluster variance, i.e., the merge is performed in such a way as to minimize the increase in variance within the resulting cluster. This method aims to create homogeneous clusters in terms of similarity, while preserving the internal consistency of each group.

### 6.3. Behind the Scenes of AI Models: Hardware & Software

Laying the groundwork for an effective development environment for AI models is a process that demands a thoughtful integration of both hardware and software elements to ensure optimal performance.

#### 6.3.1. Programming language

The choice of programming language is of crucial importance in the development of AI models, and in our case, Python was the predominant choice. There are several reasons for this preference. Firstly, **Python** offers a clear and concise syntax, making code easier to read and speeding up the development process. Its vast community of developers has contributed to the creation of a plethora of libraries specialized in machine learning.

#### 6.3.2. Environment set-up

As far as software is concerned, the choice of an integrated development environment (IDE) is of crucial importance in the creation, debugging and testing of AI models. With this in mind, we opted to use the **PyCharm IDE**, a Python distribution that facilitates package management and offers a user-friendly interface for the development of AI-based applications.

Furthermore, the configuration of this development environment implies the use of dependency management tools such as **Anaconda**, thus ensuring efficient reproducibility of the environment. This choice guarantees consistency between the different stages of development, from initial experimentation to production.

When it comes to developing AI models, the software arsenal encompasses a plethora of specialized libraries. Essential tools such as **Pandas** and **NumPy** are frequently used for data manipulation and pre-processing prior to model training. For graphical visualization, **Matplotlib** is integrated to create clear and informative visual representations.

For ML model development, **scikit-learn** emerges as an essential library, offering a variety of pre-implemented models and performance evaluation tools. In parallel, **TensorFlow** and **Keras** are proving to be wise choices for their power and flexibility, offering advanced features for the design, training and deployment of deep learning models.

### 6.3.3. Hardware

As far as hardware is concerned, selection is closely linked to the complexity of the model and the volume of data to be processed. Central processing units (CPUs) are commonly favored for their ability to run parallel calculations, offering an efficient solution for less processing power-intensive tasks. Meanwhile, graphics processing units (GPUs) are preferred for their ability to significantly accelerate deep learning operations thanks to their optimized parallel architecture. It should be noted that the specific choice of hardware used in our configuration is detailed in the **Table 14**, taking into account parameters such as computing power, GPU memory, and other relevant characteristics to ensure optimal performance according to the specific requirements of our AI models.

**Table 14: Hardware specification**

<b>CPU model</b>	AMD EPYC 7502P 32-Core Processor
<b>RAM</b>	16.4 GB
<b>GPU</b>	NVIDIA GeForce RTX 3090, 24 GB

In short, setting up an AI development environment requires a balanced combination of high-performance hardware, a suitable environment and specialized software, while taking into account the specific needs of the model.

### 6.4. Synthesis of Chapter 6

This chapter explores beyond data analysis and model training, focusing on the critical dimensions of performance metrics (see [Table 15](#)) and hardware-software configuration. The evaluation of AI models dedicated to anomaly detection requires a nuanced exploration of the metrics quantifying their success. Model performance metrics are explored, highlighting the crucial role of F1 score, MCC and AUC in analyzing anomaly detection models. The confusion matrix, although not a metric, is essential for binary classification.

On the other hand, annotator agreement metrics are crucial for assessing validation variance within the validation pool. Global metrics, such as Smyth's coefficient, and pairwise metrics, such as Cohen's Kappa coefficient and  $F_{ij}$  score, are used to assess the reliability of annotators in the validation pool.

Finally, the section on hardware and software highlights the behind-the-scenes aspects of the choice of programming language (Python), integrated development environment (PyCharm), and the use of open-access libraries such as Pandas, NumPy, scikit-learn, TensorFlow and Keras. The hardware selected includes 32 computing cores and a robust graphics card.



**Table 15: Model performance and annotator agreement metrics**

	<b>Metrics</b>	<b>Interpretation</b>	<b>Equation</b>
<b>Model Performance</b>	<i>Confusion Matrix</i>	Visually presents model predictions against ground truth labels	Table 12
	<i>Accuracy</i>	Represents the proportion of correct predictions.	Equation 9
	<i>Precision</i>	Evaluates false alarms	Equation 10
	<i>Recall</i>	Assesses missed anomalies	Equation 11
	<i>F1 score</i>	A harmonic mean of precision and recall	Equation 12
	<i>MCC</i>	Offers a balanced classification metric, considering both positive and negative instances.	Equation 13
	<i>ROC curve</i>	Visualize classifier performance at different thresholds.	Figure 6-3
	<i>PR curve</i>		Figure 6-4
	<i>AUC</i>	Quantifies the overall discriminatory ability of the model across all threshold values.	
<b>Annotator agreement</b>	<i>Cohen's Kappa</i>	Address chance agreement between annotators.	Equation 17
	<i>Smyth's coefficient</i>	Estimates the lower bound of error in annotations, considering entropy for reliability.	Equation 18
	<i>Pairwise F1 score</i>	Identifies outliers by assessing the average difference between an annotator and others.	
	<i>Dendrogram</i>	Utilizes hierarchical clustering based on pairwise F1 scores to visually represent annotator similarities.	Figure 6-6

## ***Synthesis of Part II***

This section focuses on the challenges and methodological approaches involved in validating data in the field of urban wastewater, using turbidity data from the Saint-Malo Agglomeration. The instrumentation comprises six intercept sites with turbidity and conductivity sensors. Attention is paid to the validation of turbidity measurements, with major efforts to ensure data reliability. A data pre-processing phase compensates for frequency anomalies. Statistical analysis reveals non-stationarity in time series. An expert methodology dedicated to turbidity validation is developed, comprising three manual / semi-automated steps. A quantitative assessment is undertaken to understand the impact of human subjectivity on the evaluation of artificial intelligence models. A validation pool with four experts is established to compare the performance of annotators. Annotator agreement metrics, such as Cohen's Kappa coefficient and F1 score, are essential for assessing validation variance.

The benchmark chapter introduces three models (Matrix Profile, ResNet and Autoencoders) for data validation and anomaly detection. Experiments use turbidity data from the Cottage site, assessing sensitivity to input data and testing semi-supervised approaches. Hyperparameter tuning, result visualization and enhancements such as ensemble models are explored. Evaluation includes tests on the full database, multivariate approach taking conductivity into account, and generalization to other sites with learning transfer tests.

In addition, the chapter explores critical dimensions of model performance and hardware-software configuration. Model performance metrics such as F1 score, Matthew's correlation coefficient (MCC) and area under the curve (AUC) are examined for the analysis of anomaly detection models. Finally, hardware-software aspects, including the choice of Python as programming language, PyCharm as integrated development environment are presented.

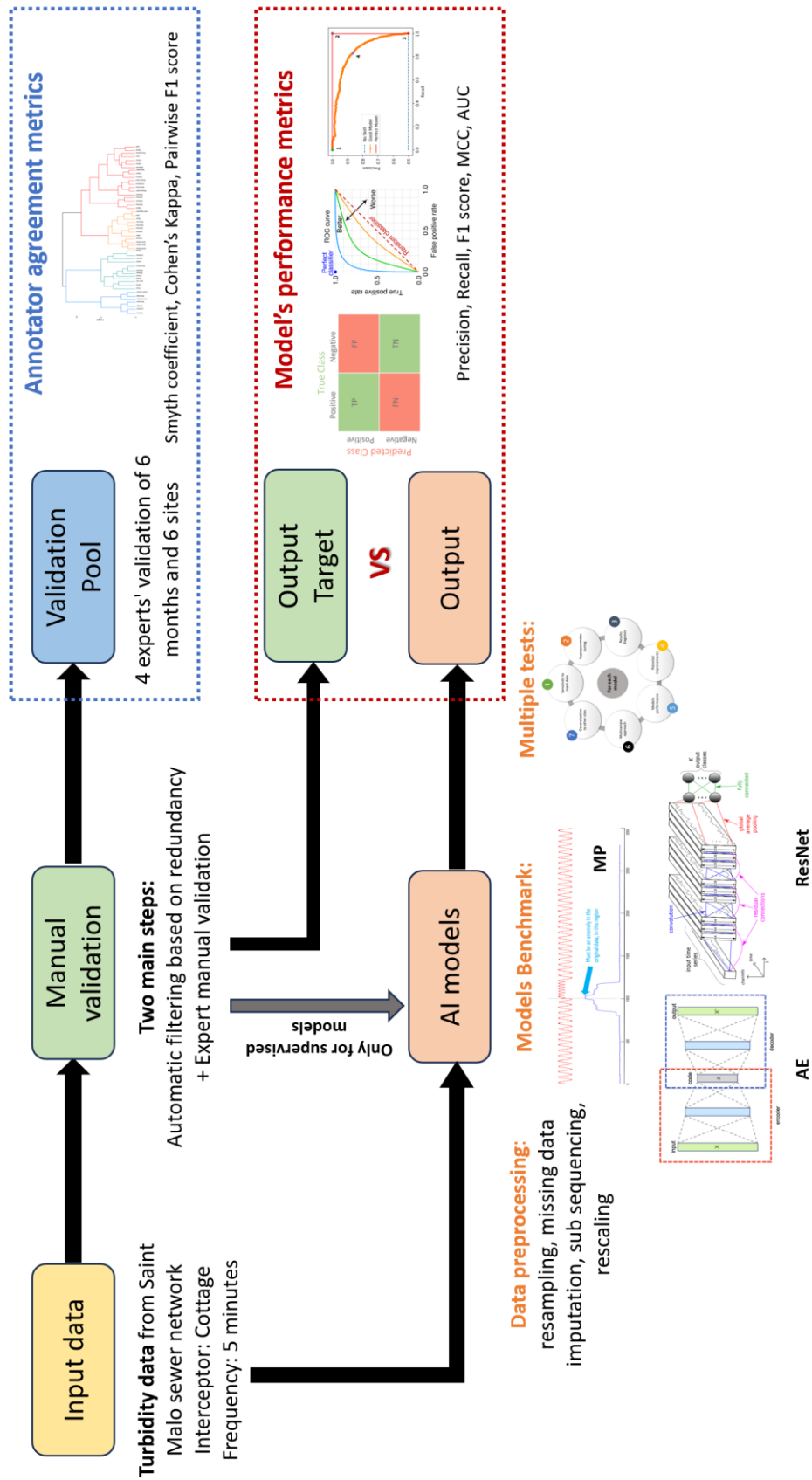



Figure 6-7: Overview of section II: Materials and methods



# **Part III**

## **Results and Discussion**

## ***Introduction of Part III***

In the upcoming chapters, a set of overarching questions will guide our exploration into the evaluation and improvement of anomaly detection model performance in the context of data validation.

- First, how can we evaluate **annotator agreement in our database** ?
- Then, what strategies can be employed to **evaluate and enhance the performance of our benchmark models**, namely Matrix Profile, ResNet and Autoencoder, taking into account factors such as sensitivity to input data, hyperparameter tuning and generalization ?
- Finally, how can we assess **the relationship between annotator agreement and model performance** ? And to what extent are the models tested to date on turbidity data suitable for **other data from wastewater networks**, such as conductivity and water level?

## Chapter 7. Annotator Agreement

---

It is widely acknowledged that annotators of an event of interest rarely meet in perfect agreement when expressing their opinions. To ensure effective learning in the case of a supervised approach, and to rigorously evaluate the performance of an algorithm in the case of a supervised and/or an unsupervised approach, the existence of a baseline, often called a "ground truth", is essential. However, in many cases, acquiring a reference test can be costly, if not impossible. In our case, there is no gold truth available (i.e. a true value of turbidity being measured), and we used a combination of redundancy and assessment by an expert as a reference.

In order to assess the reliability of this reference and to verify the frequently adopted assumption that the opinion of one (or a few) annotator(s) approximates the reference truth, an experiment has been set up consisting in the constitution of a validation pool involving various experts to assess their agreement and variability. This validation pool is based on the opinions of four experts and on a sufficiently large database to reconcile the representativeness of annual variations and time constraints. Validation of a one-month chronicle takes an average of two hours. Taking into account the training time of the experts, who nonetheless have a solid background in the understanding of wastewater network operation, two trainees were hired for 3 months. The test conditions and hypotheses were detailed in [Section 4.4.3](#). It should be noted that in this analysis, we compare the final validation result obtained through the combination of the filtering, expertise and aggregation phases. Although the filtering stage automatically validates many redundant sequences, which make up a large proportion of the valid data (see [Figure 4-8](#)), the expert may have to adjust the delimitation of certain non-redundant periods according to his expertise. Consequently, it was decided to compare the experts on the final result of the process rather than just on the expertise phase. This approach avoids the need for additional work to isolate expert intervention periods and take into account individual changes to their delimitation.

The aim of this study is therefore to assess the noise<sup>11</sup>, if any, to be taken into account when evaluating AI models, depending on the target output considered for evaluation. This evaluation will be based on various aspects and will be carried out using several metrics, including the Pairwise F1 Score, which will determine the presence of atypical experts (see [Section 7.1](#)), the dendrogram, which will assess the validity of the test conditions as well as

---

<sup>11</sup> Noise being a flaw in Human Judgment as stated by [236]

the absence of a training bias (see [Section 7.2](#)), Cohen's Kappa, used to determine whether agreement between experts results from pure chance, and finally (see [Section 7.3](#)), Smyth's coefficient, which will assess whether the variability observed is suitable for evaluating a model (see [Section 7.4](#)).

### 7.1. Identifying “Outlier” Experts with Pairwise F1 Score

The comparison between experts begins with the creation of confusion matrices using a pairwise approach, first on a monthly basis and per site, then globally (see [Appendix J](#)). In some cases, the experts do not reject any time step, meaning that the month is fully validated. In these cases, the F1 score is not defined. By convention, it is replaced by 1, as in practice this indicates complete agreement between the experts. [Figure 7-1](#) illustrates pairwise matrices, where for 4 experts, we have 6 matrices.

		Expert B				Expert A				Expert A	
		Valid	Invalid			Valid	Invalid			Valid	Invalid
Expert A	Valid	173282	7920	Expert C	Valid	169266	4269	Expert D	Valid	177648	4848
	Invalid	4005	26088		Invalid	11936	25824		Invalid	3554	25245
		Expert B				Expert B				Expert C	
		Valid	Invalid			Valid	Invalid			Valid	Invalid
Expert C	Valid	169550	3985	Expert D	Valid	174032	8464	Expert D	Valid	170907	11589
	Invalid	7737	30023		Invalid	3255	25544		Invalid	2628	26171

**Figure 7-1: Global pairwise confusion matrices issued from the validation pool**

The analysis of confusion matrices reveals that despite the use of a common database, the rates of invalid data identified by various experts show differences (see [Figure 7-2](#)). Indeed, a maximum standard deviation of around 3.3% is observed with regard to the average anomaly rate per site, indicating an overall consistency in the experts' ability to detect and classify invalid data. When we look specifically at our typical site, "Cottage", the standard deviation is even smaller, at just over 1%. This positive finding underlines the accuracy and consistency of the evaluations established by our validation pool.

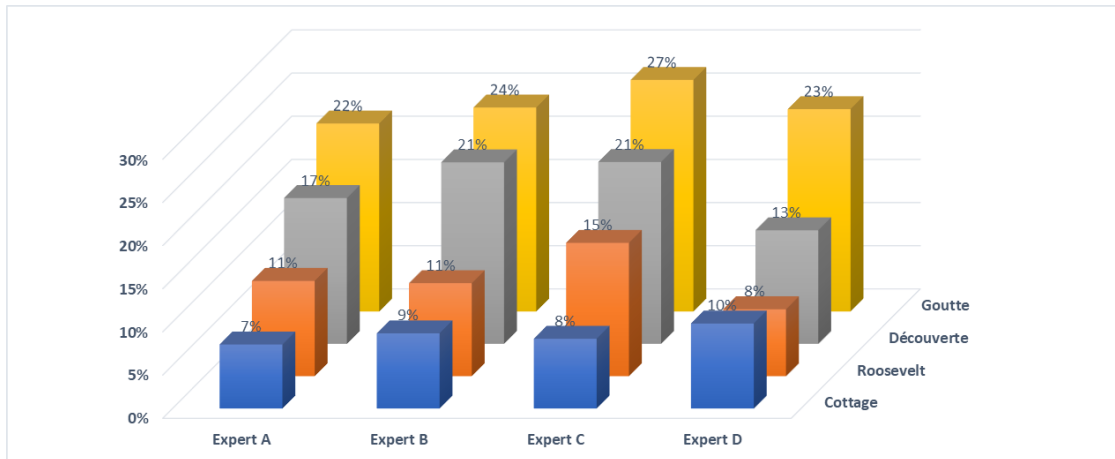


Figure 7-2: Anomaly rate by site according to each expert on the basis of data used by the validation pool

The second step in the process is to calculate the pairwise F1 score from the pairwise matrices (see Figure 7-3).

	Expert A	Expert B	Expert C	Expert D
Expert A	1			
Expert B	0.8140	1		
Expert C	0.7612	0.8367	1	
Expert D	0.8573	0.8134	0.7864	1

Figure 7-3: Pairwise F1 scores among the validation pool

The provided pairwise F1 score table reveals the agreement and discrepancies in anomaly detection assessments among the four experts: A, B, C, and D. For instance, the F1 score between Expert A and Expert D is the highest (F1 score  $A/D = 0.8573$ ), suggesting a substantial level of agreement, though not perfect. Similarly, the F1 scores between different experts display varying degrees of concordance, but which remain above 0.75. **The overall average F1 score is of 0.81.**

To assess the relevance of the results obtained, it is necessary to compare the F1 score values with reference thresholds. An F1 score close to 1 indicates a high agreement between experts' assessments. However, there is no minimum threshold at which an F1 score can be considered unsatisfactory in absolute terms. The definition of a minimum F1 score threshold will depend on the context and the issue. In our case, the latter is set to 0.5 since an F1 score below this limit suggests an imbalance between precision and recall. Therefore, we consider that the results obtained are rather satisfying.



Once the different expert assessments have been compared, it becomes useful to identify any "outlier" experts who stand out from the group. For example, Expert C's F1 score is significantly lower than that of the other experts (see [Figure 7-3](#)). The question is then whether Expert C can be qualified as an outlier. In the context of anomaly detection, an outlier among experts is defined as an annotator whose annotations deviate significantly from the general trend of the group.

In this context, [Section 6.2.2](#) proposed a methodical approach for evaluating experts using the average difference in F1 score between each annotator and all other experts. Annotators whose mean difference exceeds the overall mean plus one standard deviation (critical threshold) are identified as outliers.

[Table 16](#) summarizes the average F1 score differences for each expert, providing an overview of the discrepancies between the assessments of the different annotators. The critical threshold is set at 0.22 to identify possible expert outliers. Analysis of the results shows that, although expert C is close to this threshold, no expert exceeds it. In the case of looking for outliers in a small sample of data, the Student test offers a more robust approach than the simple rule of the average plus a standard deviation. It is a statistical tool that can be used to assess whether a value is an outlier in a given sample taking into account its reduced size. Considering a confidence interval of 68% in analogy with the rule of mean plus a standard deviation and a 95% confidence interval that is wider, we find that the upper/lower limits of the confidence interval are of (0.207/ 0.170) and (0.238/ 0.138) respectively. This evaluation allows us to conclude that, overall, **no expert can be qualified as an outlier in our validation process**. It should be noted that this assessment was also carried out at month and site level, and no outlier is identified in any case. This finding reinforces the consistency of the assessments provided by the group of experts.

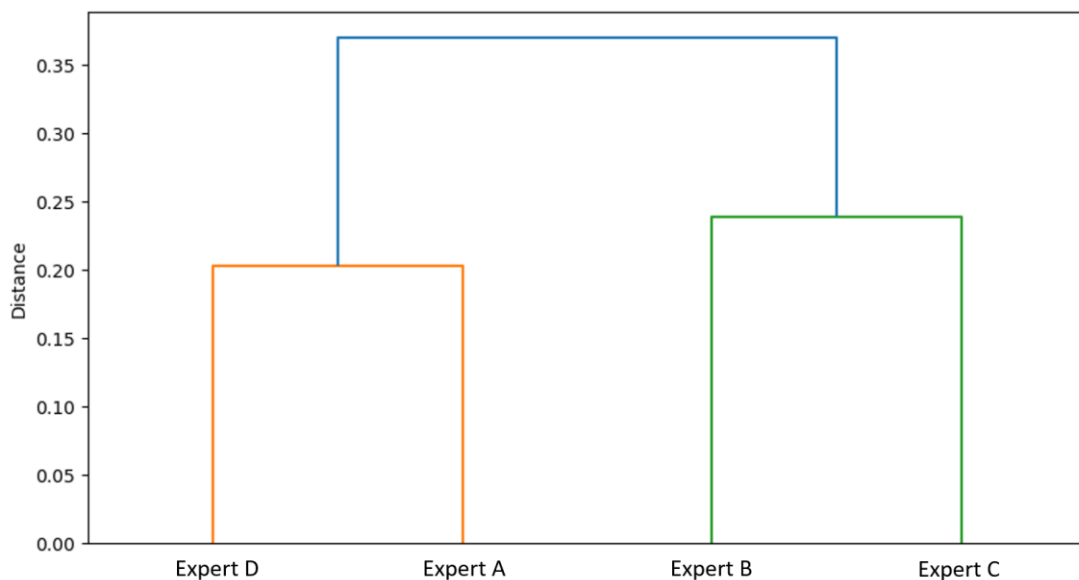
**Table 16 : Mean F1 score differences for each expert**

	Mean F1 score differences
<b>Expert A</b>	0.1892
<b>Expert B</b>	0.1787
<b>Expert C</b>	0.2053
<b>Expert D</b>	0.1810

## 7.2. Clustering Experts using Dendrogram

The fundamental aim of clustering the experts is to analyze the similarities between them, seeking to identify eventual clusters linked to levels of expertise. In addition, given the cascading nature of the training process, it is crucial to ensure that the trainer does not transmit his or her inherent bias and subjectivity to the learners, thus avoiding the creation of "clones" (see [Section 4.4.3](#)).

In this way, dendrograms provide a graphic illustration of clusters, helping to answer these various questions and ensure diversity of perspective within the team of experts.

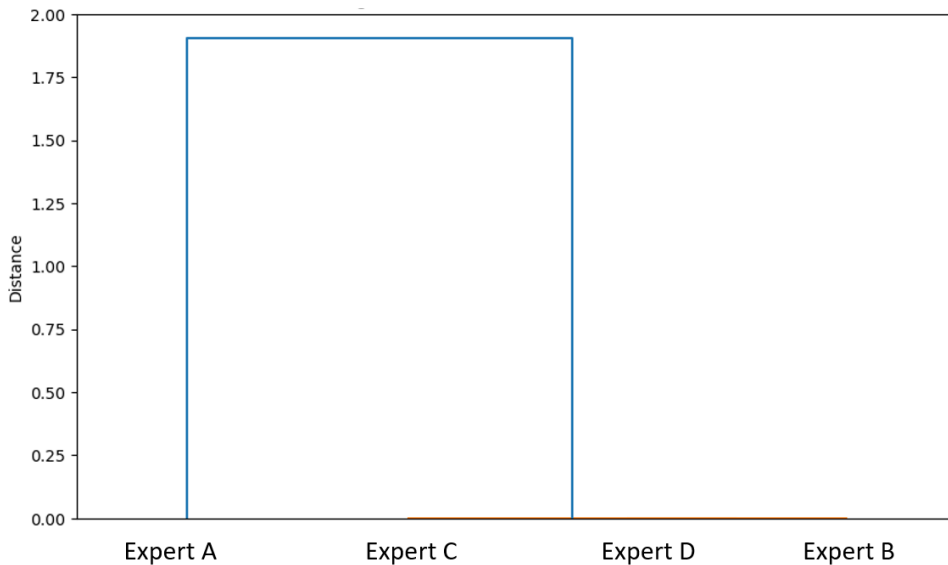


**Figure 7-4: Global Dendrogram**

Examination of the dendrogram (see [Figure 7-4](#)) reveals the presence of two distinct clusters among the assessed experts. Experts A and D cluster closely together, demonstrating a marked similarity, despite the fact that they have never been involved in mutual assessments. This proximity suggests a convergence of expertise and indicates that these two experts represent two distinct levels of expertise within the same group. On the other hand, experts B and C form another cluster with a distance that is not zero. This observation rules out any possibility that they are duplicates or clones, thus affirming that although experts share certain similarities, they retain individuality in their assessments. Cluster analysis within the dendrogram thus offers relevant insights into the dynamics of relationships between experts, highlighting both convergences and differences within the group of experts.

By investigating site- and month-specific dendrograms, new architectures emerge, highlighting the absence of absolute similarities between sets of experts. [Figure 7-5](#) illustrates one

particular dendrogram, where experts B, C, and D performed similar validations, while expert A stands out.



**Figure 7-5: Dendrogram of the validation of Cottage - March**

As a whole, March data at Cottage show overall validity, as attested by the consensus of the experts, with the exception of a sequence lasting about an hour on March 25<sup>th</sup>, 2022, around 9:30 a.m. that was validated by expert A, while invalidated by experts B, C, and D. Notably, during this sequence (see [Figure 7-6](#)), a notable increase in turbidity is followed by a sudden drop to zero, characteristics that clearly correspond to an abnormal pattern. This scenario highlights the potential human error and underscores the complexity inherent in identifying short fault periods within large databases. The task becomes particularly daunting when we consider that we had to detect this single anomalous hour among the 744 hours of March. In reality, Expert A may have missed this anomaly by mistake, or on the contrary may have deliberately decided to validate it because of its short duration, reflecting his subjective judgment. It should be noted that there is no precise duration limit beyond which a defect must be invalidated, which leaves room for a degree of subjectivity in the assessment.

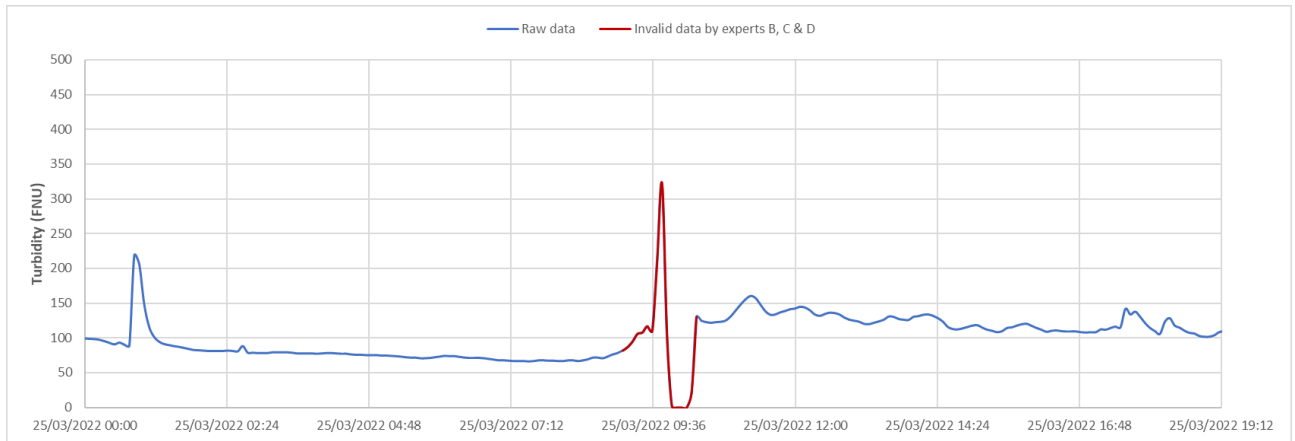


Figure 7-6: Anomaly identified by experts B, C & D

Furthermore, the dendrogram presented in Figure 7-7 reveals a cascading structure, where experts B and D share a proximity, followed by a similarity with expert A, then finally with expert C.

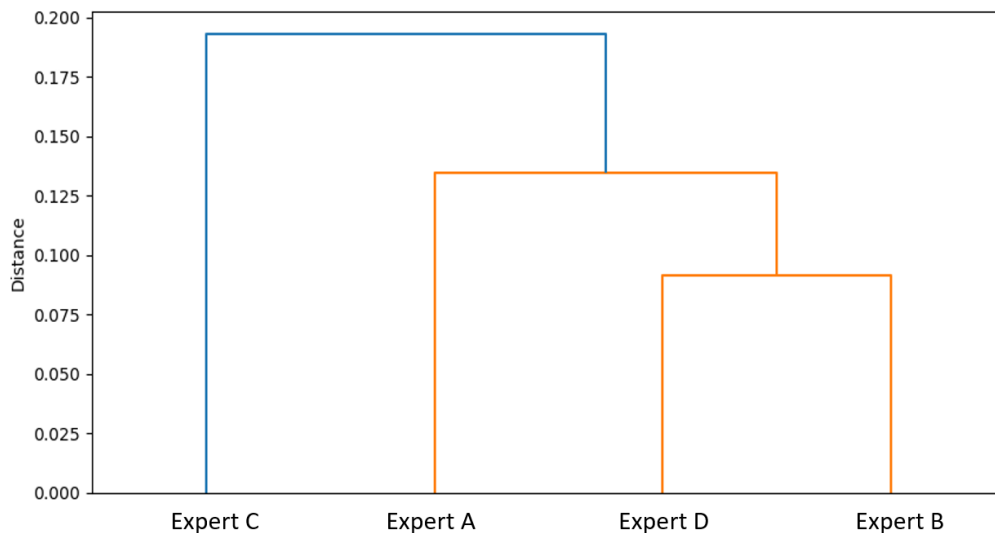


Figure 7-7: Dendrogram of the validation of Cottage - July

These observations underline the variable relationships between experts showing an **unbiased relationship between different annotators**. This variability in validation reinforces the idea that each expert brings a unique perspective to data evaluation, and that no rigid similarity prevails between different sets of experts.

### 7.3. Assessing Beyond Chance Agreement with Cohen's Kappa

The calculation of Cohen's Kappa coefficient shares similar numerical limitations to those of the F1 score. In particular, when both experts validate the entire chronicle, the Kappa is undefined. However, the Kappa presents an additional challenge, as this same configuration occurs even when both experts invalidate the entire chronicle. In this scenario, the F1 score is

equal to 1, while the Kappa remains undefined. In these cases of perfect agreement, where the two experts are in complete harmony, the Kappa is conventionally equal to 1. Another situation arises when one of the experts does not choose a specific class, validating or invalidating the whole chronicle, while the other expert provides a more nuanced result, if only for one time step. In this case, the Kappa is equal to 0, even if the disagreement concerns only a single time step. **Figure 7-8** summarizes the Kappa pairwise results between the different experts.

	Expert A	Expert B	Expert C	Expert D
Expert A	1			
Expert B	0.7808			
Expert C	0.7162	0.8034	1	
Expert D	0.8342	0.7811	0.7473	1

**Figure 7-8: Global Pairwise Cohen's Kappa**

If we consider the interpretation scale (see **Table 13**), we come to the conclusion that the results are highly satisfactory, testifying to significant agreement that is not purely by chance. More specifically, the comparisons Expert A vs Expert D, Expert B vs Expert C show high levels of consistency, demonstrating significant agreement in their respective evaluations, which is in line with the results of hierarchical clustering and those of the pairwise F1 score. Although some pairs show slightly lower consistency, such as Expert A vs. Expert C, these values remain within a range considered substantial. Overall, Cohen's Kappa results indicate **a level of agreement that transcends mere chance agreement**, reinforcing the credibility of the assessments provided by the team of experts in this study.

Cohen's Kappa coefficient, while a valuable tool for measuring inter-rater agreement, has inherent limitations that need to be considered, mainly the interpretation of its value. Let's take the example of the validation of Roosevelt's data in November, where pairwise confusion matrices are illustrated in **Figure 7-9**.

		Expert B				Expert A				Expert A	
		Valid	Invalid			Valid	Invalid			Valid	Invalid
Expert A	Valid	8300	0	Expert C	Valid	8156	265	Expert D	Valid	8280	278
	Invalid	269	71		Invalid	144	75		Invalid	20	62
		Expert B				Expert B				Expert C	
		Valid	Invalid			Valid	Invalid			Valid	Invalid
Expert C	Valid	8416	5	Expert D	Valid	8549	9	Expert D	Valid	8421	137
	Invalid	153	66		Invalid	20	62		Invalid	0	82

Figure 7-9: Pairwise confusion matrices for Roosevelt - November

When we calculate Cohen's Kappa coefficients in this scenario, values range from 0.81 to 0.25, illustrating considerable diversity in levels of agreement between expert peers, ranging from fair agreement to substantial agreement (see Figure 7-10).

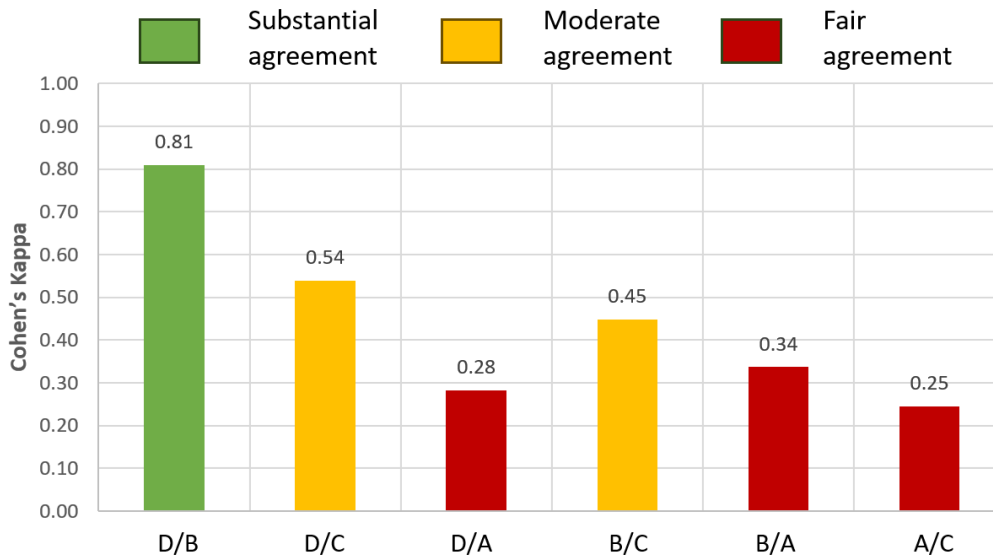


Figure 7-10: Pairwise Cohen's Kappa for Roosevelt – November

When we analyze the validation results of Roosevelt - November, a period of significant anomaly clearly emerges around November 16 (see Figure 7-11). Although each expert identifies this period, discrepancies remain in the precise delimitation of the anomaly. Moreover, when comparing the validations carried out by expert A and expert C, both experts concur in identifying this particular defect despite some variations in the delimitation. Nevertheless, expert C identifies additional anomalies. This convergence in anomaly detection calls into question the idea that agreement between A and C could be the result of chance. On the contrary, this observation suggests intentional agreement in the recognition of defects, reinforcing the validity of their joint assessment. These results underline the importance of a nuanced assessment, taking into account not only differences in anomaly delineation but also overall consistency in defect identification between the experts.



Figure 7-11: Expert validation results of November chronicle - Roosevelt

Moreover, one of the main limitations concerns the sensitivity of Kappa to class distribution, particularly in the presence of a significant imbalance between the categories evaluated and/or when classes have a small number of occurrences. In such situations, a small fluctuation in the number of agreements or disagreements can lead to large variations in the coefficient, which can make interpretation of the results tricky. Taking the example of Roosevelt dataset in May highlights the extreme cases that can arise (see [Figure 7-12](#)).

		Expert B				Expert A				Expert A	
		Valid	Invalid			Valid	Invalid			Valid	Invalid
Expert A	Valid	8913	14	Expert C	Valid	7032	0	Expert D	Valid	8922	0
	Invalid	0	1		Invalid	1895	1		Invalid	5	1
		Expert B				Expert B				Expert C	
		Valid	Invalid			Valid	Invalid			Valid	Invalid
Expert C	Valid	7018	14	Expert D	Valid	8913	9	Expert D	Valid	7027	1895
	Invalid	1895	1		Invalid	0	6		Invalid	5	1

Figure 7-12: Pairwise confusion matrices for Roosevelt – May

On one hand, Expert C invalidates significantly more data points than the other experts, resulting in negative Cohen's Kappa values, indicating a complete disagreement with the rest of the experts. On the other hand, Expert A invalidates only a single data point. In this scenario, the small sample size contributes to relatively low Cohen's Kappa values associated with this expert. Even a marginal increase in a few data points can cause an immediate shift to a moderate agreement (see [Figure 7-13](#)). Thus, this scenario with a reduced number of samples demonstrates the instability that can occur in Cohen's Kappa calculations, emphasizing the need for cautious interpretation in situations where sample sizes are constrained.

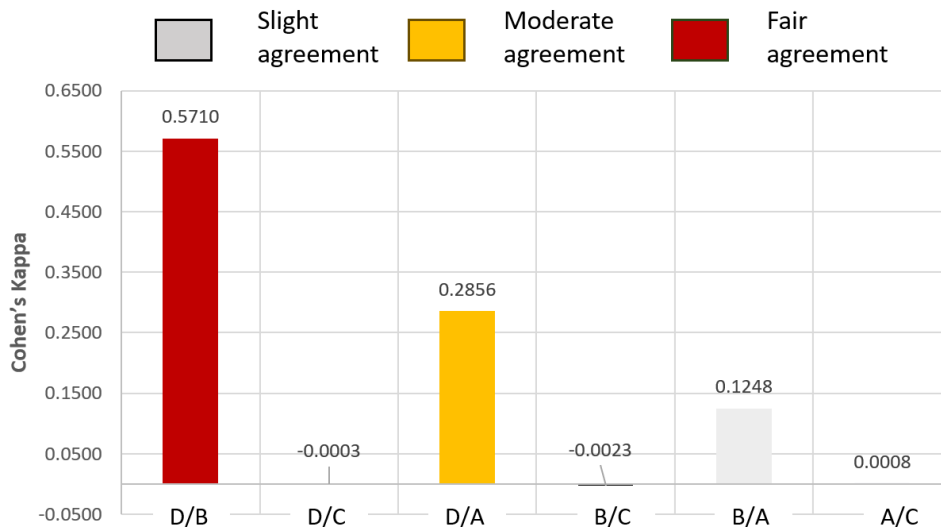


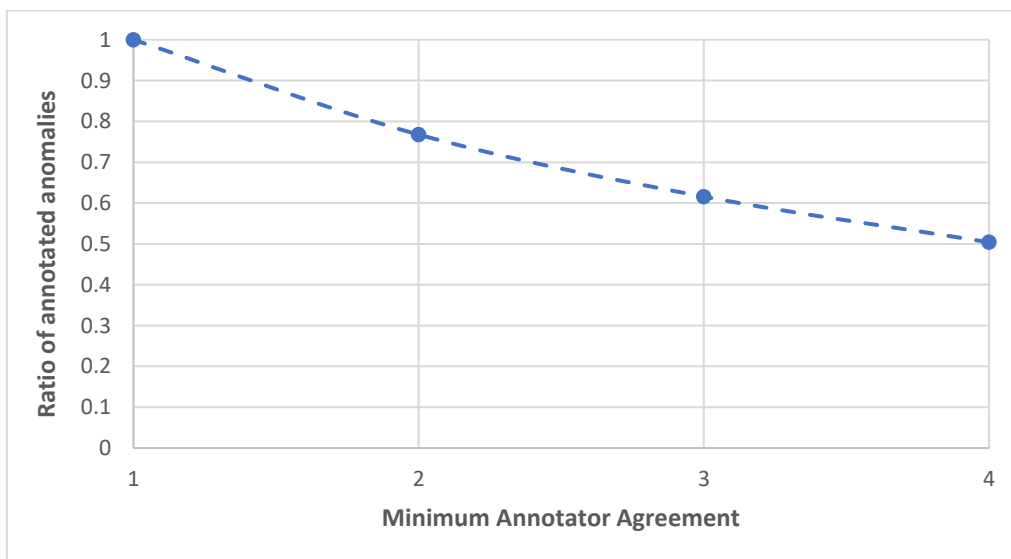
Figure 7-13: Pairwise Cohen's Kappa for Roosevelt – May



In summary, it is essential to consider the limitations of Cohen's Kappa coefficient, particularly in contexts where class imbalance is pronounced. In our particular study, and for site- or month-specific analyses, class imbalance may be a concern, and this impacts the interpretation of Cohen's Kappa coefficient. However, despite these considerations, we maintain that the overall result remains accurate, as the presence of a sufficient number of faults, albeit less frequent than valid data, allows us to ensure a robust and meaningful assessment.

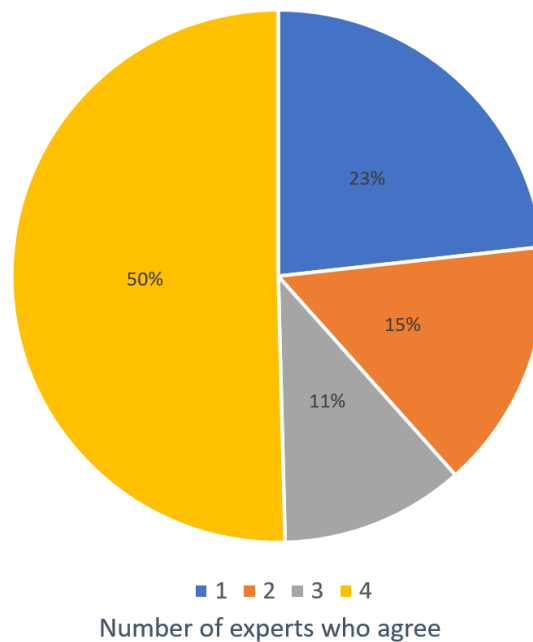
#### 7.4. Smyth's Coefficient Analysis for Evaluating global annotator agreement

Unlike metrics that focus on pairwise comparisons, the Smyth coefficient provides an overall assessment of the agreement between different experts. The lower error bound is essential derived from a parameter  $A(i)$ , representing the number of experts who marked a timestamp  $i$  as abnormal. Firstly, we study the overall evolution of this parameter by examining the ratio of abnormal data identified by at least 1, 2, 3 experts, and finally those identified unanimously (see [Figure 7-14](#)). We observe a quasi-linear trend that results in a gradual reduction in the ratio of invalid data as agreement between experts increases. More precisely, between level 1 (an expert identifying an anomaly) and level 2 (two experts agreeing on an anomaly), we note a loss of about 23% of anomalies. This means that the ratio of anomalies that were not consolidated by another expert opinion remains limited. This observation contrasts with the literature, where an exponential decrease in the agreement rate is often observed between the different levels, and where the maximum agreement rate is generally limited to a few percents [190].



**Figure 7-14: Ratio of timesteps having a minimum level of annotator agreement**

Quantitatively, the results reveal that among the anomalies detected, an absolute agreement is reached for 50% of them, while for 23% + 11%, a consensus can be established (see [Figure 7-15](#)). This category includes cases where a single expert marks the timestamp as abnormal, as well as cases where three experts share this assessment, suggesting that the opinion of the one who deviates from the group may be considered non-significant. However, in 15% of cases, two experts consider an anomaly, while the other two validate it. Thus, our observation suggests that the agreement between the experts in our study is relatively high, demonstrating a robustness in the detection of anomalies that goes beyond the trends observed in other contexts.



**Figure 7-15: Ratio of invalid timestamps following the level of agreement**

In order to ensure that these results are acceptable for experimental use, an estimate of the Smyth lower error limit, that is, the average error rate among annotators, was calculated and turned out to be 3.5%. This value is well within the recommended limit of 10% [234] and is of the same order of magnitude as those observed in the specialized literature [190]. This consistency with the recommended standards and trends observed in other studies reinforces the validity of the results obtained in our inter-expert assessment. **The relatively low error rate indicates an appreciable reliability and consistency among the annotators**, thus strengthening the credibility of the evaluations obtained as part of our experiment.

Table 17: Smyth's lower error bound for site and month- specific scenarios

	Cottage	Goutte	Découverte	Roosevelt	Average
July	0.018	0.039	0.045	0.001	0.026
September	0.039	0.024	0.006	0.067	0.034
November	0.001	0.002	0.074	0.013	0.023
January	0.023	0.022	0.084	0.019	0.037
March	0.001	0.018	0.162	0.057	0.059
May	0.018	0.051	0.001	0.054	0.031
Average	0.017	0.026	0.062	0.035	0.035

However, the analysis of the Smyth coefficient specific to the different site and month scenarios reveals a “worrying” situation for the case of “Découverte” in March, where the coefficient exceeds the acceptable threshold (see Table 17). A thorough investigation of the experts’ validation results reveals that the main differences between the experts are related to the precise delineation of the defects (see Figure 7-16) . Overall, three fault periods stand out around March 14, 21 and 26. However, there are differences among experts regarding the delimitation of these periods. For example, Expert B groups the second and third default periods. On the other hand, other experts delimit periods of slightly larger anomalies, framing the abnormal sequence, as does expert C, while others limit themselves to snippets such as expert D. These discrepancies suggest that, although experts generally agree on the presence of anomalies, significant differences remain in the exact delimitation of these defects. This variation in anomaly identification can have significant implications for the overall performance of an anomaly detection model. These observations also highlight the complexity inherent in evaluating anomaly detection models, where subtle nuances can have a significant impact on performance evaluation.

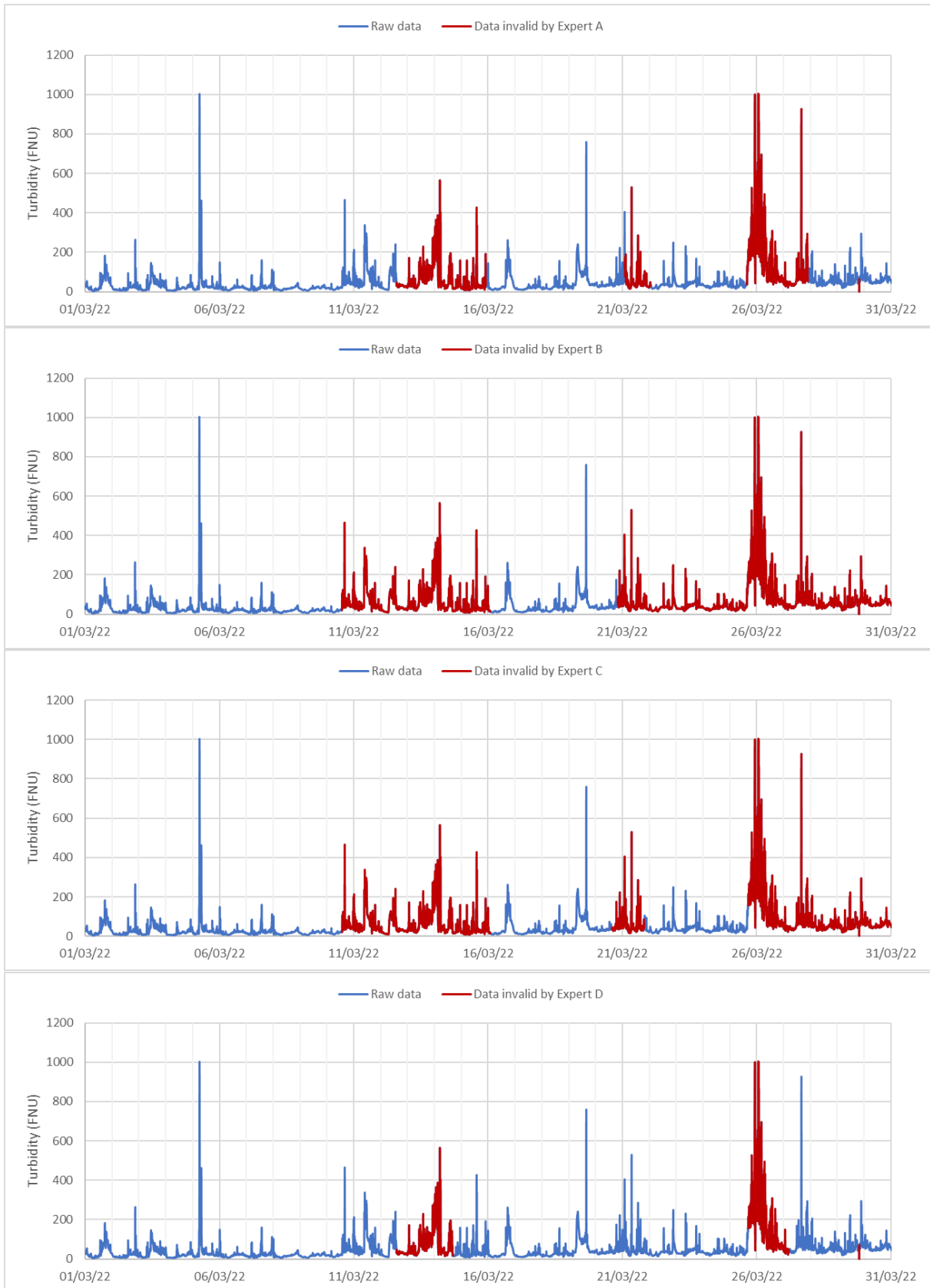


Figure 7-16: Expert validation results of March chronicle – Découverte

### 7.5. Synthesis of Chapter 7

This chapter stresses the challenges of acquiring a reference test, making the experiment with a validation pool necessary for assessing the reference truth in the evaluation process. The main goal is to evaluate the bias and variance introduced by manual validation in AI models' evaluation.

Pairwise F1 scores are calculated, revealing agreement and discrepancies among experts, with an overall average F1 score of 0.81. The results are considered satisfying, with no expert identified as deviating significantly from the group's consensus. The clustering of experts using dendrograms allows to analyze similarities and identify possible clusters between experts. The global dendrogram illustrates two distinct clusters among assessed experts, revealing mixity in expertise, with no "heritage" effect. Overall, the observations underscore the variable relationships between experts, emphasizing the unbiased nature of these relationships and the unique perspectives each expert brings to data evaluation. The global pairwise Cohen's Kappa results indicate highly satisfactory agreement, with Expert A vs. Expert D and Expert B vs. Expert C showing substantial consistency. These similarities are the same as exhibited by the clustering presented hereabove. Slightly lower consistency is observed in some pairs, but still fall within a considered substantial range. Cohen's Kappa results affirm a level of agreement beyond chance, reinforcing the credibility of expert assessments. Finally, the Smyth coefficient is employed to evaluate global annotator agreement, providing an overall assessment of agreement between different experts. The Smyth lower error limit is calculated at 3.5%, well within recommended limits. These low error rates reinforce the reliability and consistency of annotator evaluations, validating the results obtained in the experiment.

In conclusion, all the analyses converge towards a robust and credible assessment of manual validation process (filtering using redundancy + expertise + aggregation). Despite the challenges inherent in the diversity of interpretations and nuances, the results highlight considerable consistency and reliability in the assessments.

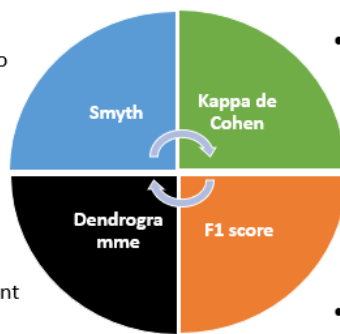
- The calculated lower error bound does not exceed 10%, indicating a satisfactory level of agreement between annotators and proximity to ground-truth

$$\bar{e} \geq 3.5\%$$

- No bias related to training additional experts with clusters that mix different levels of expertise

	Expert A	Expert B	Expert C	Expert D
Expert A	1			
Expert B	0.7808			
Expert C	0.7162	0.8034	1	
Expert D	0.8342	0.7811	0.7473	1

- The agreement between the different experts does not reveal chance but shows a strong substantial agreement



- No expert is outlier, which means that they all have a comparable and consolidated approach

$$F_1 \text{ mean} \approx 81\%$$

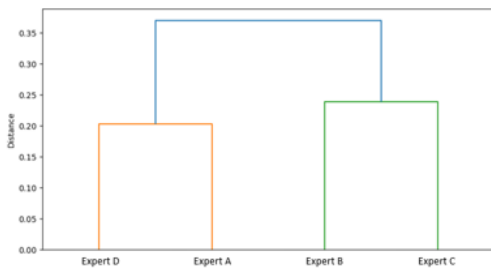


Figure 7-17: Overview of Chapter 7 - Evaluation of annotator agreement among the validation pool



## Chapter 8. Matrix Profile Evaluation

---

Matrix Profile is trained using turbidity data collected at Cottage between February 2021 and September 2021. Performance evaluation of Matrix Profile differs from that of traditional supervised models. Rather than having separate sets of training and test data, MP generally exploits the full set of data available for learning. The aim is to identify underlying structures in the time series that may indicate abnormal events. As a result, the model is generally evaluated over the entire available time range. The absence of this distinction in the context of Matrix Profile is supported by the exploratory nature of the model: to discover unusual patterns or behaviors rather than to predict specific outcomes.

The following sections, such as sensitivity to input data ( [Section 8.1](#)) with data preprocessing and input data selection, as well as hyperparameter optimization ([Section 8.2](#)), are key elements of the methodology. The learning and evaluation process is centered on the use of sliding sequences (sequence size being one of the model's hyperparameters), where a given timestamp can receive different labels. In order to interpret these results, postprocessing is performed, enabling all time steps constituting an invalid sequence to be instantly flagged as such. This eliminates the need to assign labels to sequences as a whole and directly exploit the results of manual validation (filtering + expertise + aggregation), although further testing is envisaged in [Section 8.3.3](#). Other potential improvements to the results, discussed in [Section 8.3](#), include the use of ensemble models and pre-validation steps. The chapter goes on to explore broader aspects such as generalization to other sites ([Section 8.4](#)) and multivariate anomaly detection ([Section 8.5](#)).

### 8.1. Sensitivity to input data

The following tests delve into the evaluation of sensitivity to input data, focusing on various preprocessing steps essential for initiating Matrix Profile for anomaly detection. The experiments primarily revolve around imputation techniques for missing data, encompassing considerations such as sub-sampling and data smoothing. Moreover, we explore the sensitivity of Matrix Profile to different input data sources, focusing on raw turbidity data and reconstructed turbidity such as described in [Section 4.4.1](#). The purpose is to define the type of data to be used as input, and the processing steps required before using the MP model.

#### 8.1.1. Preprocessing

The first tests concern the pre-processing approaches to be implemented before using the model. **Data scaling** has not been implemented since it is implicitly established when using



MP. On the other hand, **data segmentation** with a constant time window is mandatory. This point will be handled during the training of the model since the window size is a hyperparameter of the model. Thus, the tests mainly concern the **imputation techniques for missing data** and the possibility of **sub-sampling** and the **smoothing of the data**. These experiments are conducted using a window-length of 576 (48 hours) and an anomaly ratio of 10%.

#### 8.1.1.1. Missing values imputation

Although we have previously admitted that the adequate method for replacing missing data is to fill it with zeros so as to preserve its abnormality, it is important to test this hypothesis by taking advantage of Matrix Profile's distinctive advantage in this context. Unlike the other two evaluated models, MP demonstrates an ability to run even in the presence of data gaps. So, as part of our approach, we initiated tests to determine the best approach for imputing missing data. **Table 18** synthesizes the results using different imputation techniques.

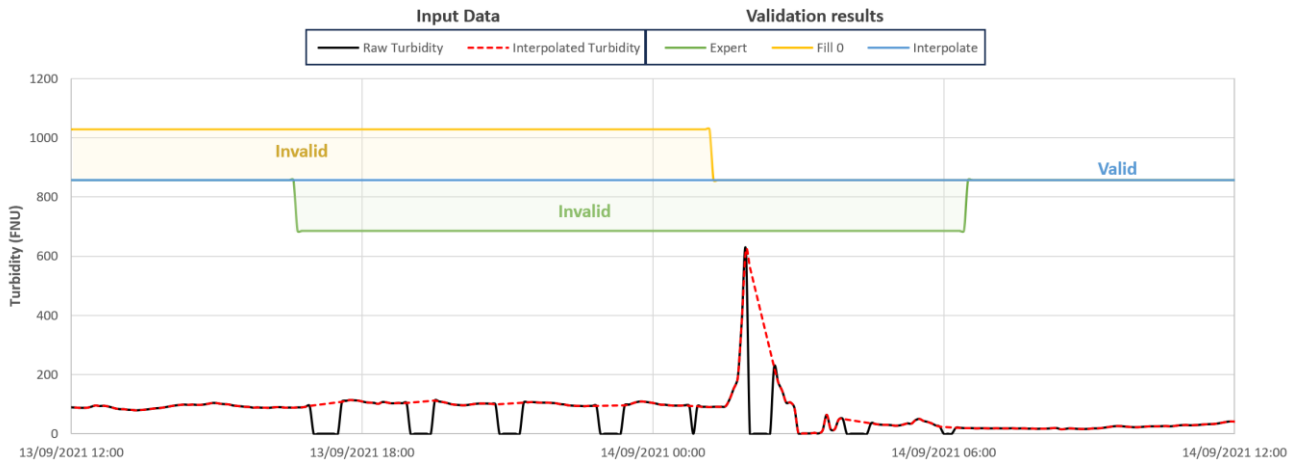
**Table 18: Missing values imputation techniques results**

	Precision	Recall	F1 score	MCC
<b>No filling</b>	0.360	0.327	0.343	0.254
<b>Filling with 0</b>	0.689	0.633	0.660	0.615
<b>Interpolation</b>	0.661	0.626	0.643	0.594

Looking at the table of results, it becomes clear that not replacing missing data generates the poorest performance. Despite Matrix Profile's ability to operate under such conditions, in practice the model ignores all sequences with missing data. With a window size equal to 48 hours, one missing value is enough to neglect 96 hours of measurement (48 hours before and after the missing value included); this corresponds to 1152 subsequences. This statement explains the poor score obtained in the case where no imputation is set up.

As for imputation with zeros and interpolation of missing data, the scores are equivalent with a slight advantage in the case of replacing missing data with zeros. **Figure 8-1** illustrates a case where the latter approach is advantageous for MP. In this example, the sensor experiences several successive disconnections during which it does not transmit any measurement. In the meantime, the signal automatically recovers and sends measurement. In this configuration, the domain expert considers that there is no way to evaluate the reliability of these measured data and invalidates the whole period that includes these repetitive malfunctions. In the case of missing data interpolation, the algorithm does not see them anymore and has to deal with a chronicle that is rather consistent. On the other hand, replacing these data with zeros, injects MP with a chronicle with successive and abnormal falls to zero

for a turbidity pattern. This configuration allows the algorithm to select the period surrounding this pattern as abnormal. It should be noted, however, that the period following the peak appears consistent to the model and does not identify it as an anomaly, which illustrates a first case of false negatives.



**Figure 8-1: Results of different missing values imputation techniques.**

**Bottom: (in black), the raw turbidity measurement, (in red) the interpolated turbidity. Top: the Boolean time series representing the data validation result according to the domain expert (in green) and the matrix profile algorithm using interpolation (in blue) and imputation with zeros (in yellow).**

Hence, if missing data is interpolated, this can lead to a false negative when evaluating the model. In other words, the model may fail to perceive any apparent problem in the sequence, as the interpolation masks the gaps, creating a biased and potentially underestimated assessment of model performance. We conclude that missing data imputation with zeros is the most convenient approach to fill in missing values in turbidity time series, as it reproduces the assessment performed by experts.

Meanwhile, the detection of missing data can be efficiently performed using simple Boolean tests. With this in mind, a pre-validation step will be incorporated to identify and flag missing values in data sequences.

#### 8.1.1.2. Downsampling

The acquisition strategy currently implemented by SMA consists of taking a measurement every 20 seconds and calculating an average value every 5 minutes. It is this last measurement that is recorded and exploited in this study. In a degraded operational scenario, there is consideration for initiating measurements only every X minute, maintaining the same acquisition strategy of averaging over the preceding 5 minutes. In practical terms, only

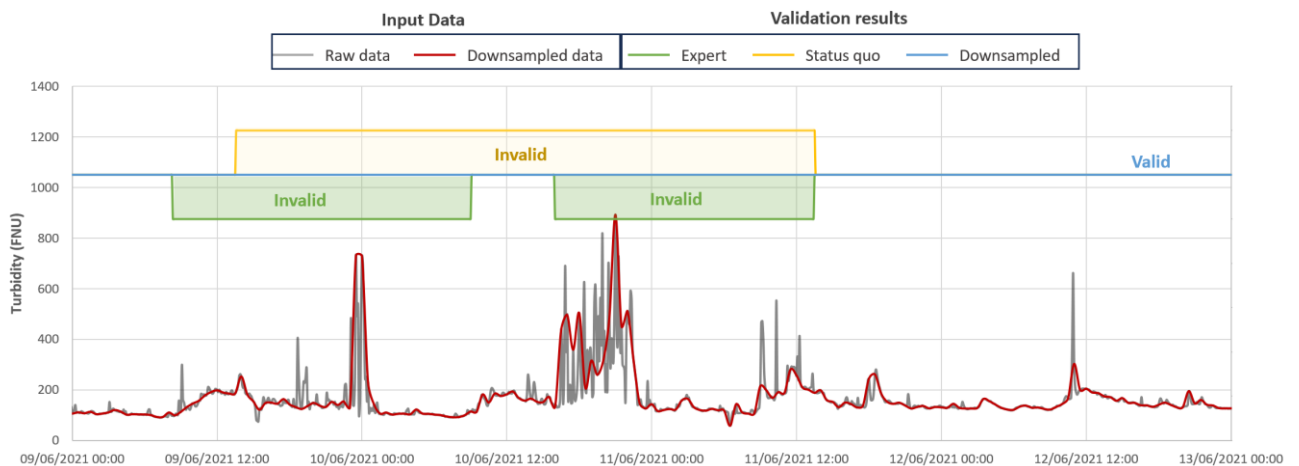
measurements aligning with the defined frequency are retained, and the remainder are discarded. The label assigned by the expert is retained based on the preserved timestamp.

**Table 19** synthesizes the results. Downsampling relatively degrades the performance of the model at 15 minutes and significantly deteriorates it at 30 minutes frequency.

**Table 19: Downsampling results**

Downsampling frequency X	Precision	Recall	F1 score	MCC
Status quo	0.689	0.633	0.660	0.615
15 min	0.643	0.594	0.618	0.563
30 min	0.449	0.413	0.430	0.354

**Figure 8-2** shows the impact of downsampling at 30 minutes on anomaly detection. The same result is also observed using a downsampling frequency of 15 minutes. Indeed, the algorithm without downsampling (status quo) identifies the whole period from the 9<sup>th</sup> of June at 8:20 am to the 11<sup>th</sup> of June at 8:20 am (which corresponds to a 48-hour subsequence) as discord. Whereas the algorithm with an under sampled time series as input does not notice the anomaly at all. The downsampling relatively smoothens the measurement to the point where it is no longer aberrant and can be assimilated to a normal pattern of rainy weather.



**Figure 8-2: The results of downsampling on anomaly detection.**

**Bottom:** (in gray) the raw turbidity and (in red) the downsampled turbidity using a 30-minute frequency . **Top:** the Boolean time series representing the data validation result according to the domain expert (in green) and the matrix profile algorithm using raw data (in yellow) and downsampled data (in blue).

These tests provide a general trend, but the results lack complete accuracy as the output target was not specifically defined for this database; instead, it was simply downsampled in alignment with the input time series. Nevertheless, it can be assumed that, for detecting rapid

disturbances in turbidity data, the 5-minute acquisition frequency appears to be the most relevant, and this is indeed recommended by the [37] guide.

#### 8.1.1.3. Data smoothing

As already explained in [Section 4.4.1](#), the strategy of manual data validation is mainly based on the calculation of the redundancy criterion using sliding centered average values on 13-time steps (1/2 hour before and after each measure). This approach allows to get rid of useless noise, intrinsic to the measurement. Thus, it is interesting to assess the same approach using MP. In this work, we evaluated 3 different smoothing window lengths. The results described in [Table 20](#) show that smoothing data degrades the results.

**Table 20: Data smoothing results**

Smoothing window	Precision	Recall	F1 score	MCC
Status quo	0.689	0.633	0.660	0.615
3-time steps	0.516	0.563	0.539	0.476
13-time steps	0.477	0.439	0.457	0.384
25-time steps	0.414	0.380	0.396	0.314

This outcome is closely tied to the nature of anomalies recognized by the domain expert. In certain instances, the anomaly is specifically characterized by the presence of a significant noise in the measurement. The alert thresholds for the manual validation's filtering phase are triggered by the centered moving average of turbidity. However, this does not exempt the expert from examining the raw data during the expertise phase. In the MP model, using a univariate model and inputting only the smoothed data results in the algorithm losing access to the raw data. The smoothing process "absorbs" this noise, hindering the identification of defects. Consequently, the peaks can be likened to a rainy event. It is worth noting that in this case as well, there is a bias associated with the output target, which has not been adjusted to the smoothed input data but has been retained in its original form.

#### 8.1.1.4. Conclusion

This section aimed to outline the preprocessing steps essential for initiating MP. Common to various anomaly detection algorithms for time series, resampling data to maintain a consistent time step and scaling data are crucial. Matrix profile inherently incorporates data standardization, eliminating the need for additional preprocessing. To ensure a complete data chronicle while preserving dropout phenomena resulting from data gaps, missing values were filled with zeros. For accurate anomaly identification, a sufficiently fine acquisition frequency is necessary. It is crucial not to smooth noise, as pronounced noise in turbidity measurements

within sewerage networks is considered an anomaly, potentially indicating probe malfunction. These preprocessing approaches will be applied universally across all our tests.

### 8.1.2. Input data

The objective of this section is to analyze the sensitivity of Matrix Profile to diverse input data. This involves two aspects: firstly, assessing the model's performance using unprocessed turbidity data from the two turbidimeters, and secondly, evaluating the model's performance using the reconstructed turbidity as detailed in [Section 4.4.1](#). To achieve this, we use a sequence length of 48 hours and an anomaly rate of 10%. For raw data, the output target is directly derived from the expert-established validation result. In contrast, for reconstructed turbidity, the output target is a composite of individual labels obtained through combinations outlined in [Table 6](#).

[Table 21](#) presents a comparison of data validation results, considering various input data while utilizing consistent hyperparameters.

**Table 21: Performance metrics for different input data using the same hyperparameters**

Input data	Precision	Recall	F1 score	MCC
Turbidity 1	0.674	0.399	0.501	0.430
Turbidity 2	0.645	0.359	0.462	0.383
Reconstructed T	0.689	0.633	0.660	0.615

The results reveal important nuances in the performance of the MP model according to the different input data sources. Firstly, when examining the performance metrics using raw data (Turbidity 1 and Turbidity 2), the model shows poor performance compared to that using the reconstructed turbidity. It is noteworthy that the main disparity between the various results lies in recall, representing the number of invalid sequences that the model failed to identify. Indeed, this problem can be directly attributed to the tuning of the hyperparameters. Analysis of the output target for each set of input data reveals that the raw turbidity logs have an anomaly rate around 20%, while the reconstructed turbidity logs have a lower anomaly rate of 12% approximately. Consequently, it is clear that the anomaly ratio  $k$  imposed on the model is more restrictive for raw data, resulting in the non-detection of many anomalous sequences, given that the number of anomalies to be reported has been conditioned. This problem will be assessed in [Section 8.2](#), where we will fine-tune the hyperparameters specific to each input dataset.

In addition, overall observation reveals a precision around 0.65. In other words, if we use the model results as a pre-selection, the expert will have to examine 35% of additional sequences in order to spot the 65% actually invalid sequences. Thus, it is crucial to determine the comparative effectiveness of this approach versus the filtering tool that relies on turbidity redundancy. Ultimately, the precision of the latter is 0.39, indicating that the use of the Matrix Profile still represents an advantage in reducing the potential number of false alarms.

### 8.2. Hyperparameters tuning

As part of performance evaluation of the MP model, we have undertaken hyperparameters tuning tests. This process aims to identify, using a grid search, the optimum pair window-size / anomaly ratio that maximizes the model's ability to effectively detect anomalies in wastewater network turbidity data (provided the reliability of the reference).

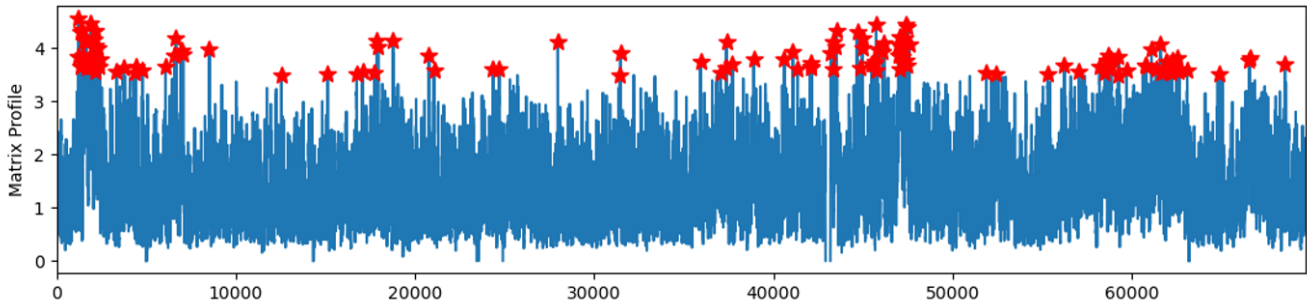
In order to facilitate the understanding of the obtained results from these tests, we have adopted a visual approach using a heat map. This graphical representation provides a concise visualization of the variations in model performance as a function of the different hyperparameter combinations tested. A green cell corresponds to an F1 score higher than 0.5 while a red cell has an F1 score lower than 0.5. The input data used in [Figure 8-3](#) is the reconstructed turbidity at Cottage, however the same strategy was also implemented using raw turbidity data

Window size	Anomaly ratio																														
	0.05	0.055	0.06	0.065	0.07	0.075	0.08	0.085	0.09	0.095	0.1	0.105	0.11	0.115	0.12	0.125	0.13	0.135	0.14	0.145	0.15	0.155	0.16	0.165	0.17	0.175	0.18	0.185	0.19	0.195	0.2
2	0.366	0.363	0.38	0.373	0.37	0.371	0.37	0.389	0.4	0.405	0.41	0.408	0.42	0.417	0.42	0.422	0.42	0.411	0.41	0.409	0.4	0.402	0.4	0.397	0.4	0.391	0.39	0.385	0.38	0.382	0.38
4	0.393	0.397	0.39	0.398	0.39	0.399	0.41	0.412	0.41	0.412	0.42	0.445	0.44	0.439	0.43	0.437	0.43	0.431	0.43	0.426	0.42	0.415	0.41	0.411	0.4	0.41	0.4	0.4	0.4	0.4	0.41
6	0.408	0.408	0.43	0.422	0.43	0.435	0.43	0.424	0.42	0.431	0.45	0.435	0.44	0.443	0.44	0.443	0.45	0.449	0.44	0.439	0.44	0.431	0.43	0.445	0.44	0.437	0.44	0.43	0.42	0.427	0.43
8	0.460	0.479	0.51	0.508	0.51	0.524	0.52	0.526	0.53	0.516	0.5	0.504	0.5	0.496	0.49	0.477	0.48	0.48	0.47	0.464	0.47	0.461	0.46	0.455	0.46	0.454	0.45	0.441	0.43	0.428	0.43
10	0.418	0.447	0.45	0.46	0.48	0.485	0.49	0.478	0.48	0.501	0.49	0.494	0.5	0.501	0.49	0.507	0.5	0.487	0.48	0.47	0.47	0.465	0.46	0.463	0.46	0.448	0.45	0.442	0.43	0.428	0.42
12	0.417	0.42	0.47	0.479	0.49	0.514	0.5	0.49	0.48	0.491	0.51	0.518	0.53	0.524	0.51	0.504	0.5	0.5	0.51	0.525	0.53	0.529	0.54	0.528	0.52	0.51	0.5	0.497	0.49	0.494	0.49
14	0.417	0.457	0.45	0.483	0.49	0.501	0.51	0.52	0.53	0.519	0.52	0.549	0.56	0.547	0.55	0.545	0.55	0.557	0.55	0.537	0.53	0.52	0.52	0.522	0.51	0.518	0.53	0.517	0.51	0.531	0.52
16	0.484	0.503	0.52	0.503	0.52	0.519	0.51	0.517	0.53	0.54	0.53	0.515	0.53	0.543	0.53	0.525	0.51	0.52	0.52	0.537	0.53	0.517	0.51	0.503	0.53	0.518	0.53	0.522	0.52	0.509	0.5
18	0.521	0.513	0.53	0.52	0.55	0.538	0.53	0.515	0.51	0.501	0.51	0.507	0.52	0.521	0.53	0.543	0.56	0.547	0.55	0.552	0.54	0.548	0.54	0.531	0.54	0.527	0.52	0.512	0.52	0.529	0.52
20	0.451	0.479	0.5	0.489	0.51	0.498	0.48	0.481	0.47	0.459	0.49	0.512	0.5	0.518	0.51	0.518	0.54	0.54	0.53	0.537	0.55	0.541	0.54	0.533	0.52	0.515	0.5	0.498	0.49	0.483	0.47
22	0.557	0.546	0.54	0.553	0.54	0.533	0.52	0.513	0.5	0.487	0.48	0.472	0.5	0.515	0.53	0.543	0.55	0.567	0.56	0.546	0.57	0.558	0.54	0.537	0.53	0.542	0.54	0.529	0.52	0.513	0.51
24	0.575	0.563	0.57	0.556	0.59	0.563	0.55	0.548	0.54	0.529	0.55	0.562	0.58	0.571	0.55	0.544	0.56	0.549	0.54	0.525	0.55	0.538	0.53	0.523	0.54	0.529	0.52	0.515	0.51	0.496	0.49
26	0.588	0.574	0.56	0.549	0.54	0.526	0.54	0.564	0.57	0.564	0.58	0.591	0.58	0.586	0.58	0.606	0.63	0.656	0.65	0.636	0.63	0.629	0.62	0.611	0.6	0.584	0.58	0.568	0.56	0.553	0.55
28	0.491	0.525	0.51	0.547	0.56	0.547	0.58	0.574	0.6	0.633	0.62	0.608	0.63	0.613	0.6	0.591	0.59	0.593	0.59	0.578	0.57	0.573	0.56	0.555	0.55	0.538	0.53	0.522	0.51	0.536	0.53
30	0.459	0.5	0.49	0.525	0.51	0.548	0.58	0.565	0.58	0.596	0.58	0.582	0.61	0.6	0.59	0.576	0.58	0.574	0.56	0.553	0.55	0.539	0.54	0.564	0.55	0.546	0.57	0.582	0.57	0.564	0.56
32	0.525	0.525	0.51	0.497	0.48	0.518	0.51	0.531	0.57	0.6	0.61	0.639	0.64	0.625	0.61	0.599	0.61	0.594	0.6	0.59	0.58	0.569	0.56	0.558	0.55	0.543	0.53	0.546	0.54	0.529	0.52
34	0.492	0.537	0.52	0.507	0.51	0.493	0.53	0.515	0.52	0.549	0.54	0.536	0.57	0.558	0.55	0.578	0.57	0.572	0.57	0.561	0.55	0.538	0.53	0.522	0.51	0.512	0.5	0.494	0.49	0.477	0.47
36	0.502	0.502	0.55	0.532	0.52	0.502	0.5	0.488	0.52	0.511	0.54	0.536	0.52	0.559	0.55	0.533	0.53	0.521	0.52	0.53	0.52	0.513	0.51	0.502	0.49	0.482	0.47	0.473	0.46	0.456	0.45
38	0.460	0.511	0.56	0.559	0.54	0.525	0.51	0.51	0.5	0.482	0.52	0.558	0.56	0.586	0.57	0.557	0.56	0.544	0.57	0.556	0.56	0.544	0.53	0.525	0.52	0.514	0.5	0.493	0.48	0.484	0.47
40	0.454	0.454	0.51	0.558	0.54	0.54	0.52	0.507	0.52	0.521	0.51	0.536	0.54	0.575	0.56	0.546	0.55	0.532	0.56	0.544	0.54	0.531	0.52	0.519	0.51	0.496	0.49	0.492	0.48	0.472	0.46
42	0.420	0.475	0.53	0.529	0.58	0.557	0.54	0.539	0.52	0.561	0.56	0.588	0.57	0.572	0.61	0.595	0.6	0.579	0.56	0.55	0.55	0.537	0.52	0.524	0.52	0.524	0.52	0.512	0.5	0.501	0.51
44	0.426	0.482	0.48	0.533	0.58	0.577	0.56	0.596	0.6	0.623	0.6	0.604	0.59	0.569	0.57	0.553	0.59	0.593	0.58	0.562	0.56	0.57	0.56	0.555	0.54	0.529	0.53	0.516	0.51	0.508	0.51
46	0.431	0.431	0.49	0.544	0.54	0.595	0.63	0.632	0.61	0.611	0.64	0.622	0.62	0.602	0.63	0.628	0.61	0.592	0.59	0.631	0.63	0.619	0.6	0.602	0.59	0.572	0.57	0.558	0.56	0.565	0.56
48	0.436	0.436	0.5	0.5	0.55	0.603	0.6	0.64	0.64	0.678	0.66	0.66	0.64	0.639	0.62	0.599	0.6	0.641	0.64	0.663	0.66	0.658	0.65	0.645	0.63	0.612	0.61	0.617	0.62	0.602	0.59
50	0.370	0.441	0.44	0.487	0.47	0.472	0.45	0.453	0.5	0.553	0.55	0.534	0.53	0.516	0.52	0.504	0.55	0.552	0.55	0.552	0.54	0.549	0.55	0.533	0.53	0.54	0.54	0.526	0.51	0.513	0.5
52	0.373	0.446	0.45	0.508	0.51	0.548	0.55	0.531	0.51	0.51	0.55	0.554	0.6	0.604	0.6	0.603	0.59	0.603	0.6	0.584	0.58	0.566	0.57	0.55	0.55	0.556	0.54	0.54	0.53	0.526	0.51
54	0.460	0.46	0.45	0.448	0.51	0.547	0.55	0.53	0.53	0.509	0.51	0.563	0.56	0.605	0.6	0.621	0.62	0.619	0.6	0.599	0.58	0.583	0.57	0.57	0.55	0.553	0.56	0.558	0.54	0.543	0.53
56	0.376	0.376	0.45	0.453	0.51	0.51	0.55	0.548	0.6	0.601	0.58	0.582	0.56	0.559	0.54	0.538	0.58	0.624	0.62	0.64	0.64	0.625	0.62	0.61	0.61	0.592	0.59	0.595	0.6	0.579	0.58
58	0.379	0.379	0.42	0.425	0.41	0.409	0.39	0.39	0.44	0.444	0.43	0.426	0.47	0.472	0.53	0.527	0.55	0.546	0.53	0.534	0.52	0.521	0.5	0.504	0.55	0.546	0.53	0.53	0.51	0.515	0.5
60	0.406	0.489	0.49	0.465	0.47	0.505	0.5	0.486	0.49	0.465	0.47	0.446	0.45	0.49	0.49	0.545	0.55	0.562	0.56	0.544	0.54	0.525	0.53	0.525	0.51	0.512	0.55	0.554	0.54	0.537	0.52
62	0.397	0.482	0.48	0.47	0.47	0.451	0.45	0.451	0.49	0.49	0.53	0.535	0.51	0.512	0.49	0.491	0.55	0.545	0.53	0.525	0.54	0.542	0.54	0.523	0.52	0.506	0.51	0.549	0.55	0.536	0.54
64	0.401	0.401	0.49	0.488	0.48	0.48	0.46	0.46	0.5	0.495	0.54	0.538	0.54	0.515	0.51	0.493	0.49	0.546	0.55	0.527	0.53	0.507	0.51	0.507	0.49	0.489	0.51	0.505	0.55	0.55	0.55
66	0.405	0.405	0.4	0.4	0.44	0.44	0.42	0.422	0.42	0.468	0.47	0.511	0.51	0.488	0.49	0.488	0.47	0.468	0.52	0.521	0.5	0.502	0.48	0.483	0.48	0.466	0.47	0.475	0.48	0.527	0.53
68	0.408	0.408	0.4	0.403	0.4	0.442	0.44	0.423	0.42	0.469	0.47	0.469	0.51	0.51	0.49	0.487	0.47	0.467	0.47	0.519	0.52	0.5	0.5	0.481	0.48	0.481	0.46	0.465	0.48	0.476	0.53
70	0.411	0.411	0.41	0.452	0.45	0.444	0.44	0.425	0.42	0.425	0.47	0.47	0.45	0.447	0.45	0.485	0.49	0.539	0.54	0.517	0.52	0.517	0.5	0.497	0.48	0.478	0.48	0.535	0.53	0.545	0.54

Figure 8-3: Grid Search Results to identify best hyperparameters for Cottage Dataset.

A green cell corresponds to an F1 score higher than 0.5 while a red cell has an F1 score lower than 0.5. The framed cell correspond to the best hyperparameters pair.

Therefore, the MP model in this case is rather overly sensitive to the window size than to the anomaly rate. Indeed, a variation of the rate of 1% adds hardly one abnormal subsequence for a window size equal to 48h and 5 subsequences for a window size equal to half a day. For small window sizes, the matrix profile is too noisy with a small range of variability; all subsequences are more or less similar. This configuration does not allow to identify peaks nor drops corresponding to anomalies and motifs (see [Figure 8-4](#)).



**Figure 8-4: Matrix profile for turbidity data with a window length of 2 hours and an anomaly ratio of 0.5%. Red stars point potential anomalies**

**Table 22** summarizes the results and the optimal parameters issued from the different input data.

**Table 22: Performance results of MP using different input data**

Dataset	Best hyperparameters		Metrics			
	Window size	Anomaly rate	Precision	Recall	F1 score	MCC
<b>Turbidity 1</b>	44 hours	15.5%	0.646	0.528	0.581	0.486
<b>Turbidity 2</b>	28 hours	19%	0.555	0.511	0.532	0.404
<b>Reconstructed T</b>	48 hours	9.5%	0.729	0.633	0.678	0.637

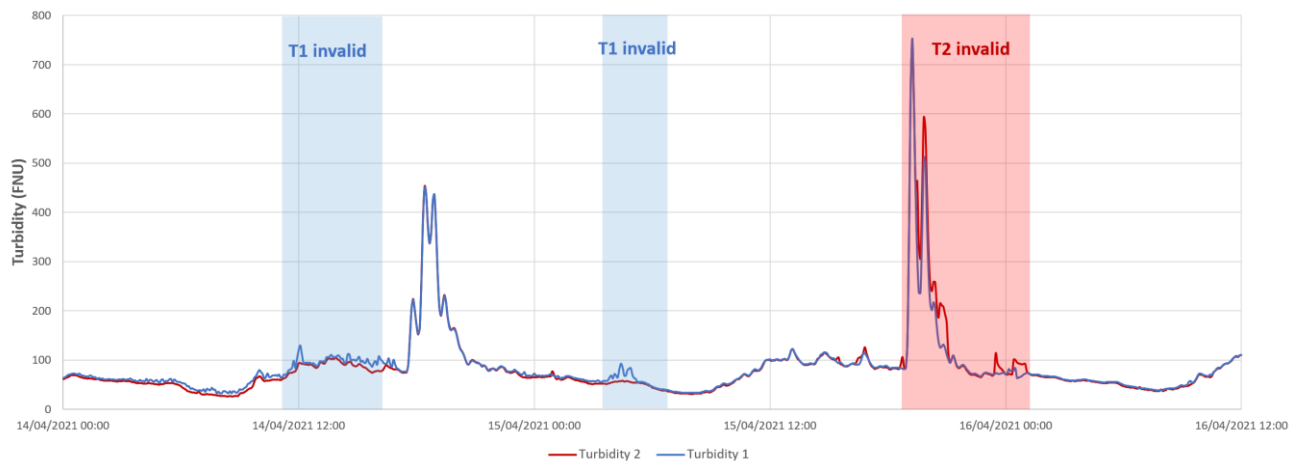
As a result, the optimal parameters are different depending on the studied data. The best window length for the first turbidimeter is 44 hours while for the other one, the optimal window size is of 28 hours. Moreover, the first chronicle has an optimal anomaly rate of 15% whereas the second has a higher anomaly ratio of 19% approximately. A thorough check was undertaken to determine whether one sensor is more flawed than the other, however, no significant disparity was found. This result therefore suggests a potential complexity in calibrating a universal model if hyperparameters can vary even on a local scale. However, the results of window size calibration using raw data remain broadly consistent with the results of grid research applied to reconstructed turbidity, where two optimal window size ranges are identified for the model, at around 24 hours and 48 hours (see [Figure 8-3](#)). Thus, although the



hyperparameters may differ, they nevertheless appear to fall within a predefined range of optimal performance.

Overall, the results obtained using raw turbidity remain lower than those obtained using reconstructed turbidity. This problem can be attributed to the filtering phase of manual validation, which sometimes invalidates the turbidimeter with the highest turbidity, even though its measurements are not necessarily outliers (see [Figure 8-5](#)). This introduces a bias into the raw data output target, which is avoided when using reconstructed turbidity, where an invalid label effectively corresponds to an outlier sequence identified by the domain expert. As a result, detected anomalies are more reliable when using reconstructed turbidity.

In practice, abnormal sequences in the reconstructed turbidity chronicle correspond to periods when both turbidimeters fail simultaneously. Thus, one approach to improvement would be to merge the validation results of the two raw turbidities, identifying only common anomalies to both sensors. This point will be explored in [Section 8.3.1](#).



**Figure 8-5: Example of manual invalid sequences and of the bias introduced by the filtering phase and the redundancy criterion**

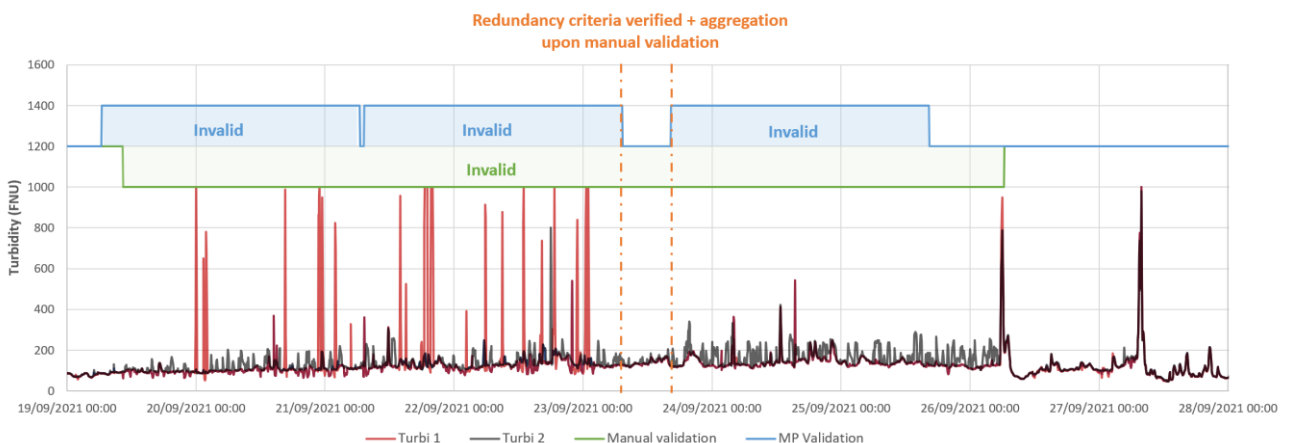
At present, the aim is to analyze and diagnose the results obtained using reconstructed turbidity, which has demonstrated better performance. [Figure 8-6](#) compares the results of the manual validation and the MP model.



**Figure 8-6: Comparison of validation results by the domain expert (in Orange) and the algorithm (in Green) using Boolean time series.**

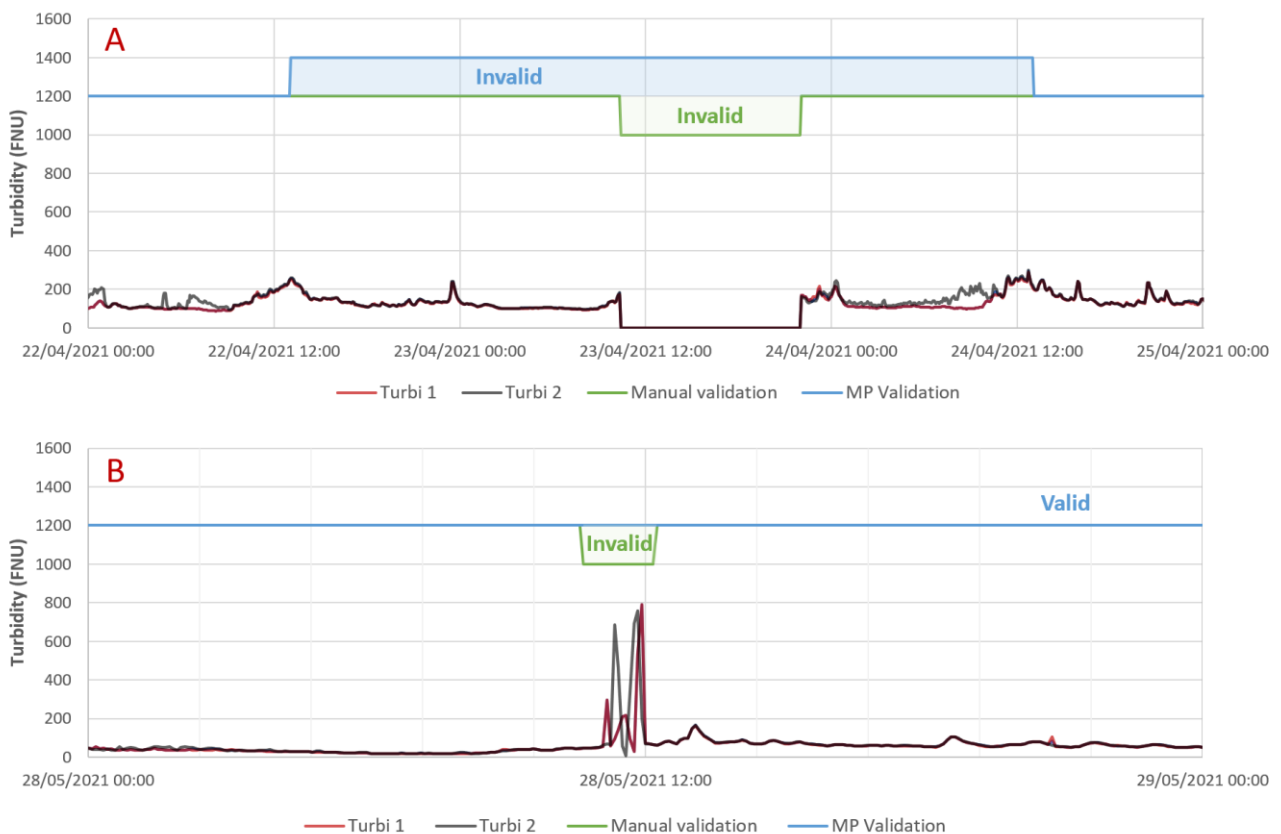
It is remarkable that the main mismatch between the model and the manual validation lies in the delimitation of defects. This discrepancy is attributable to one of the biases inherent in manual validation. Indeed, **delimitation problems** in the reference can arise from a variety of factors. One of the main aspects to consider is the subjective process. Individual interpretations and the complex nature of the data can make it difficult to delimit anomalies precisely, especially when they manifest themselves gradually, evolve slowly or have ambiguous characteristics. To illustrate this, a concrete example is shown in [Figure 8-2](#). The discrepancy between the results of the model and those of manual validation highlights a lag in the identification of the start and the end of the abnormal sequence.

Moreover, in some configurations, the fault aggregation phase merges consecutive anomalies according to the criteria described in [Section 4.4.1](#). This post-processing step is not implemented when using MP. As a result, some false negatives are linked to periods between two merged anomalies, even though they may actually be valid (see [Figure 8-7](#)).



**Figure 8-7: Example of anomalies fusion while the in-between subsequence is valid**

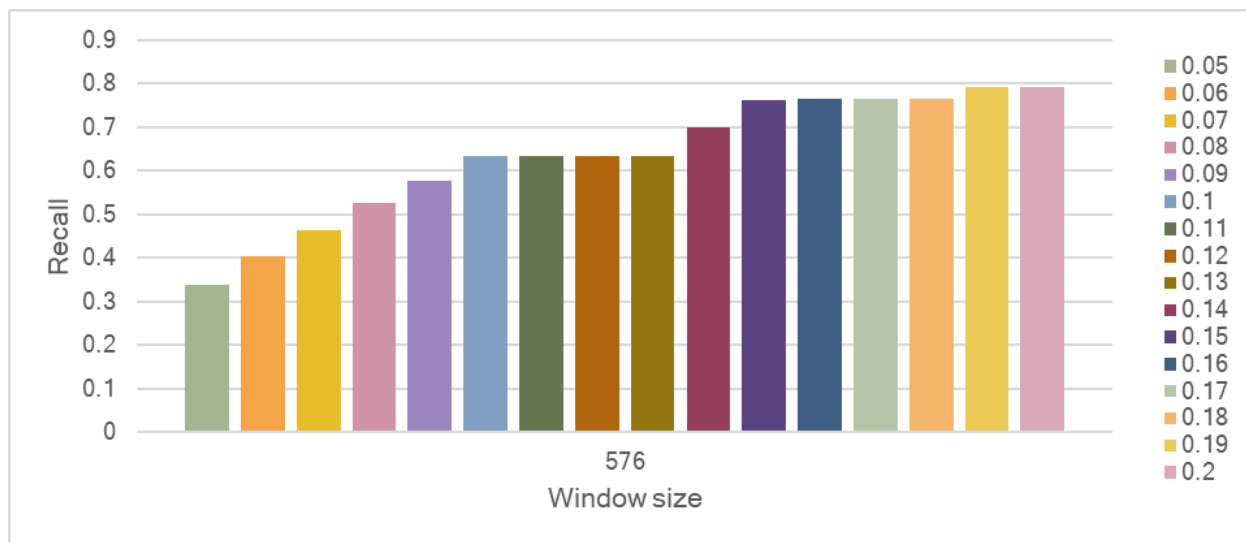
In addition, the problem of fault delimitation is also linked to the fact that MP imposes a **fixed sequence size**. On the one hand, as soon as a sequence is invalid, all the time steps that make it up are considered invalid. However, sometimes the anomaly is truly local and does not extend across the entire sequence, thus leading to false positives when comparing the results with the output target from manual validation (see [Figure 8-8-A](#)). On the other hand, imposing a fixed sequence size makes it difficult to detect anomalies of shorter duration. In accordance with the principle of similarity join, the anomaly must be large enough when calculating the distance to invalidate the sequence as a whole. Consequently, if only a few points are erroneous, they will have little impact on the matrix profile and may result in false negatives. [Figure 8-8-B](#) shows an example with an anomaly that lasted 1 hour and 25 minutes and which was not identified by the algorithm. This observation highlights the fact that anomalies in sewerage networks vary in duration and amplitude. Consequently, using a fixed window size can lead to misclassification in both directions, generating both false positives and false negatives. It therefore becomes interesting to adopt a multi-window size approach in order to detect anomalies of various durations (see [Section 8.3.2](#)).



**Figure 8-8: Example of bias introduced by a fixed window size using Matrix Profile**

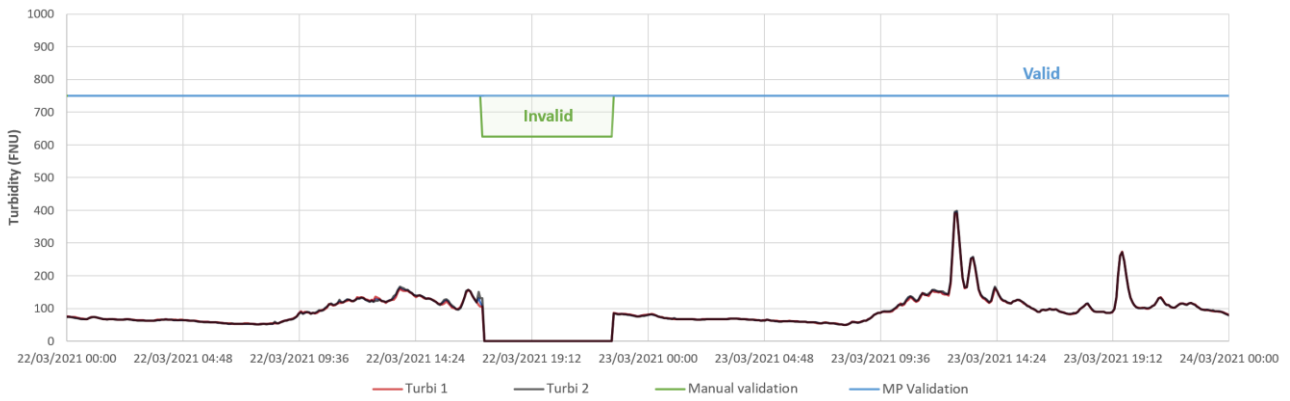
Moreover, apart from defect delineation, we also observe that some anomalies are omitted by the MP model. The first reason, as mentioned above, is linked to the size of the window, which can be significantly larger than the defect, making the latter less detectable. Another aspect is linked to the imposition of a specific number of defects or rate of anomalies to be identified. As a result, the model selects only those anomalies that show the greatest divergence from their nearest neighbor.

**Figure 8-9** illustrates that increasing the anomaly rate for a given window size, here 48 hours, progressively improves recall. However, it is important to note that increasing the anomaly rate can lead to a decrease in precision. Thus, the challenge lies in establishing an optimal balance between recall and precision to guarantee effective anomaly detection.



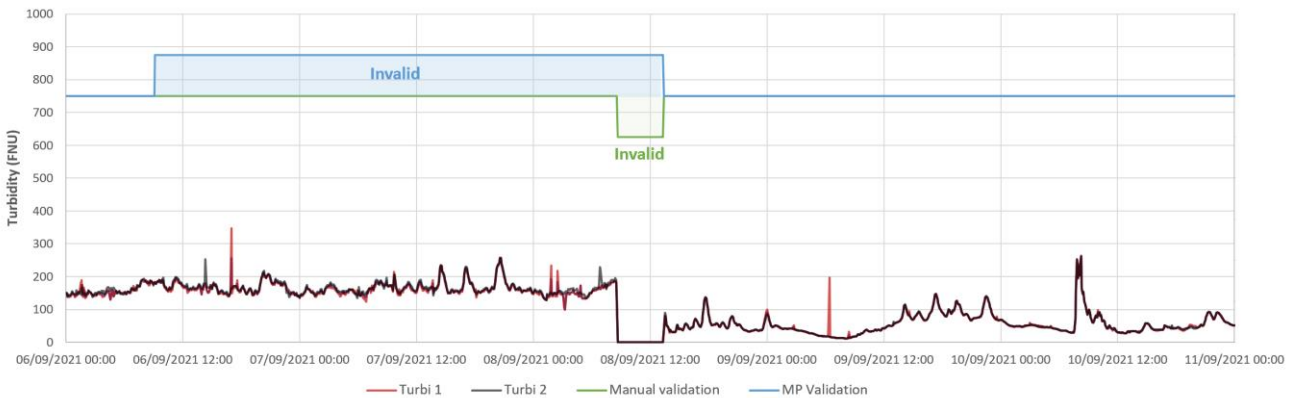
**Figure 8-9: Recall of matrix profile according to the defined anomaly rate**

On the other hand, some false negatives are linked to the very definition of the anomaly. For example, **Figure 8-10** shows that the model fails to identify the period of missing data as abnormal. This is because this type of sequence is not considered unique, and as a result, MP manages to find similarities in the chronicle, limiting its ability to identify these periods as abnormal. Unlike the example shown in **Figure 8-8**, the period of missing data is shorter here, and this type of fault occurs regularly. By contrast, in **Figure 8-8**, the missing data is spread over half a day, so it's rare to observe such a prolonged interruption. This highlights one of the limitations of Matrix Profile, which considers anomaly to be a sequence that is unique. Consequently, a defect that is repeated identically is no longer identifiable by the model. In general, turbidity data is so dynamic that it is difficult to find exactly repeated faults. So, while the use of Matrix Profile is not called into question, it is essential to add a pre-validation step to exclude all **trivial anomalies** (this point will be assessed in **Section 8.3.3**).



**Figure 8-10: Example of trivial repetitive anomalies non-identified by the MP model**

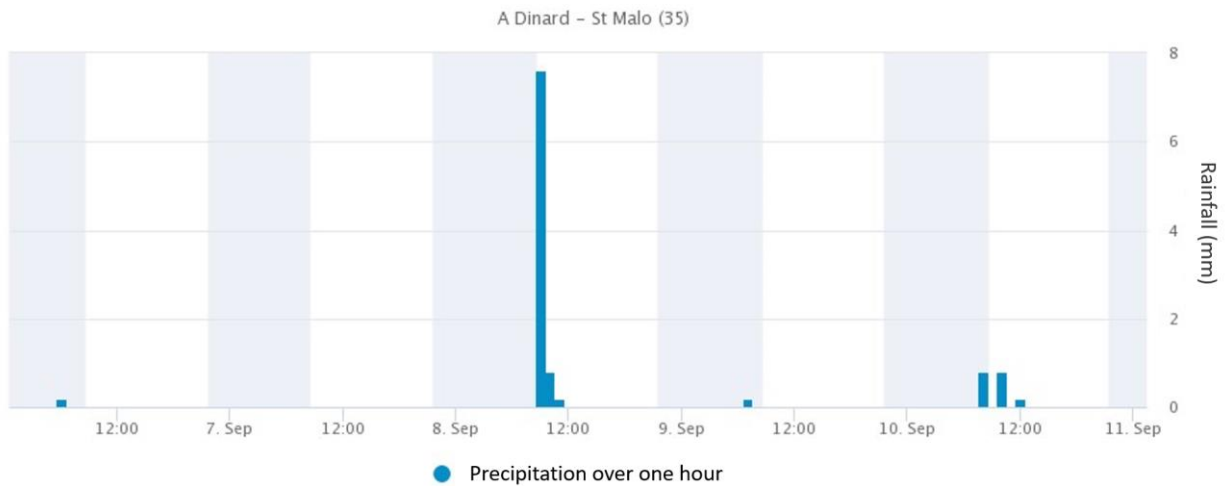
**Figure 8-11** highlights an intriguing case of false positives: during the manual validation process, we identified only the missing sequence as abnormal, while the MP model identified the entire period in blue as abnormal. One might assume that this is a problem of fault delineation, but with hindsight, the pattern nevertheless appears abnormal, characterized by fluctuating turbidity, a drop to 0, followed by smoother and lower turbidity values.



**Figure 8-11: Anomalies delimitation problem between the expert and the algorithm validation**

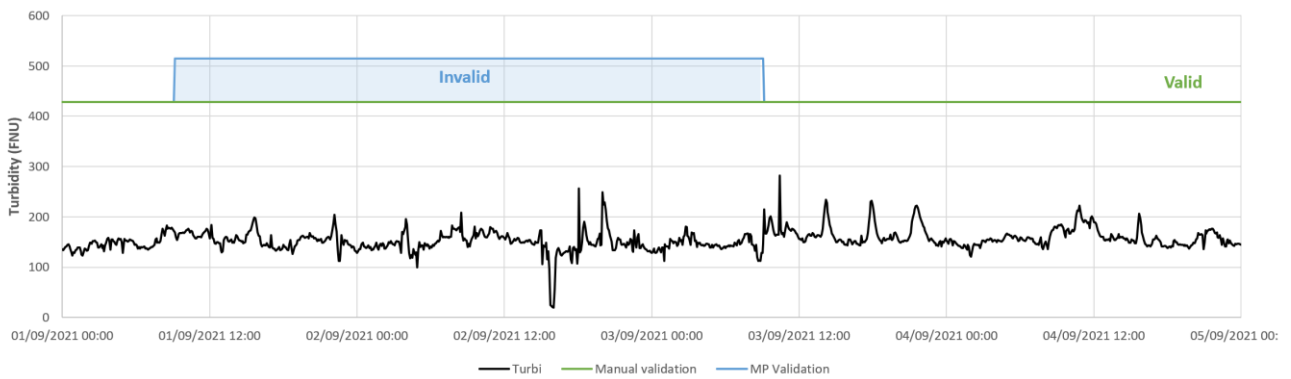
However, considering the two turbidities reveals that this sequence meets the redundancy criterion and was not even presented to the expert, suggesting that it is indeed valid despite its distinctive pattern. Specifically, upon examining the rainfall records (conductivity is also missing during this period), we note the occurrence of a rainfall event during the period of missing data (see **Figure 8-12**). This leads us to presume that the gap corresponds to the peak of the event, and consequently, the shift corresponds to the runoff effect typically observed after a rainy event, accompanied by a decrease in average turbidity. Thus, we come to realize that the Matrix Profile's monovariate approach may lead to misclassifications in the absence

of other **exogenous data** to support the validation process. This aspect will be thoroughly explored in **Section 8.5**, employing a multivariable approach.



**Figure 8-12: Rainfall history at Saint Malo - © Infoclimat**

Finally, we note a single period identified as a distinctive false positive by the algorithm around September 1<sup>st</sup>. **Figure 8-13** shows the reconstructed turbidity chronicle for this period. We can clearly see a particular turbidity pattern with fluctuations. Using domain knowledge, this behavior can be viewed as a fault, suggesting that it could be an **error on the part of the domain expert** who failed to identify this period as abnormal.



**Figure 8-13: Example of a false positive subsequence.**

**Conclusion**

To sum up, we observe that the MP model is sensitive to its hyperparameters. When utilizing a window size of 48 hours, the identification of shorter defects becomes challenging. Moreover, the delimitation of defects is rather imprecise. Consequently, exploring a multi-window approach to detect anomalies of various durations becomes intriguing. With a fixed anomaly rate, the algorithm is forced to prioritize the most prominent anomalies, which may not align with those identified by the expert. The disparity between MP and manual validation stems also from challenges in delimiting flaws and potential errors issued from the manual validation

process (filtering + expertise + aggregation). Another scenario arises when the expert validates a subsequence relying on additional exogenous data, such as redundancy. It is therefore interesting to explore matrix profiles as part of a multivariate approach.

### 8.3. How can we improve the results ?

Once we have adjusted the model's hyperparameters and evaluated its performance, obtaining an F1 score of 0.678 using reconstructed turbidity data, a 48-hour window and an anomaly rate of 9.5%, we note that the model shows promising results while facing challenges inherent to its principles. Consequently, this section aims to explore various avenues for improving performance. Firstly, we consider the potential benefits of combining results using raw data rather than going through reconstructed turbidity. Secondly, we examine the possibility of using an ensemble model to evaluate anomalies of different durations using multiple windows. Finally, we explore the integration of a pre-validation step to enhance the robustness of the model.

#### 8.3.1. Combining the results using raw data

The first strategy discussed for improving results is to merge the results obtained by applying Matrix Profile to the raw turbidity data. In practice, abnormal sequences in the reconstructed turbidity chronicle correspond to periods when both turbidimeters are faulty simultaneously. We have therefore combined the validation results of the two raw turbidities, identifying only common anomalies between the two sensors. The output target in this case is that of reconstructed turbidity. Since their optimal hyperparameters differ, we tested two distinct approaches:

- The first used the optimal hyperparameters for each data set independently (see [Table 22](#)).
- The second used adjusted hyperparameters: a window size of 48 hours (corresponding to the ideal window size with reconstructed turbidity) and a mean anomaly rate of 0.17.

The results are summarized in [Table 23](#).

**Table 23: Results of the combination of anomaly detection using raw data and selecting only common defects**

	Metrics	T1	T2	Combined
<b>Optimal hyperparameters</b>	<b>Precision</b>	0.646	0.555	0.667
	<b>Recall</b>	0.528	0.511	0.489
	<b>F1 score</b>	0.581	0.532	0.564
	<b>MCC</b>	0.486	0.404	0.520
<b>Adjusted hyperparameters</b>	<b>Precision</b>	0.599	0.524	0.609
	<b>Recall</b>	0.525	0.435	0.488
	<b>F1 score</b>	0.560	0.475	0.542
	<b>MCC</b>	0.452	0.343	0.488

Overall, the use of optimal hyperparameters leads to better results, in particular higher precision, i.e. the generation of fewer false alarms. However, in practice, this configuration is difficult to implement, as it requires hyperparameter tuning of each database, whereas in reality it involves the same type of sensor, in the same location and with the same configuration. Theoretically, there shouldn't be such a remarkable difference. What's more, comparing these results with those of reconstructed turbidity, which here has the same output target, we nonetheless observe inferior performance. Consequently, it is more interesting to use the reconstructed turbidity directly, but this requires the processing of the hardware redundancy using the filtering phase, as described in [Section 4.4.1](#).

### 8.3.2. Ensemble model

The aim of this section is to present the outcomes of an ensemble model employing various window sizes to evaluate anomalies of diverse durations. We categorize the ensemble models into two types: majority voting and minority voting. The primary objective of the former is to improve anomaly detection, thus enhancing precision, while the latter focuses on maximizing discord identification, ultimately increasing recall. To achieve this, three distinct window sizes are employed: 12 hours, 24 hours, and 48 hours. The latter two lengths are identified through grid search (see [Figure 8-3](#)) as optimal window sizes. The selection of the first window size is intended for detecting shorter anomalies. However, a deliberate decision is made to maintain a uniform anomaly rate across different window sizes to avoid introducing domain knowledge related to the ratio of short vs. long anomalies and to avoid specific tuning for each window size. The determination of the common anomaly ratio involves varying it between 5% and 20%, using a 0.5% increment, and identifying the optimal ratio based on the F1 score within the ensemble model results. The ROC curves in [Figure 8-14](#) validate the choice of the three



window sizes showing good AUC. For these various tests and in view of the conclusions drawn above, we use the reconstructed turbidity from Cottage (our test site).

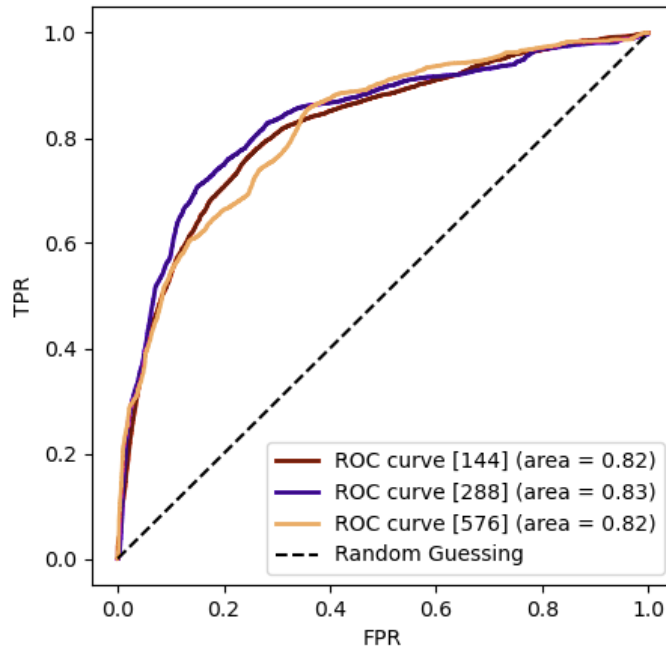
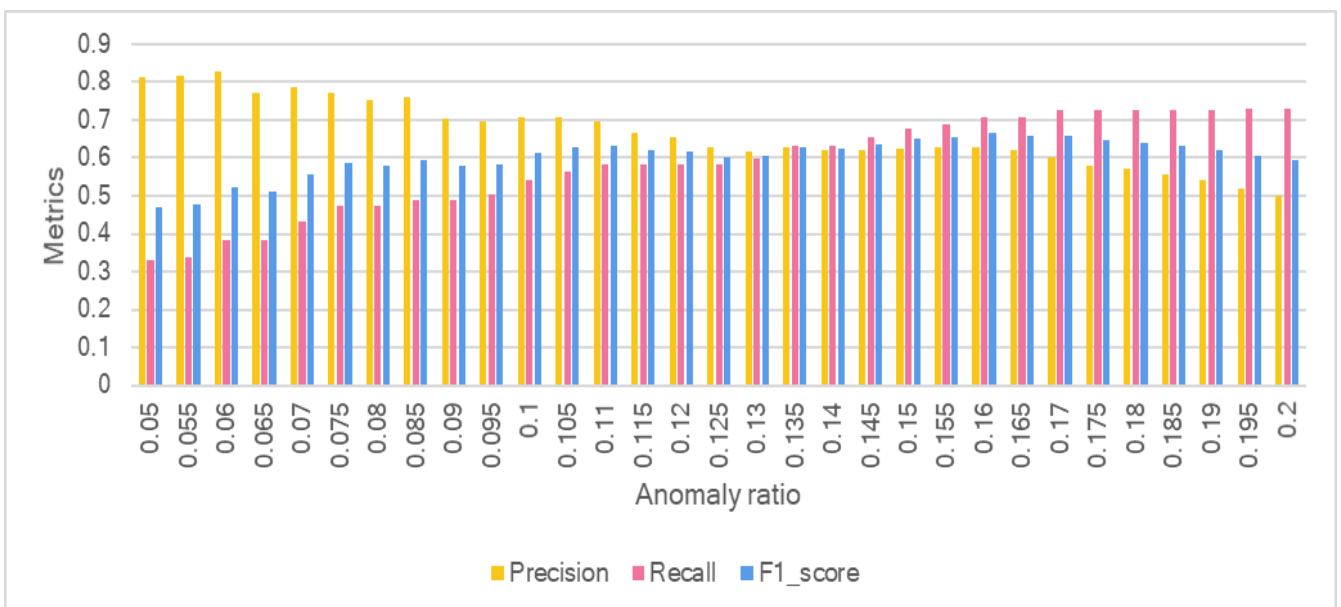


Figure 8-14: ROC Curves for the three window sizes used in ensemble model

### 8.3.2.1. Majority voting

The optimal anomaly ratio for majority voting is determined to be 0.16. Upon examining the sensitivity graph of the ensemble model to the anomaly rate (see Figure 8-15), we observe that the benefits of the majority vote are only apparent at low anomaly rates, where precision exceeds 80%. In other cases, the adoption of this approach does not contribute any additional value.



**Figure 8-15: Sensitivity of the ensemble model with majority vote to the anomaly ratio**

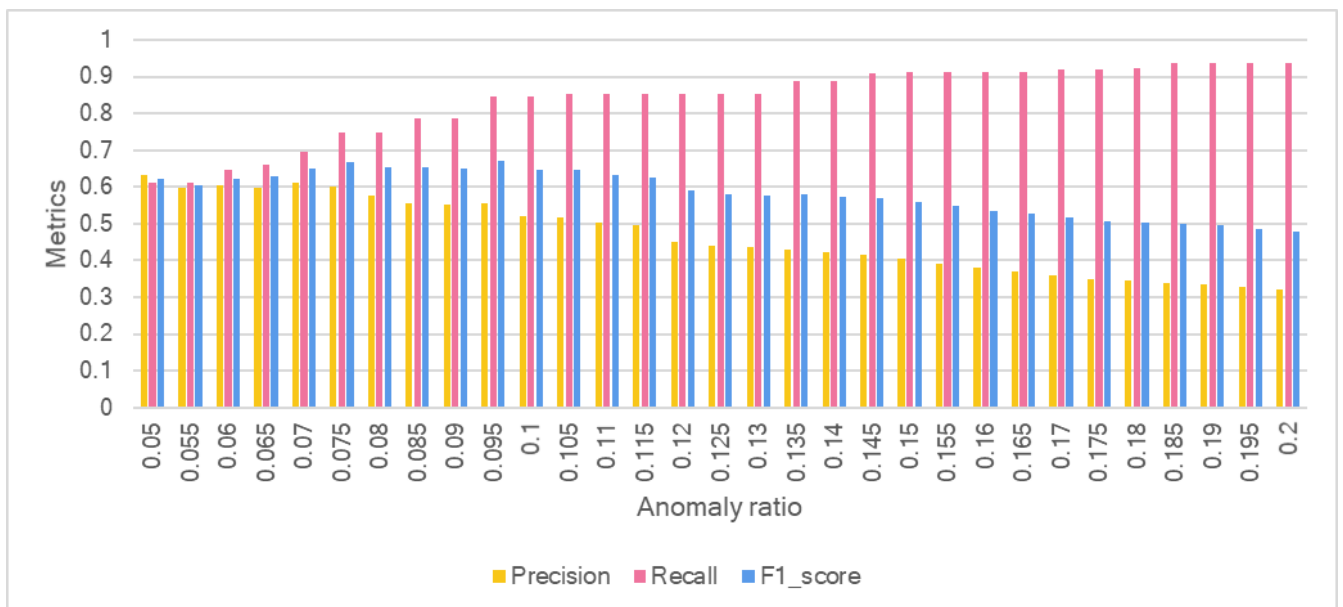
**Table 24** summarizes the results obtained. We find that the final result of the ensemble model is superior to that of the three sub-models, but inferior to that of our best model so far with a 48-hour window and an anomaly rate of 0.095. Indeed, through majority voting with three window sizes, a fine analysis of the results reveals that it is typically the 24-hour window that acts as the balancing factor for the other two, occurring approximately 75% of the time.

**Table 24: Majority voting Results**

	Precision	Recall	F1 score	MCC
<b>Window size = 12 hours</b>	0.465	0.630	0.535	0.464
<b>Window size = 24 hours</b>	0.464	0.619	0.531	0.458
<b>Window size = 48 hours</b>	0.557	0.766	0.645	0.595
<b>Ensemble Model</b>	0.628	0.706	0.664	0.614

### 8.3.2.2. Minority vote

**Figure 8-16** shows the sensitivity of the model to the anomaly rate. The optimum anomaly ratio in this case is 0.095. **Table 25** summarizes the results obtained.



**Figure 8-16: Sensitivity of the ensemble model with minority voting to the anomaly ratio**

The F1 score of the ensemble model falls short of the individual scores achieved by each window size independently. Nevertheless, the minority voting model effectively fulfills its role with a recall of 0.85, signifying the ability to identify 85% of abnormal days in the dataset.

However, this accomplishment comes at the expense of an increased number of false alerts. The precision stands at approximately 55%, indicating that for nearly every genuine discord identified, a false alert is generated. This model may prove valuable in scenarios involving the pre-selection of defects, followed by expert review. Despite the higher rate of false alerts, the precision remains noteworthy when compared to the manual validation filtering phase, which stands at approximately 39%. This postulate assumes that we prefer to miss 15% of anomalies (accept a recall = 85%) to save time in the expertise phase (precision = 55% vs. 39% previously).

**Table 25: Minority voting results**

	Precision	Recall	F1 score	MCC
<b>Window size = 12 hours</b>	0.539	0.450	0.491	0.427
<b>Window size = 24 hours</b>	0.591	0.478	0.529	0.472
<b>Window size = 48 hours</b>	0.729	0.633	0.678	0.637
<b>Ensemble Model</b>	0.554	0.846	0.670	0.630

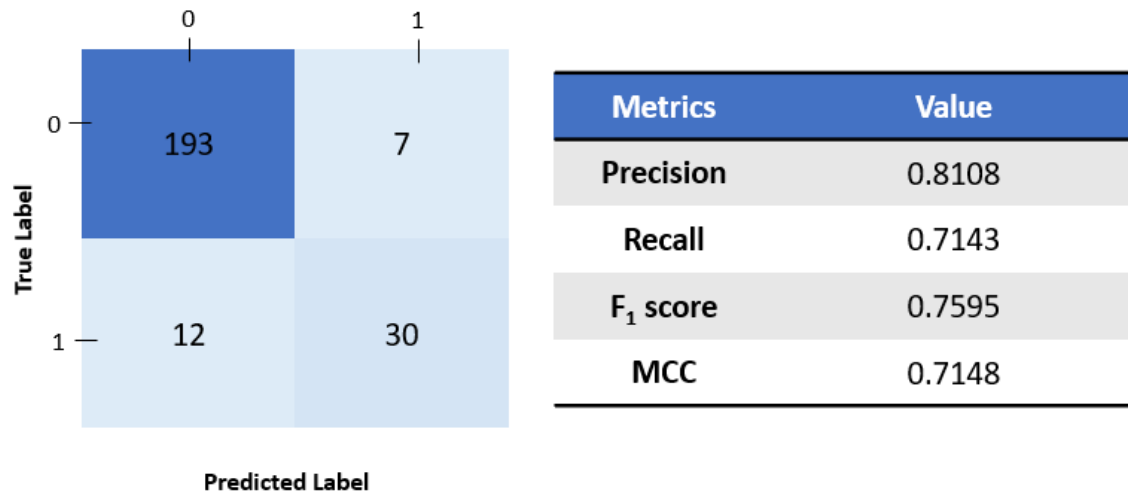
In conclusion, the aim of this section was to present the results of an ensemble model with different window sizes, so as to be able to identify anomalies of different durations. However, the results show that the ensemble model does not outperform the individual model with a single 48-hour window and an anomaly rate of 9.5%. So, while each ensemble approach may have a particular utility (maximizing precision or recall), overall performance remains inferior to that of a single model with an optimal window and a specific anomaly rate.

### 8.3.3. Pre-validation

In this section, we implement a first pre-validation step for the model. The aim is to mitigate the model's inherent bias, which discards the repetition of certain common defects. These faults can be easily identified using simple rules but may escape the model due to the principle of "fault uniqueness". This step automatically invalidates trivial anomalies such as missing data, data outside the range [1,1000], and blocking or saturation. Due to the sample size, these faults are not frequent in our case, and therefore, although this additional step is operationally necessary, it does not significantly improve the results: we move to an F1 score of 0.679 instead of 0.678.

What's more, up to now, we've assigned a label equivalent to that of the sequence for each time step that makes it up, with priority given to the invalid label due to overlapping sequences, this means that a single invalid sequence is enough to invalidate a time step. It's then interesting to transform the results to a daily scale, which is more practical and operational.

Thus, we re-sequence, a posteriori, the chronicle into non-overlapping daily sequences and invalidate a day if half of its constituent time steps are invalid, likewise for the results of the manual validation (reference) and those of the MP model. **Figure 8-17** summarizes the results. We observe an F1 score of 0.759 and an MCC of 0.715. These results show that this approach allows better capacity to identify anomalies in a wider, day-long context.



**Figure 8-17: Validation results at the daily sequence level using our best monovariate MP model**

#### 8.4. Generalization to other sites

To extend the applicability of MP model, this section aims to assess the algorithm's sensitivity to its hyperparameters across various measurement points. To achieve this, we conducted evaluations at three different sites characterized by distinct real anomaly rates (see **Table 26**).

**Table 26: Inherent anomaly ratio of different database**

	Cottage	Goutte	Découverte	Roosevelt
<b>Real anomaly ratio</b>	11%	26%	30%	3%

Here, we present heat maps depicting the behavior of each of the three sites based on different hyperparameters. Beginning with Goutte (see **Figure 8-18**), we note a similar pattern to Cottage, where there is a greater sensitivity to the window size compared to the anomaly rate. However, in this case, we observe a wide window length range around 24 hours. This range aligns with what was observed for Cottage, with similar performance metrics.



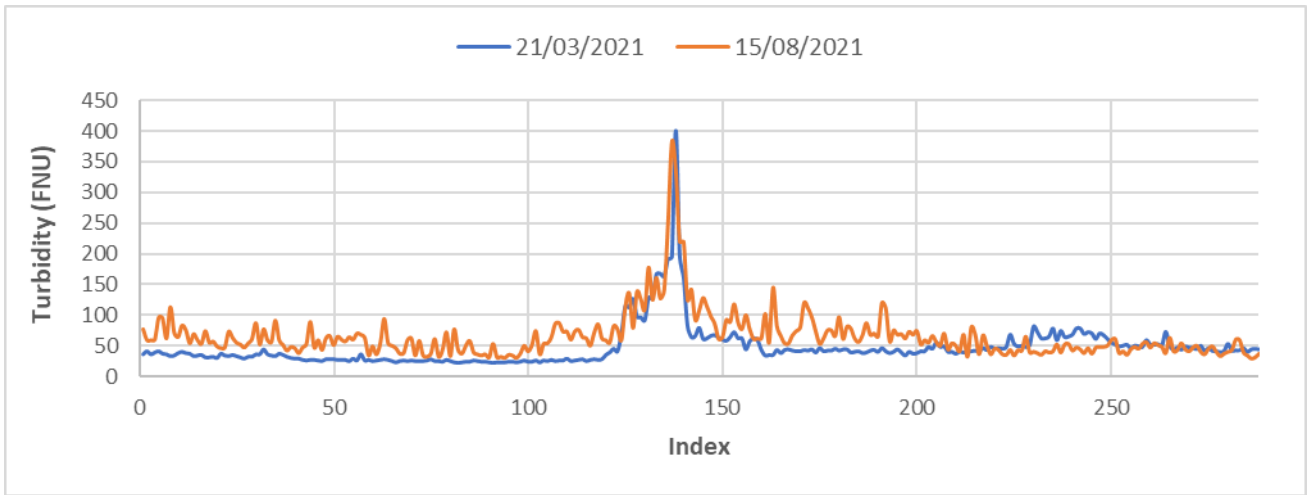


Figure 8-20: Example of two different abnormal sequences of 24-hours in Découverte dataset

The final test was carried out at Roosevelt (see Figure 8-21). It reveals a near-insensitivity to the window size, with a slight sensitivity to the anomaly rate. This observation aligns with the findings reported by [179], indicating that the matrix profile tends to be indifferent to the window size, while the choice of the anomaly rate remains a user-driven decision. Indeed, at Roosevelt, the anomaly rate test closely resembles those studied by [179], [200] and [237], leading us to draw similar conclusions.

Window size	Anomaly ratio																													
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18	0.19	0.2	0.21	0.22	0.23	0.24	0.25	0.26	0.27	0.28	0.29	0.3
2	0.620	0.617	0.594	0.563	0.543	0.526	0.518	0.490	0.466	0.437	0.419	0.396	0.375	0.356	0.343	0.331	0.320	0.311	0.299	0.287	0.276	0.266	0.260	0.253	0.245	0.239	0.233	0.225	0.221	0.216
4	0.620	0.685	0.558	0.618	0.576	0.536	0.511	0.481	0.459	0.439	0.422	0.398	0.385	0.379	0.361	0.351	0.336	0.322	0.308	0.297	0.287	0.276	0.267	0.258	0.250	0.242	0.234	0.228	0.221	0.215
6	0.620	0.649	0.672	0.640	0.599	0.582	0.540	0.501	0.487	0.472	0.443	0.418	0.397	0.377	0.358	0.343	0.328	0.322	0.310	0.298	0.286	0.277	0.267	0.258	0.250	0.242	0.234	0.228	0.222	0.216
8	0.645	0.713	0.686	0.666	0.642	0.593	0.571	0.549	0.514	0.484	0.456	0.429	0.407	0.388	0.370	0.351	0.337	0.323	0.311	0.298	0.287	0.277	0.268	0.258	0.250	0.243	0.235	0.228	0.222	0.216
10	0.620	0.689	0.773	0.729	0.721	0.670	0.620	0.575	0.536	0.502	0.477	0.450	0.426	0.405	0.385	0.372	0.355	0.340	0.326	0.313	0.316	0.304	0.294	0.284	0.274	0.265	0.259	0.251	0.243	0.236
12	0.620	0.658	0.697	0.626	0.606	0.555	0.520	0.482	0.449	0.446	0.419	0.422	0.430	0.406	0.389	0.371	0.356	0.344	0.330	0.318	0.306	0.295	0.285	0.277	0.268	0.260	0.252	0.245	0.238	
14	0.683	0.664	0.671	0.611	0.562	0.520	0.475	0.477	0.506	0.477	0.451	0.428	0.407	0.384	0.368	0.374	0.359	0.345	0.332	0.320	0.309	0.299	0.289	0.280	0.275	0.267	0.257	0.250	0.243	0.237
16	0.620	0.692	0.689	0.614	0.603	0.561	0.526	0.524	0.495	0.461	0.439	0.412	0.412	0.395	0.374	0.356	0.343	0.327	0.316	0.303	0.291	0.282	0.270	0.260	0.252	0.244	0.238	0.232	0.226	
18	0.675	0.746	0.697	0.654	0.637	0.589	0.547	0.511	0.470	0.443	0.448	0.425	0.404	0.380	0.384	0.368	0.353	0.335	0.322	0.311	0.300	0.290	0.278	0.269	0.261	0.251	0.243	0.237	0.231	0.225
20	0.620	0.707	0.703	0.723	0.677	0.619	0.570	0.529	0.493	0.493	0.510	0.481	0.455	0.432	0.410	0.391	0.375	0.359	0.344	0.330	0.322	0.310	0.299	0.288	0.279	0.271	0.261	0.253	0.247	0.241
22	0.620	0.715	0.738	0.681	0.673	0.631	0.578	0.532	0.507	0.472	0.441	0.449	0.425	0.408	0.386	0.367	0.355	0.338	0.323	0.313	0.300	0.288	0.281	0.270	0.265	0.255	0.245	0.239	0.233	0.227
24	0.620	0.674	0.763	0.673	0.623	0.591	0.537	0.601	0.555	0.516	0.498	0.465	0.449	0.422	0.397	0.386	0.366	0.347	0.339	0.323	0.315	0.302	0.289	0.287	0.277	0.272	0.262	0.253	0.246	0.240
26	0.699	0.730	0.693	0.634	0.636	0.591	0.618	0.581	0.531	0.503	0.477	0.455	0.426	0.407	0.390	0.375	0.367	0.347	0.334	0.324	0.313	0.298	0.289	0.281	0.272	0.261	0.254	0.248	0.241	0.241
28	0.704	0.766	0.690	0.628	0.654	0.605	0.618	0.643	0.604	0.580	0.547	0.517	0.492	0.457	0.436	0.419	0.402	0.386	0.371	0.357	0.344	0.333	0.322	0.312	0.302	0.293	0.281	0.273	0.273	0.265
30	0.620	0.697	0.620	0.657	0.688	0.631	0.584	0.543	0.507	0.475	0.491	0.463	0.441	0.419	0.399	0.390	0.372	0.356	0.342	0.328	0.316	0.305	0.294	0.284	0.275	0.276	0.267	0.259	0.252	0.245
32	0.620	0.799	0.734	0.658	0.596	0.569	0.605	0.561	0.522	0.487	0.471	0.446	0.420	0.398	0.377	0.359	0.350	0.335	0.320	0.307	0.299	0.293	0.282	0.272	0.262	0.254	0.249	0.242	0.234	0.234
34	0.721	0.757	0.695	0.658	0.594	0.541	0.518	0.558	0.534	0.496	0.489	0.452	0.425	0.412	0.390	0.369	0.351	0.343	0.327	0.313	0.306	0.293	0.286	0.280	0.270	0.260	0.251	0.246	0.247	0.239
36	0.620	0.674	0.763	0.673	0.623	0.591	0.537	0.601	0.555	0.516	0.498	0.465	0.449	0.422	0.397	0.386	0.366	0.347	0.339	0.323	0.315	0.302	0.289	0.287	0.277	0.272	0.262	0.253	0.246	0.240
38	0.729	0.767	0.717	0.634	0.620	0.654	0.623	0.570	0.546	0.507	0.488	0.455	0.440	0.413	0.400	0.377	0.367	0.347	0.330	0.322	0.307	0.300	0.291	0.285	0.275	0.271	0.261	0.256	0.246	0.242
40	0.738	0.679	0.723	0.677	0.634	0.603	0.653	0.622	0.567	0.543	0.524	0.484	0.467	0.436	0.422	0.396	0.384	0.369	0.358	0.340	0.331	0.322	0.307	0.300	0.287	0.280	0.269	0.263	0.253	0.248
42	0.743	0.682	0.727	0.684	0.642	0.667	0.650	0.589	0.562	0.541	0.499	0.481	0.463	0.431	0.417	0.391	0.379	0.368	0.347	0.343	0.325	0.317	0.309	0.295	0.288	0.281	0.272	0.266	0.256	0.250
44	0.748	0.684	0.788	0.681	0.658	0.688	0.614	0.583	0.558	0.508	0.487	0.468	0.432	0.417	0.402	0.376	0.364	0.353	0.332	0.323	0.314	0.302	0.298	0.291	0.283	0.270	0.264	0.258	0.247	0.242
46	0.745	0.679	0.784	0.756	0.683	0.643	0.576	0.548	0.523	0.499	0.461	0.443	0.426	0.410	0.382	0.369	0.357	0.346	0.325	0.321	0.312	0.295	0.288	0.280	0.274	0.261	0.255	0.250	0.244	0.244
48	0.756	0.687	0.795	0.734	0.716	0.690	0.649	0.612	0.580	0.527	0.503	0.482	0.462	0.443	0.412	0.397	0.383	0.370	0.358	0.336	0.331	0.321	0.312	0.295	0.288	0.280	0.273	0.267	0.255	0.248
50	0.754	0.682	0.795	0.731	0.760	0.728	0.639	0.603	0.570	0.541	0.515	0.491	0.453	0.435	0.418	0.402	0.389	0.375	0.350	0.339	0.328	0.318	0.309	0.300	0.288	0.280	0.273	0.266	0.260	0.256
52	0.764	0.691	0.806	0.841	0.776	0.743	0.694	0.651	0.579	0.549	0.523	0.499	0.476	0.455	0.436	0.425	0.408	0.379	0.365	0.354	0.343	0.332	0.322	0.312	0.303	0.287	0.279	0.272	0.265	0.259
54	0.764	0.686	0.807	0.762	0.796	0.738	0.688	0.645	0.606	0.572	0.542	0.515	0.491	0.468	0.429	0.412	0.397	0.382	0.370	0.357	0.345	0.333	0.323	0.313	0.304	0.295	0.279	0.271	0.264	0.258
56	0.762	0.683	0.807	0.738	0.679	0.648	0.603	0.564	0.530	0.555	0.526	0.499	0.475	0.453	0.433	0.415	0.398	0.382	0.369	0.355	0.344	0.332	0.321	0.311	0.301	0.292	0.276	0.268	0.261	0.254
58	0.761	0.679	0.867	0.736	0.676	0.625	0.582	0.560	0.587	0.553	0.524	0.497	0.473	0.450	0.430	0.412	0.395	0.379	0.365	0.352	0.340	0.328	0.317	0.307	0.297	0.288	0.280	0.272	0.264	0.257
60	0.620	0.761	0.677	0.803	0.731	0.670	0.619	0.638	0.596	0.573	0.541	0.511	0.484	0.460	0.438	0.419	0.401	0.384	0.369	0.354	0.341	0.330	0.318	0.308	0.298	0.288	0.280	0.271	0.264	0.256
62	0.620	0.757	0.672	0.798	0.724	0.663	0.611	0.629	0.587	0.564	0.532	0.503	0.476	0.452	0.430	0.410	0.392	0.376	0.361	0.347	0.334	0.322	0.311	0.301	0.291	0.282	0.273	0.265	0.257	0.250
64	0.620	0.754	0.667	0.796	0.721	0.659	0.609	0.629	0.586	0.550	0.550	0.530	0.501	0.474	0.450	0.428	0.408	0.390	0.373	0.358	0.344	0.344	0.331	0.320	0.309	0.298	0.289	0.280	0.271	0.263
66	0.620	0.751	0.662	0.794	0.718	0.655	0.663	0.615	0.615	0.572	0.549	0.515	0.487	0.461	0.437	0.415	0.415	0.396	0.378	0.362	0.347	0.333	0.321	0.310	0.310	0.299	0.289	0.280	0.271	0.262
68	0.620	0.748	0.657	0.786	0.665	0.717	0.717	0.658	0.608	0.567	0.544	0.510	0.481	0.481	0.455	0.431	0.410	0.390	0.373	0.357	0.357	0.342	0.328	0.317	0.305	0.294	0.284	0.284	0.274	0.266
70	0.620	0.745	0.653	0.780	0.523	0.523	0.550	0.614	0.567	0.528	0.493	0.493	0.462	0.436	0.422	0.400	0.380	0.362	0.346	0.331	0.317	0.304	0.304	0.293	0.282	0.272	0.263	0.254	0.254	0.244
72	0.620	0.742	0.648	0.755	0.517	0.517	0.595	0.614	0.567	0.528	0.528	0.492	0.462	0.434	0.410	0.410	0.398	0.378	0.360	0.344	0.344	0.329	0.315	0.302	0.290	0.290	0.279	0.269	0.260	0.252

Figure 8-21: Grid search results to identify best hyperparameters for Roosevelt dataset.

Table 27 provides a summary of these findings. In essence, MP proves effective for anomaly detection in datasets with a low anomaly prevalence (less than 5%) without the need for hyperparameter calibration. This aligns with Keogh's observations and is further validated by our experiments, resulting in satisfactory outcomes with F1 scores surpassing 80%. However,

in datasets featuring higher anomaly rates (between 5% and 25%), the model's performance becomes contingent on the window size. This parameter necessitates calibration based on domain knowledge regarding the intrinsic seasonality of the data and / or manual validation results. In the context of wastewater data, which typically exhibits 24-hour patterns, a similar value was identified at both Cottage and Goutte, with F1 scores ranging between 65% and 70%.

Nevertheless, when the anomaly rate exceeds approximately 25%, MP becomes inadequate and fails to effectively identify anomalies. This limitation arises from a fundamental conflict with the model's principle, which relies on the uniqueness of defects. One option for reducing the global rate of anomalies in the database is to divide the chronicle into sub-chronicles, and to use MP on each sub-chronicle independently. However, this requires each sub-chronicle to be representative enough to autonomously define the various normal operating modes, by having an adequate number of dry weather patterns and varied rainfall events. In our situation, given the length of the chronicle (7 months), this approach proves ineffective.

**Table 27: Matrix profile generalization results to other sites**

Dataset	Best hyperparameters		Metrics		
	Window size	Anomaly rate	Precision	Recall	F1 score
Cottage	48 hours	9.5%	0.729	0.633	0.678
Goutte	16 hours	17%	0.818	0.591	0.686
Découverte	24 hours	30%	0.486	0.567	0.523
Roosevelt	52 hours	4%	0.787	0.904	0.841

### 8.5. Multivariable anomaly detection

The first multivariate matrix profile tests are assessed using two variables in order to provide an interpretation of the results. To do so, we first use the turbidity redundancy by introducing the two raw turbidity chronicles as input. Then, we use the reconstructed turbidity combined with conductivity. The third test is conducted in a multivariate approach by integrating the three measured variables: namely redundant turbidity and conductivity. For this, we use the mstomp code developed by the UCR [205]. However, this code was initially introduced for motif identification and not for anomaly detection. Hence we adapted this code to identify the maximum distance.

### 8.5.1. Bivariate matrix profile

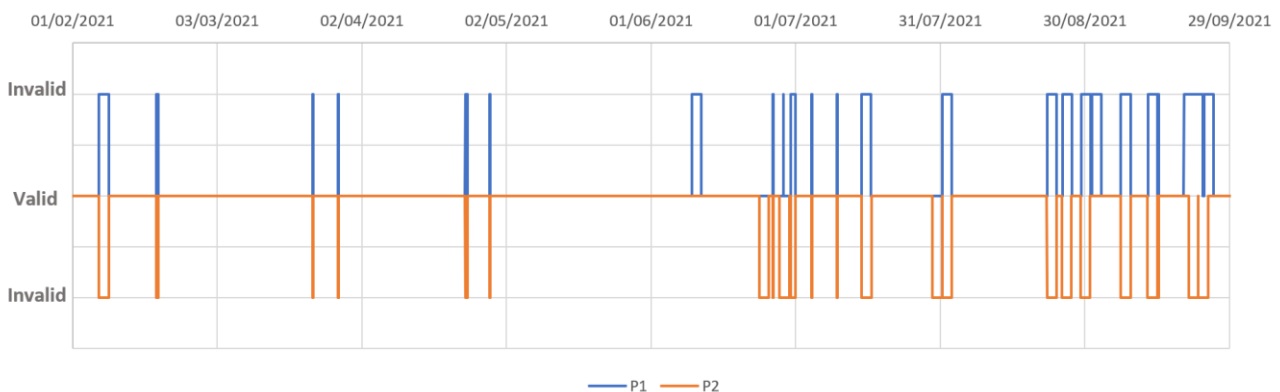
#### 8.5.1.1. Redundancy

The multivariate matrix profile yields two distinct profiles, denoted as P1 and P2 (see [Section 5.3.2 – Definition 11](#)). Consequently, the discords are identified, and various metrics are computed independently for each of these two profiles. We observe that the optimal hyperparameters differ between the two profiles. [Table 28](#) summarizes the results with the best window size for each profile and its respective anomaly ratio. In addition, we also determined the best anomaly rate for each profile by considering the optimal window size of the opposite profile.

**Table 28: Matrix profile results using redundancy – The best results for each profile are in bold**

Profile	Window size	Anomaly ratio	Precision	Recall	F1 score	MCC
P1	44	0.11	0.602	0.598	0.600	0.543
	<b>48</b>	<b>0.1</b>	<b>0.700</b>	<b>0.661</b>	<b>0.680</b>	<b>0.636</b>
P2	<b>44</b>	<b>0.08</b>	<b>0.807</b>	<b>0.606</b>	<b>0.692</b>	<b>0.664</b>
	48	0.07	0.792	0.538	0.640	0.614

[Figure 8-22](#) presents a comparison of the anomalies' identification outcomes using the two profiles with a window size of 48 hours and an anomaly rate of 0.1. Various pairs of anomalies and valid data are observed.



**Figure 8-22: Comparison of anomaly detection using P1 and P2 (w = 48 hours and k = 0.1)**

Indeed, in order to understand the distinction between the two profiles, it is necessary to review the calculation principle underlying each of them. [Figure 8-23](#) simplifies the process when two variables are involved. The calculation of P1 takes into account the "best" distance between



the two dimensions. In the context of anomaly detection (as we've programmed it), best translates into minimum. In other words, we're interested in the least penalizing / least flawed between the two variables, which is similar to what we do with manual validation, where we exclude the strongest variable and focus on the weakest.

For each multi-dimensional subsequence

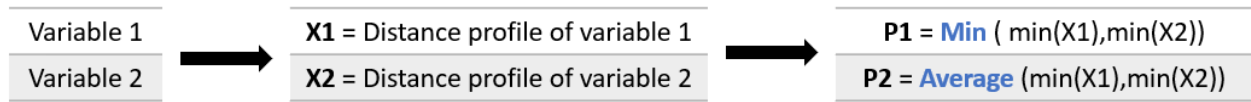


Figure 8-23: Calculation of P1 and P2 for anomaly detection

Table 29 allows to analyze the different cases and their interpretations.

Table 29: Interpretation of P1 and P2 results

		<i>P1</i>	
		Valid	Invalid
<i>P2</i>	Valid	Both turbidity data are valid	The sensor with the smallest distance is invalid with regard to its own chronicle => Although the average distance of the two turbidities is consistent with the rest of the chronicle, both turbidities remain deficient.
	Invalid	The turbidity chronicle with the smallest distance is valid => One of the two turbidimeters is deficient but the other one is reliable	The turbidity chronicle with the smallest distance is invalid => Both turbidimeters are deficient

Figure 8-24 illustrates a scenario where P1 identifies an anomaly while P2 validates it. This involves the sub-sequence spanning from the 9<sup>th</sup> to the 12<sup>th</sup> of June. It is evident that the turbidity data from the first sensor T1 exhibits abnormal behavior, moreover data from the second sensor T2 is quite noisy, leading to the invalidation of the data by P1. Furthermore, domain knowledge confirms the invalidity of this subsequence.

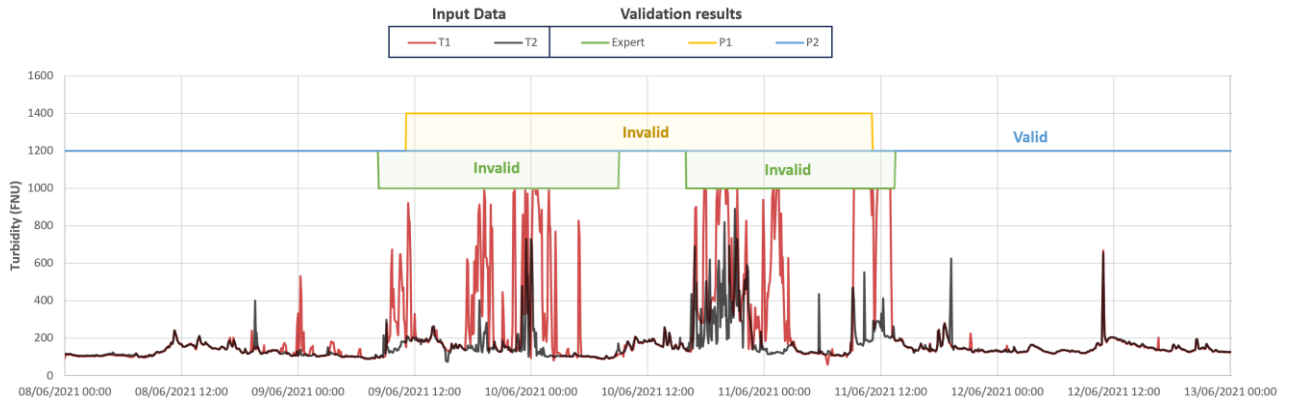


Figure 8-24: Data validation using the raw turbidities as input data

The aim is to determine which of the two profiles is more suitable for anomaly detection. From a practical standpoint, an anomaly of interest typically corresponds to a period when both turbidimeters exhibit faults, making P1 more relevant in such instances. Combining the outcomes of both profiles by selecting only the shared discords leads to a degradation in results, with an associated F1 score of 0.60. Figure 8-25 illustrates the results of anomaly detection, utilizing P1 for achieving the best F1 score (window size = 48 hours and anomaly ratio = 10%).

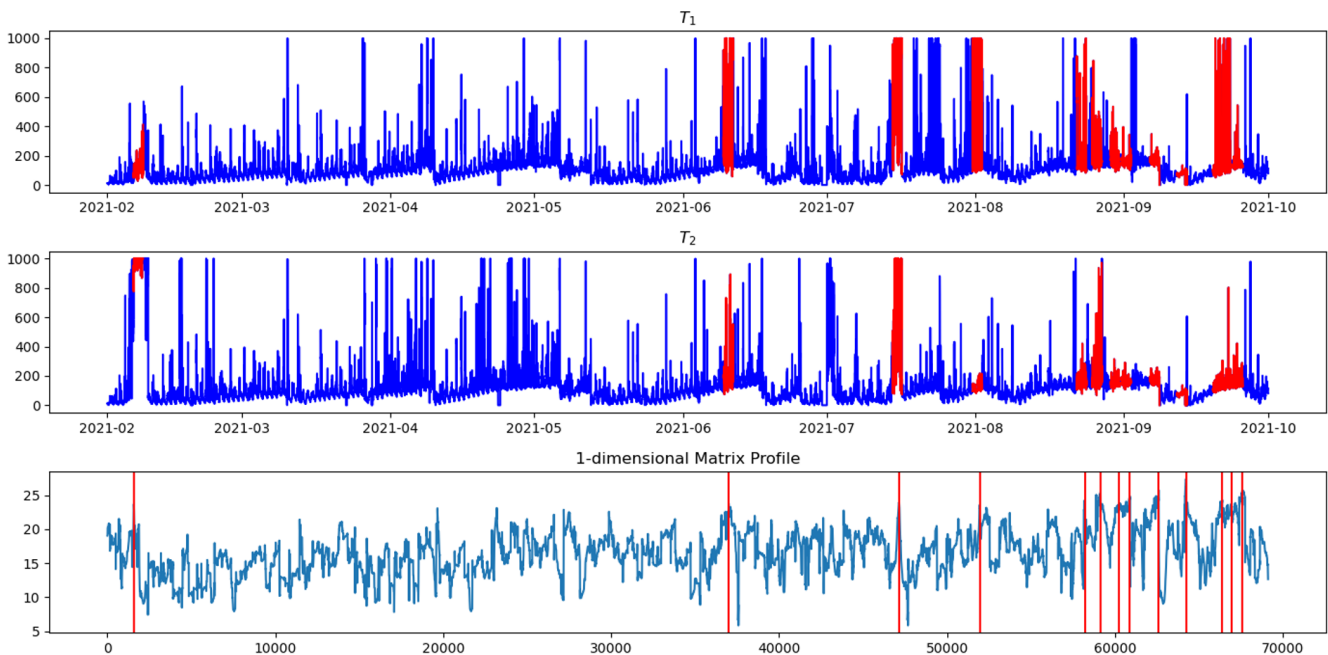
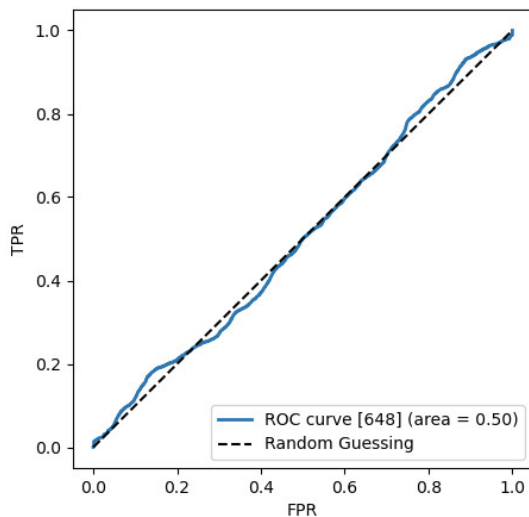


Figure 8-25: Anomaly detection using raw turbidity from the two sensors (T1 & T2) and P1

In conclusion, the use of the multivariate matrix profile yields two profiles, namely P1 and P2. The objective is to select the profile that better captures the dynamics of the variable and facilitates optimal anomaly detection. Considering the usage context and the characteristics of

the two profiles, P1 aligns more closely with our requirements. P1 consistently prioritizes the sensor with the smallest distance, and if this sensor is deemed invalid, it implies the invalidity of the other sensor as well. This configuration, where both turbidimeters malfunction, is particularly pertinent to our anomaly detection needs. In terms of quantitative results, anomaly detection using P1 achieves an F1 score of 0.68. Remarkably, this score matches the performance obtained through the monovariate matrix profile approach using the reconstructed turbidity. Therefore, the bivariate approach reinforces the importance of redundancy, allowing us to bypass the preprocessing step for turbidity reconstruction by directly inputting the two raw chronicles into the model, while maintaining the same level of performance.

### 8.5.1.2. Turbidity and conductivity



**Figure 8-26: ROC curve for bivariate matrix profile P1 using reconstructed turbidity and conductivity**

After conducting the analysis above, our discussion centers on the results of the bivariate matrix profile using reconstructed turbidity and conductivity. Indeed, given the reliability of the conductivity data, P1 consistently chooses it as the data with the smallest distance. In the presence of a fixed anomaly rate, discords are identified in the conductivity, but these do not align with any meaningful physical reality. The likelihood that these identified subsequences correspond to a turbidity anomaly is essentially random, leading to results akin to random guessing (see [Figure 8-26](#)). Consequently, we will focus our analysis on P2, assuming that since conductivity is generally reliable, its matrix profile is not very variable, and

consequently an anomaly detected by P2 (average of conductivity and turbidity matrix profiles) would be more likely linked to a defect in the turbidity data. On the other hand, an apparent anomaly in conductivity would inhibit a synchronous anomaly in turbidity, since both could be due to a physical phenomenon, such as rain.

[Table 30](#) provides a summary of the obtained results. Notably, even with optimized hyperparameters, the F1 score and the MCC remain below 50%. We conclude that conductivity does not contribute value to the detection of anomalies in turbidity within a bivariate approach.

Table 30: Matrix Profile Results on turbidity and conductivity

Profile	Window size	Anomaly ratio	Precision	Recall	F1 score	MCC
P1	54	0.11	0.459	0.445	0.452	0.376

### 8.5.2. Multivariate matrix profile

The aim of this section is to assess the multivariate MP using the raw data from the three on-site measurements: namely, the two turbidimeters and the conductometer. Table 31 provides a summary of the results obtained with a window size of 24 hours. This choice is based on the tuning of P3 (see Section 5.3.2 – Definition 11), the only profile with an F1 score exceeding 0.5 among the three calculated (see Figure 8-27).

Table 31: Results of multivariate matrix profile for a window size of 24 hours

	Window size	Anomaly ratio	Precision	Recall	F1 score	MCC
P1	24 hours	0.135	0.304	0.355	0.328	0.224
P2		0.15	0.426	0.553	0.481	0.401
P3		0.105	0.679	0.639	0.658	0.611

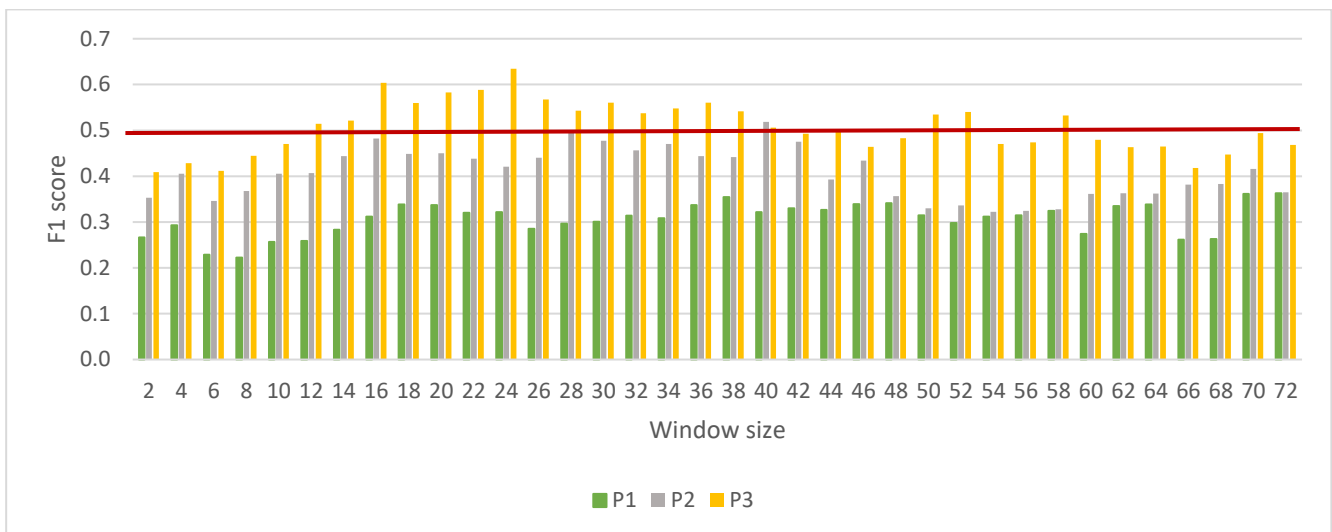


Figure 8-27: F1 score using multivariate matrix profile depending on the window size

Similar to the bivariate approach and following the same tuning strategy, the following conclusions emerge:

- Optimal hyperparameters differ across various profiles.
- Profile P1 exhibits performance akin to random guessing, as indicated by its ROC curve.
- The performance metrics, including F1 score and MCC, for both P1 and P2 are below 0.5.

When contrasting these findings with the bivariate validation using both turbidimeters, a degradation in results is evident. We infer that the inclusion of conductivity data disrupts the model and does not contribute any added value.

### 8.5.3. How can we improve the results ?

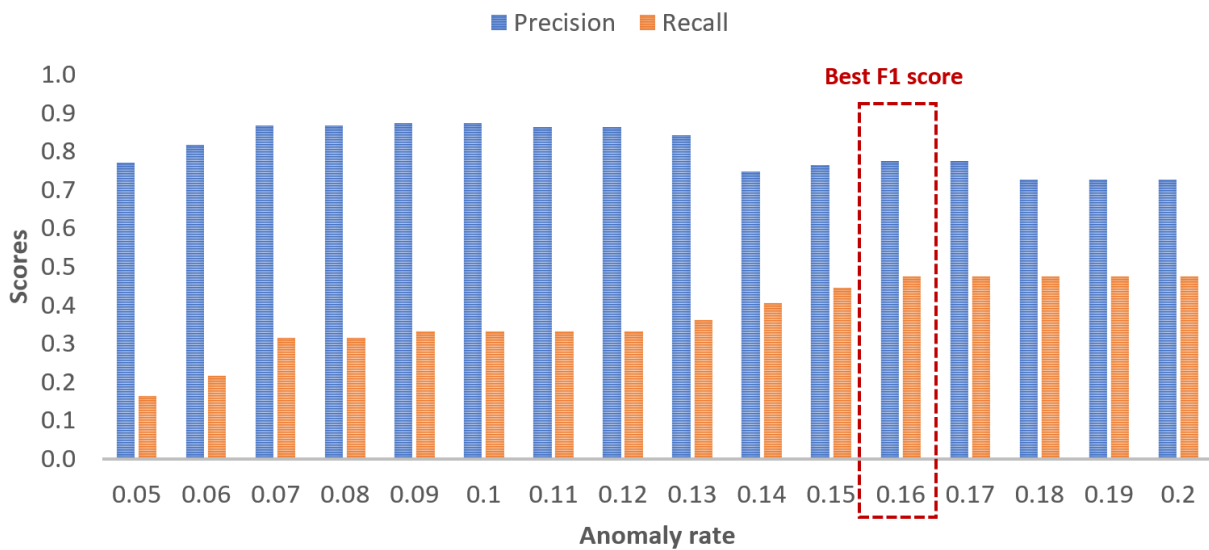
#### 8.5.3.1. Adjusted multivariate strategy

The purpose of this section is to draw an analogy between the manual validation process carried out by an expert and the validation process established by the MP model. Given that defects of the turbidity sensors consistently moves towards higher values, an anomaly results in a substantial distance. As elucidated in [Section 4.4.1](#), the expert initiates validation by comparing the two turbidities. If the anomaly threshold is triggered, the expert invalidates the stronger sequence and assesses the one with lower turbidity. This step aligns with the construction of the P1 profile using the bivariate matrix profile with redundancy. P1 compares the distance of the two subsequences based on their respective chronicles and selects the one with the lowest distance. If the weaker turbidimeter does not follow its expected dynamics, indicating an anomaly, the expert scrutinizes the pattern of the two turbidimeters to evaluate their inter-coherence. Naturally, if both measure approximately the same thing, the average distance is lower than if one of them deviates. If both deviate, the distance regarding the rest of the dataset remains significant. This step corresponds to the construction of the P2 profile, which calculates the distance of each turbidimeter from itself, averages the two distances, and compares the result to the rest of the dataset. Finally, if the two turbidimeters lack inter-coherence, meaning P2 is invalid, the expert examines the conductivity to determine if it explains the turbidity pattern. To achieve this, we integrate the three variables by considering the profile P3, which exhibits the best score in a multivariate approach. Consequently, an anomaly identified by the expert leads to an invalid subsequence according to the three profiles: P1 and P2 from a bivariate validation using the two turbidimeters and P3 from a multivariate validation. This scenario corresponds to a unanimous vote. The hyperparameter tuning is carried out using the same approach as in [Section 8.2](#). [Table 32](#) summarizes the results obtained using this multivariate approach, adjusted in accordance with the domain expert's methodology. The overall performance of the model is weaker than the baseline bivariate model. Although the optimum hyperparameters differ, they still fall within a similar range.

Table 32: Results of adjusted multivariate matrix profile

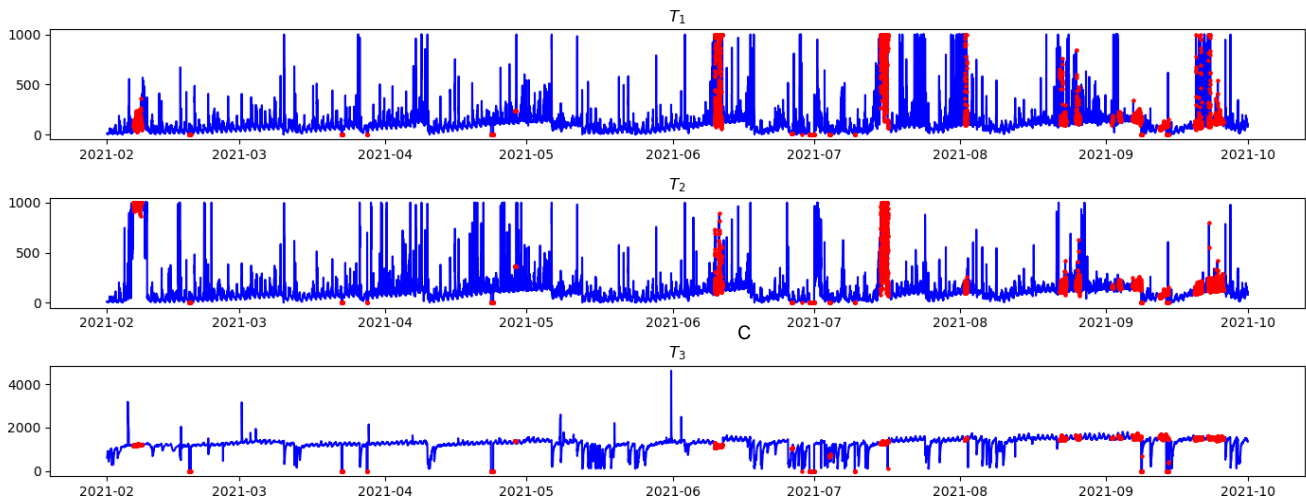
Window size	Anomaly ratio	Precision	Recall	F1 score	MCC
44 hours	0.16	0.776	0.474	0.589	0.566

The advantage of this approach is that it can provide good precision. **Figure 8-28** shows the evolution of the precision according to the anomaly ratio for a window size of 44 hours. Precision can be above 80%, i.e., very few false alarms are generated, and the anomalies identified are accurate. However, this is done at the expense of false negatives with an extremely low recall.



**Figure 8-28: Precision and recall using adjusted multivariate matrix profile depending on the anomaly ratio**

The anomaly rate of the final database is 7%, which is lower than the real rate of 12.5%, hence the substantial number of false negatives. **Figure 8-29** shows the identified anomalies using this approach.



**Figure 8-29: Anomalies such as identified on the three chronicles using the adjusted multivariate matrix profile**

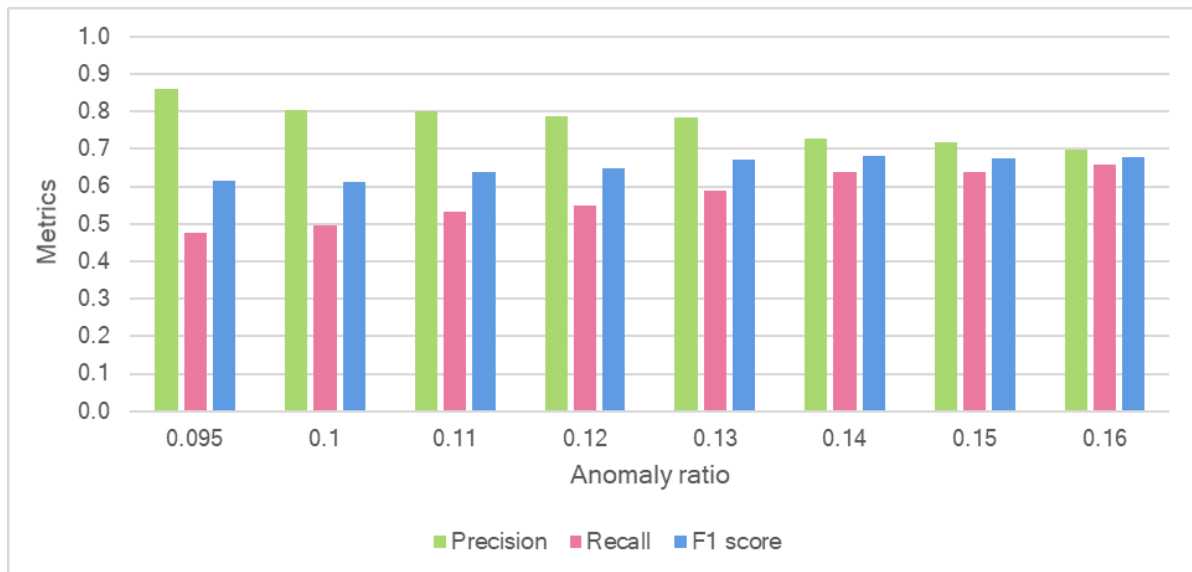
This approach could be enhanced by determining the turbidimeter selected during the construction of P1 for each subsequence. Consequently, we could pinpoint the turbidimeter exhibiting the least deviation. Instead of employing P3 to integrate conductivity, we could leverage P2, taking into account both conductivity and the identified turbidimeter. The retrieval of these elements could be facilitated by identifying the subspaces of the matrix profile (see [Section 5.3.2 – Definition 12](#)) using the stumpy implementation. Regrettably, due to time constraints, this avenue has not been explored.

### 8.5.3.2. Global model combining ensemble model and multivariate approach

This section explores the combination of the multivariate approach (see [Section 8.5.2](#)) with the ensemble approach (see [Section 8.3.2](#)), aiming to harness their respective advantages. The first approach enhances anomaly detection by mitigating false positives, while the second, utilizing minority voting, minimizes false negatives, leading to improved anomaly identification. To achieve this, we integrate the multivariate approach discussed earlier with a multi-window analysis employing window sizes identical to those defined in [Section 8.3.2](#), namely 12, 24, and 48 hours.

Given that the window sizes are predetermined, it becomes essential to determine the anomaly rate. However, the optimal anomaly rate differs for the two approaches: the first necessitates an anomaly rate of 0.16, while the second requires a rate of 0.095. Consequently, we conducted an evaluation across varying anomaly rates between these two limits. [Figure 8-30](#) illustrates the results, indicating that the optimal anomaly rate in this case is approximately 0.14, with an associated F1 score of 0.68. As the anomaly rate increases, the advantages of

the adjusted multivariate model diminish, and precision declines, reaching standard scores akin to those calculated previously.



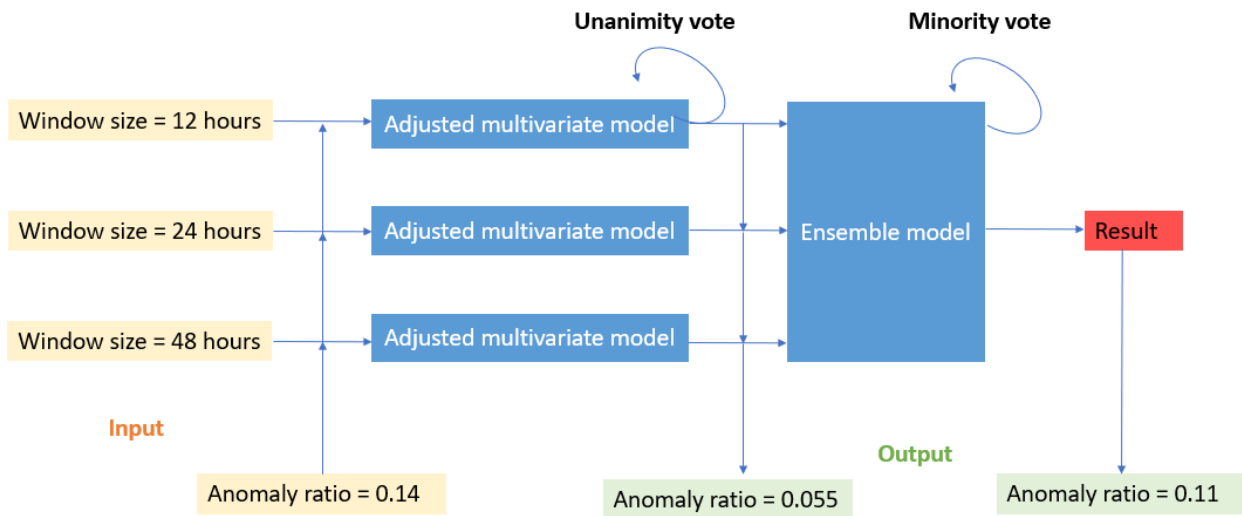
**Figure 8-30: Metrics using global model depending on the anomaly ratio**

In terms of quantitative metrics, the number of false positives is at most equal to the sum of the false positives identified by each sub-model. As for false negatives, they are maximized by considering the minimum count of false negatives across the sub-models. While this observation might suggest the possibility of identifying an absolute optimum between the two approaches, the number of true positives exhibits a broader range of variability, spanning from the minimum among different sub-models to their sum.

Finding a compromise between the two approaches proves challenging, especially considering that the ensemble model relies on a minority vote applied to the multivariable model, which, in turn, is based on a unanimous vote. These two approaches are in competition, making it difficult to reach an optimum while retaining the advantages of each method. When setting an anomaly rate 'x' for the calculation, the results of each adjusted model for a given window size inherently have a lower anomaly rate due to the unanimity vote. Applying an ensemble approach with a minority vote to these results tends to increase this intrinsic anomaly rate of



the model. An example using an anomaly rate of 14% is illustrated in [Figure 8-31](#). Since the final result aligns closely with the actual rate, there is no need to further increase 'k'.



**Figure 8-31: Illustration of the global model process and its impact on the anomaly rate – Note that the real anomaly ratio is of 12.5%**

Therefore, this approach does not yield an improved final result compared to a monovariate approach using the reconstructed turbidity or a bivariate approach with both turbidimeters. Considering the definition of the ensemble model and the adjusted multivariate model, achieving an optimum that effectively combines their advantages proves challenging.

## 8.6. Synthesis of Chapter 8

The aim of this section is the application and evaluation of the Matrix Profile for anomaly detection, on turbidity data collected at Cottage. The sensitivity of the model to input data is explored through various preprocessing steps, such as missing values imputation, downsampling, and data smoothing. The results indicate that imputing missing data with zeros performs better than other techniques, and downsampling degrades the model's performance. Additionally, data smoothing negatively affects the model's ability to identify anomalies, particularly when noise is one of the features of interest. The section highlights the model's sensitivity to different input data sources, suggesting better performance with reconstructed turbidity compared to raw data.

Hyperparameter tuning tests are conducted using a grid search to find the best combination of window size and anomaly ratio. The results show that optimal hyperparameters vary depending on the input dataset. For example, the best window size for T1 is 44 hours, while for the other, it is 28 hours. The anomaly rates also differ. The section delves into the challenges of defect delineation and biases in data validation, highlighting discrepancies

between the model and manual validation results. Issues such as anomalies merging during anomalies aggregation phase and the impact of a fixed window size on anomaly detection are discussed. The introduction of an ensemble model, combining different window sizes to detect anomalies of varying durations, is explored using minority and majority vote, revealing however that this approach does not match the performance of an individual model. Finally, a pre-validation step is implemented to correct model biases linked to the repetition of common faults, leading to a modest improvement in results when evaluated on a daily scale.

Furthermore, the generalization of the MP model to different measurement sites is explored to assess its sensitivity to hyperparameters. Evaluations are conducted at three distinct sites, each characterized by varying real anomaly rates. The results indicate that MP is effective for datasets with low anomaly prevalence without hyperparameter calibration. However, for datasets with higher anomaly rates, calibration becomes crucial, and beyond approximately 25%, MP becomes inadequate.

Finally, the focus shifts to multivariable anomaly detection using the Matrix Profile model, exploring bivariate and multivariate approaches with different combinations of turbidity and conductivity data. The analysis reveals that P1, prioritizing the sensor with the smallest distance, aligns more closely with anomaly detection needs, achieving an F1 score of 0.68. The bivariate approach with turbidity and conductivity, however, yields suboptimal results, indicating that the inclusion of conductivity disrupts the model. The multivariate matrix profile, integrating raw data from three variables, exhibits lower results. An adjusted multivariate strategy aligning with domain expert methodology is proposed, but the overall performance is weaker than the baseline bivariate model. A global model combining the multivariate and ensemble approaches is explored, aiming to leverage their respective advantages, but achieving an optimum proves challenging.

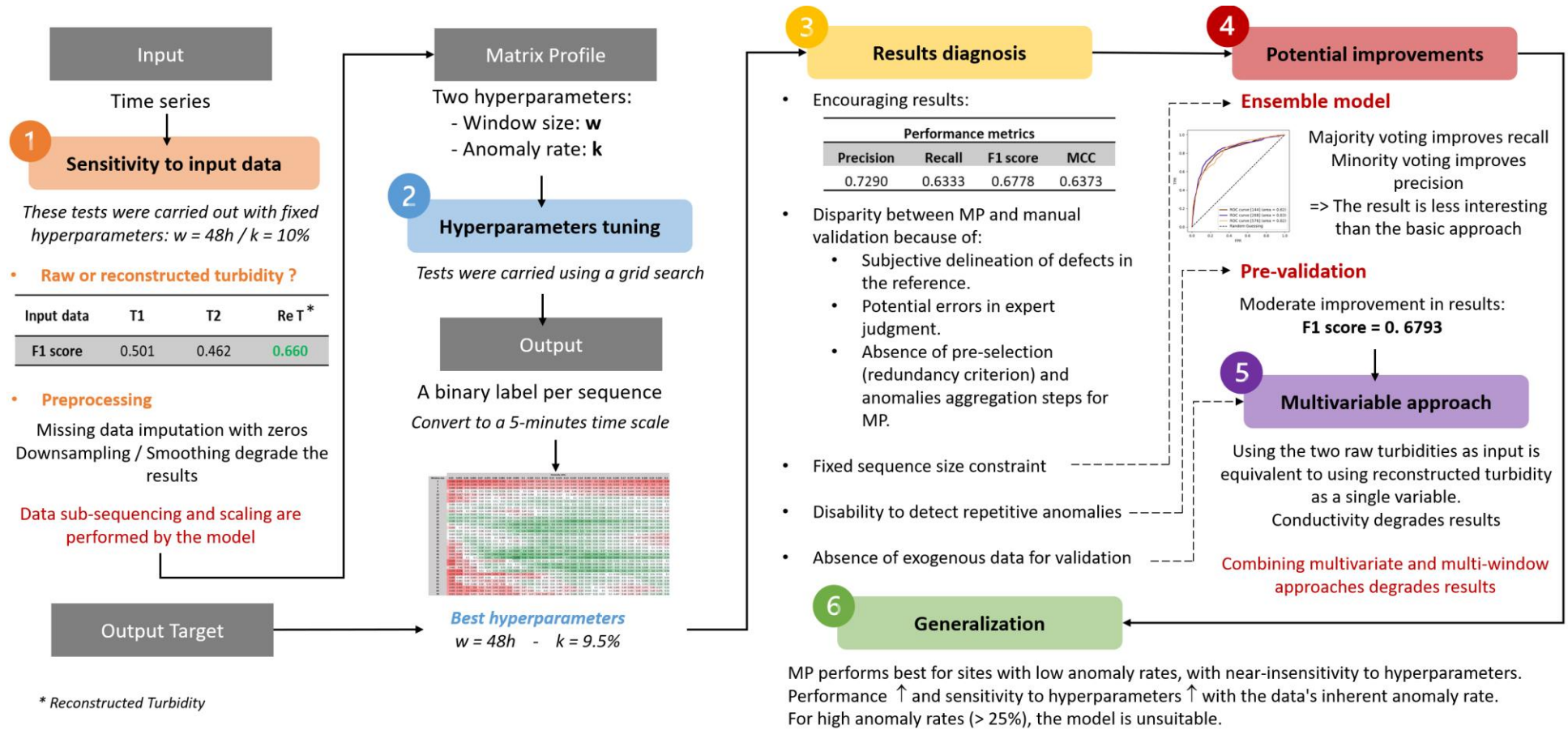


Figure 8-32: Overview of Matrix Profile tests and results for anomaly detection using turbidity data

## Chapter 9. ResNet Evaluation

---

In this section focusing on the ResNet model, we delve into its performance, considering various factors that impact its efficacy. The architectural foundation of ResNet, as detailed in [Section 5.4.3](#), remains a fixed reference point throughout our evaluation. The dataset used is derived from the turbidity data at Cottage, spanning from February 2021 to January 2022. However, the database has undergone augmentation during testing, extending its coverage until July 2022.

Our investigation starts with an exploration of the model's sensitivity to input data (see [Section 9.1](#)). Within this domain, preprocessing techniques, data enhancement strategies, and the inherent characteristics of input data are analyzed. Then, in [Section 9.2](#), we inspect the impact of hyperparameters on ResNet performance, considering factors such as the input window size and the transformation of probabilities into sequence labels.

To enhance the model's validation capabilities, we consider various approaches (see [Section 9.3](#)), such as pre-validation approaches, alongside with multiclass classification strategy. Additionally, we explore the potential benefits of predicting anomaly rates per sequence, contributing to an investigation of ResNet's capabilities for another, still related, task.

Expanding our evaluation scope, we evaluate the generalization of the best ResNet identified from the previous tests across different sites. Direct evaluations, cross-site training methodologies, and site-specific tuning, with a particular focus on the Roosevelt site, offer insights into the model's adaptability to diverse environmental contexts (see [Section 9.4](#)). Finally, our evaluation extends to multivariable anomaly detection, exploring ResNet's ability to identify anomalies using multiple variables (see [Section 9.5](#)).

### 9.1. Sensitivity to input data

This section aims to assess the sensitivity of the ResNet model to input data and its preprocessing. As a supervised model, it takes as input both sequential measurement data and its classification. A key question then arises: how long should the measurement sequences be to optimize the model's performance? Furthermore, how should we pre-process these sequences in addition to the basic steps mentioned in [Section 8.1.1](#), including resampling and imputation of missing data? Furthermore, given that the manual validation provides labels on a time-step scale, it is essential to understand how to move from this scale to a per-sequence label. Sensitivity tests will be carried out in this respect. Hence, the question of the threshold is to be considered in this transition to ensure accurate sequence classification.

### 9.1.1. Preprocessing

Input data pre-processing for the ResNet model is based mainly on standard practices and analogies with tests carried out using Matrix Profile. 24-hour sliding sequences are used, in line with what was used for MP, and which demonstrated good results reconciling numerical performance and operability. Given the importance of data scaling for deep learning models [238], a standardization similar to [133] is applied to the input data.

The tests focus on window size and stride. Initially, very high scores, with an F1 score in excess of 90%, were obtained when evaluating the model without prior tuning of the hyperparameters. In fact, the 5-Folds cross-validation learning strategy introduced a risk of overfitting, as similar sequences were found in different training and test sets. To overcome this bias, the stride parameter was adjusted to avoid overfitting, set at half the window size. This configuration reconciles data improvement with avoidance of overfitting. Tests on window size will be detailed in [Section 9.2.1](#).

As for the label assigned to each sequence, it has been decided that a sequence is considered invalid as soon as half of its time steps are invalid, whether consecutively or not, in analogy with the approach used for Matrix Profile evaluation. However, this parameter will also be the subject of sensitivity tests in [Section 9.2.2.2](#).

### 9.1.2. Data Enhancement

When we examine the database of Cottage turbidity over the whole year, we find an anomaly rate of 8%. In other words, the number of valid sequences is 11 times greater than the number of invalid sequences. Given that training the model involves injecting samples from both classes, and that our main objective is to detect anomalies, i.e. to focus on invalid data, the imbalance between the classes poses a problem. This is why the use of data enhancement, aimed at balancing the two classes, becomes relevant.

The results of this evaluation are summarized in [Table 33](#). These different methods are evaluated independently (a cross in the table means that this strategy was not used for the test in question), then simultaneously by varying their parameters (see [Section 5.4.4.1](#)). This evaluation is based on the use of 2-hour time sequences with a half-window step, using reconstructed turbidity.

Thus, we observe that an excessive oversampling of invalid samples has a negative impact on the results. This degradation is explained by the fact that the model may overlearn from these extra samples, compromising its ability to correctly assimilate the valid samples. The best ratio is at a moderate increase, from 8% initially to 25%, which leads to better

performance, notably an F1 score of 0.56. Similarly, the addition of noise should be measured, with a scaling factor of 0.05 producing promising results reaching 0.53. However, excessive noise addition leads to degraded results. The cost-sensitive approach stands out by improving results over the status quo, albeit modestly, with a maximum F1 score of 0.49. On the other hand, the combination of different approaches leads to a deterioration in results. Thus, we conclude that the model benefits from an increase in the number of invalid samples for better learning, but the addition of noise and/or sample duplicates presents limitations, underlining the need for a balanced approach to data enhancement.

**Table 33: Results of ResNet model for different enhancement approaches**

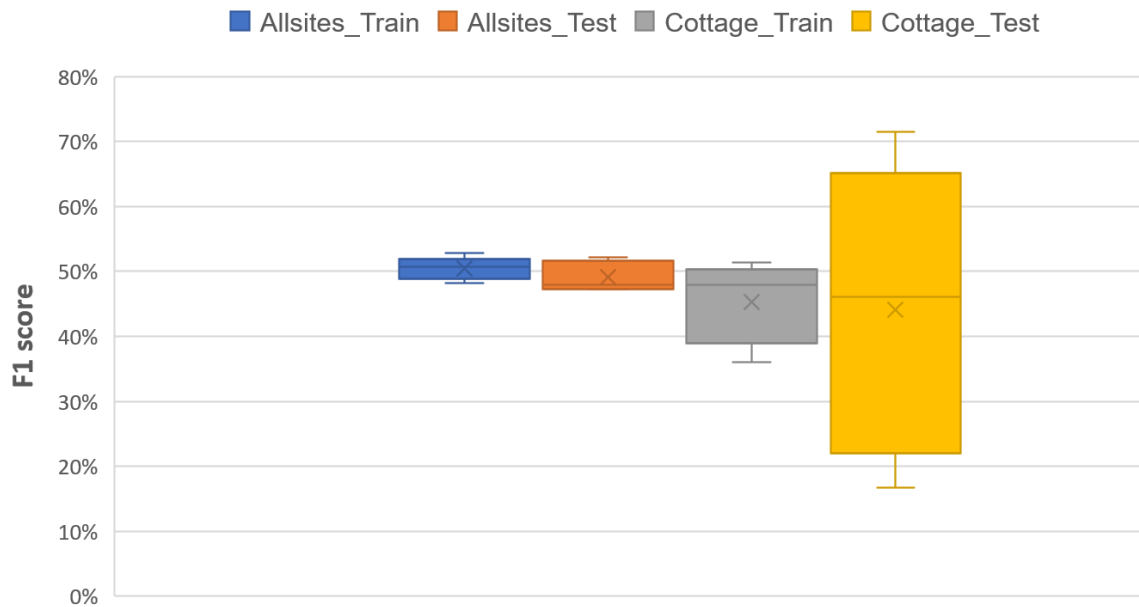
Database	Enhancement strategies			Performance metrics		
	Oversampling	Noise scale	Cost-sensitive	Precision	Recall	F1 score
Raw	X	X	X	0.842	0.123	0.215
With enhancement strategies	X	X	1:2	0.706	0.369	0.485
	X	X	1:5	0.405	0.616	0.489
	X	X	1:10	0.233	0.806	0.362
	Minority	X	X	0.270	0.897	0.415
	0.75	X	X	0.313	0.826	0.454
	0.5	X	X	0.417	0.678	0.517
	0.25	X	X	0.648	0.490	0.558
	X	0.05	X	0.697	0.428	0.530
	X	0.1	X	0.765	0.338	0.469
	Minority	0.05	X	0.262	0.783	0.392
	0.25	0.05	X	0.428	0.428	0.508
	X	0.05	1:8	0.215	0.857	0.343
	X	0.05	1:5	0.239	0.869	0.375
	Minority	X	1:2	0.198	0.950	0.328
	0.25	X	1:5	0.231	0.862	0.364
0.25	0.05	1:5	0.200	0.837	0.323	

To remedy the above problem, one potential approach is to take advantage of available data from other sites, thus eliminating the need to artificially create data. **Table 34** illustrates model performance when using reconstructed 24-hour turbidity sequences. Hence, there is no improvement in performance over the exclusive use of Cottage data.

**Table 34: Results of ResNet model using different sites as input**

	Cottage	All sites
% Anomalies	8.22%	19.71%
Precision	91.67%	90.70%
Recall	36.67%	36.19%
F1 score	52.38%	51.74%
MCC	56.03%	52.14%

Although, **Figure 9-1** shows that there is a reduction in standard deviation, indicating a stabilization of inter-fold results. This stability is directly attributable to the substantial increase in data volume, underlining the positive impact of using data from multiple sites for model training.



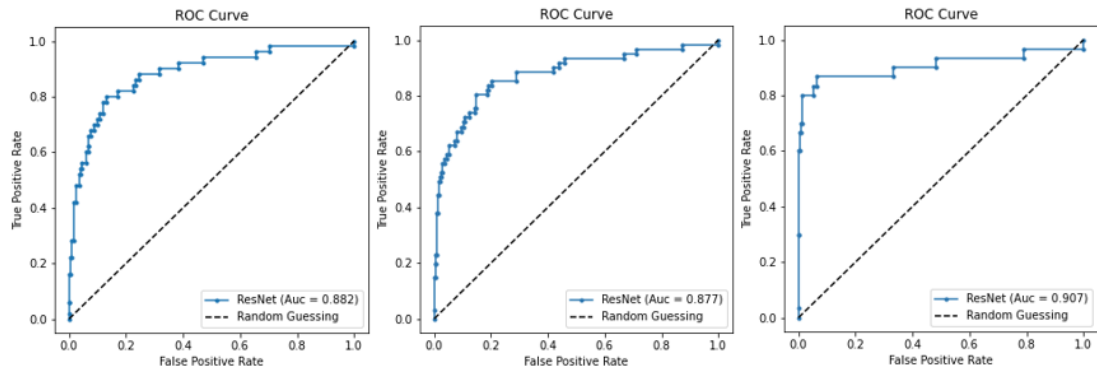
**Figure 9-1: Variation of the F1 score between different folds depending on the input database**

⇒ To sum up, this paragraph highlights the challenges of class imbalance in the Cottage turbidity database, where anomaly represents only 8% of the data. Given the primary objective of detecting anomalies, this asymmetry poses a problem when training the model, which requires a balance between the two classes. To remedy this, several data improvement approaches are explored, such as oversampling, white noise addition, and cost-sensitive learning. Oversampling is preferred by increasing the minority class (invalid sequences), and results show that a moderate ratio (from 8% to 25%) leads to interesting performance, illustrated by an F1 score of 0.56. Another approach is to use data available from other sites, thus eliminating the need to artificially create data. The results show that this method does not lead to a significant improvement in performance over the exclusive use of Cottage data, apart from the stabilization of inter-fold performance.

### 9.1.3. Input data

The objective of this test is to assess the model's responsiveness to various inputs, specifically raw data and reconstructed turbidity. For this purpose, we establish the pre-processing framework, which includes standardization and a 24-hour window size, with a half-sequence stride. The sequence label corresponds to the majority, meaning it is deemed invalid only if

more than half of its constituent time points are invalid. Training is carried out utilizing the ResNet model described in [Section 5.4.3](#) and using a 5-Fold cross-validation approach. Model evaluation, on the other hand, occurs on the complete database, considering non over-lapping sequences of 24 hours. The results obtained are summarized in [Figure 9-2](#).



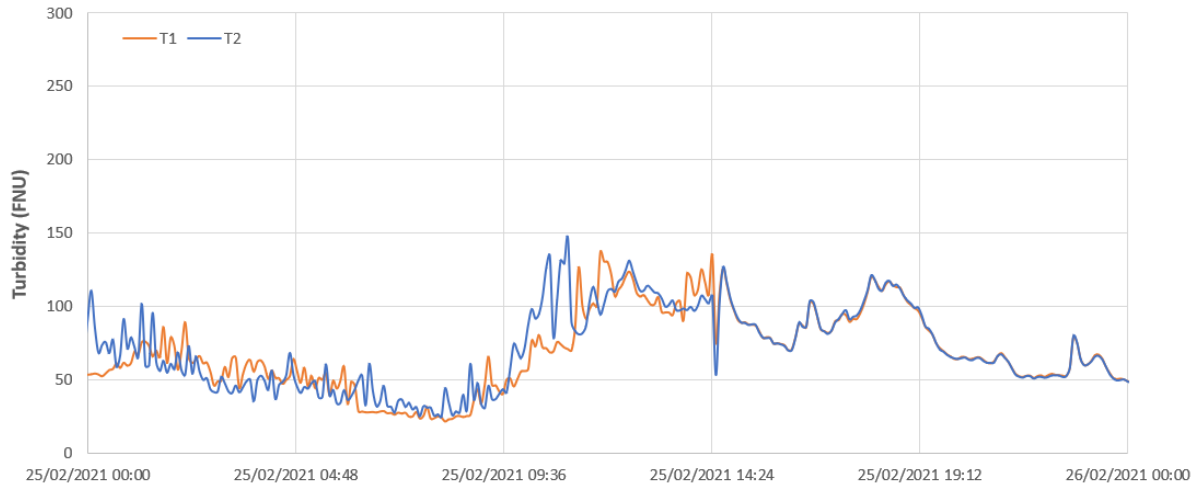
	Raw T1	Raw T2	Reconstructed T
<b>Precision</b>	0.7826	0.7805	0.9167
<b>Recall</b>	0.3600	0.5246	0.3667
<b>F1 score</b>	0.4932	0.6275	0.5238
<b>MCC</b>	0.4869	0.5790	0.5602

**Figure 9-2: Results for different input data using the ResNet model**

It can be seen that there are few notable distinctions between the use of the various inputs. A slight positive elevation is noted for Raw T2 considering the F1 score, with a 10% improvement over the reconstructed turbidity. However, this variation is minimal when the MCC is considered. This observation is directly associated with the disparity in anomaly rates in the database: 17% for T2 versus 8% for reconstructed turbidity.

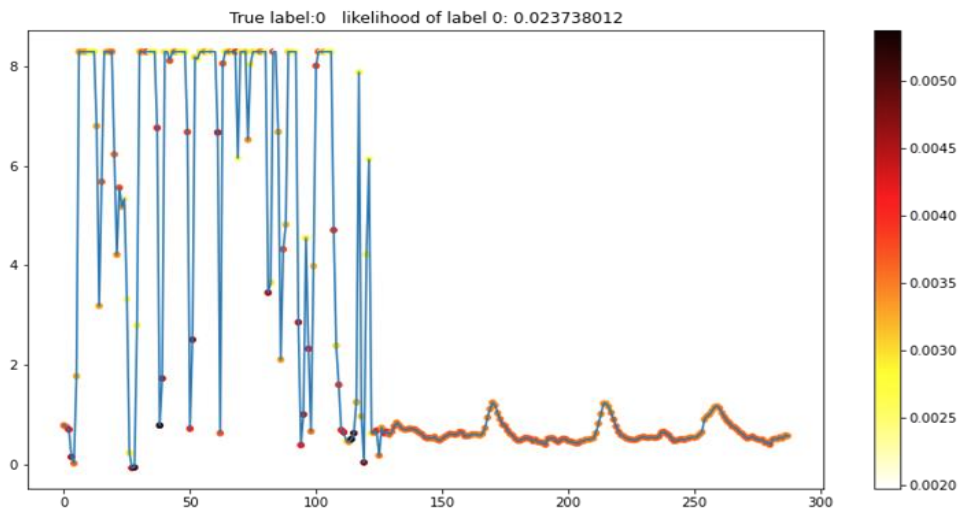
The highest precision, observed for Reconstructed T (0.9167), suggests that the model has an increased likelihood to minimize false positive predictions when applied to reconstructed data. This tendency can be attributed to the definition of the consistency threshold by redundancy, which automatically invalidates the highest turbidity (see [Equation 3](#)). It should be noted that the presence of high turbidity does not necessarily imply outlier data, as it could share the same structure as lower turbidity, which would be assessed by an expert. This scenario, illustrated in [Figure 9-3](#), generates errors in the Raw T2 reference database, resulting in false negatives. These false negatives are eliminated when reconstructed turbidity is taken into account. Indeed, in this context, the invalidation of a sequence is justified by its structure and/or context, which contributes to greater precision in classification.





**Figure 9-3: Example of False Negative, biased by the filtering criterion**

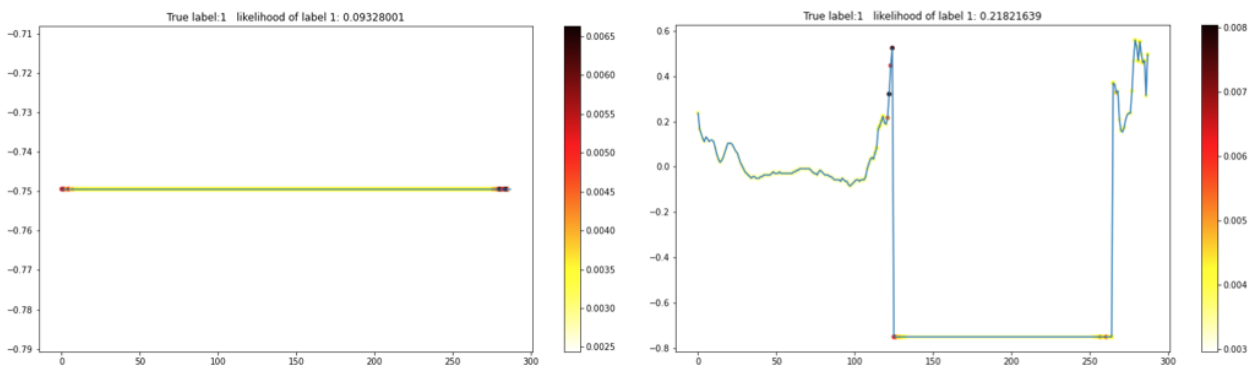
Across the various inputs, false-positive errors also stem from the threshold that determines whether a sequence is considered invalid. **Figure 9-4** illustrates an obviously invalid sequence with significant saturation. Of this 24-hour temporal sequence (i.e. 288-time steps), the expert invalidated only 122-time steps (less than half), thus assigning a valid label to the sequence. On the other hand, the model invalidates this sequence with a probability of 98% (indicating a strong belief in its invalidity). This is a threshold issue for assigning a label to a sequence. This question will be the subject of further sensitivity tests. Nevertheless, these results highlight scenarios where the model outperforms the baseline.



**Figure 9-4: Example of False Positive sequence with saturation**

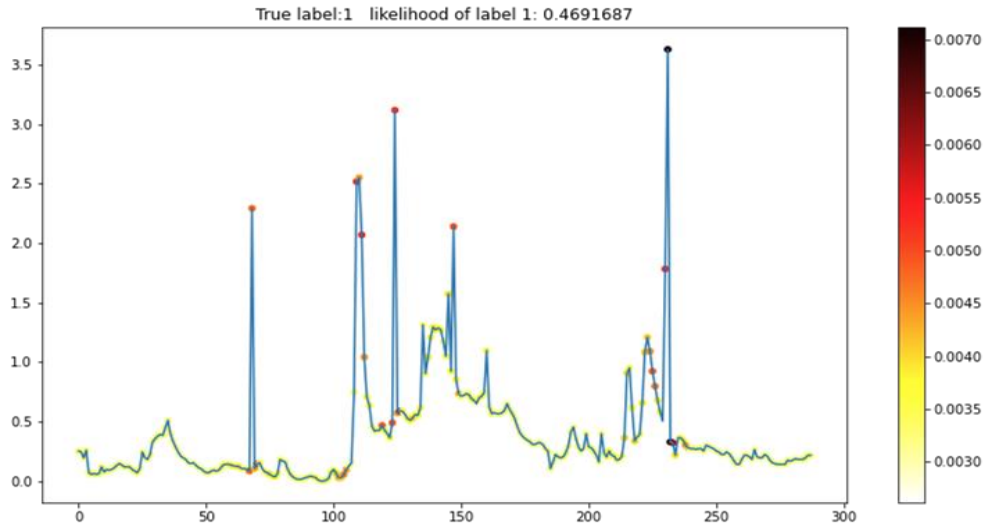
In general, recall shows a lower value, indicating a problem in defect identification, with a significant proportion of omitted anomalies. This is partly due to the difficulty of identifying trivial anomalies associated with null sequences. Indeed, sequences characterized by null data over the entire 24-hour time span are very rare, with only one day missing from the entire database

(30<sup>th</sup> June 2021). Consequently, if a null sequence is present in the test data, the network may find it difficult to generalize correctly, having not been exposed to such sequences during training. On the other hand, if exposed to such sequences during training, the multiplication of null values by the neural network weights will result in a null output for each neuron. This can lead to a significant loss of information, as the weights associated with such sequences will not contribute to updating the network parameters during training. This scenario also applies to sequences with a lot of missing data, which are replaced by zeros (see [Figure 9-5](#)). Detection of such anomalies can benefit from specific pre-validation processes to identify such sequences appropriately.



**Figure 9-5: Example of False Positive sequence with null data**

Problems related to the classification threshold at the output of the ResNet neural network can also be observed. Indeed, the output of the network consists of a probability of belonging to each respective class. The class considered predominant is the one whose probability exceeds 0.5, and it is assigned to the input sequence. However, in certain borderline situations, we may end up with probabilities close to 50-50, forcing the model to make a decision based on a tiny difference in probability in favor of one class or the other (see [Figure 9-6](#)). The model's classification threshold will therefore be the subject of subsequent sensitivity tests in order to adjust this decision boundary and improve the stability of classifications in such circumstances.



**Figure 9-6: Example of False Negative sequence**

Consequently, analysis of the results requires careful consideration of the reference database. The supervised nature of model learning implies the assimilation of certain biases present in the reference, which may lead to the reproduction of these biases in subsequent evaluations. Alternatively, the model may contradict the reference according to what it has learned elsewhere (from other sequences), thus generating false positives or false negatives likely to bias performance evaluation. It therefore becomes essential to conduct sensitivity tests on the classification thresholds applied to the reference and the model when analyzing input sequences, in order to better understand and mitigate these potential sources of bias in the results.

Various tests have been implemented to neutralize the bias associated with the redundancy criterion and the classification threshold applied during manual validation. In order to bypass these issues, attention is focused exclusively on sequences that are either 100% valid or 100% invalid, derived from the raw data of T1 and T2. **Figure 9-7** illustrates the distribution of sequences according to their anomaly rate. Thus, in this configuration (Test A in **Table 35**), training is carried out on a set of sequences totaling a maximum of (675 + 96).

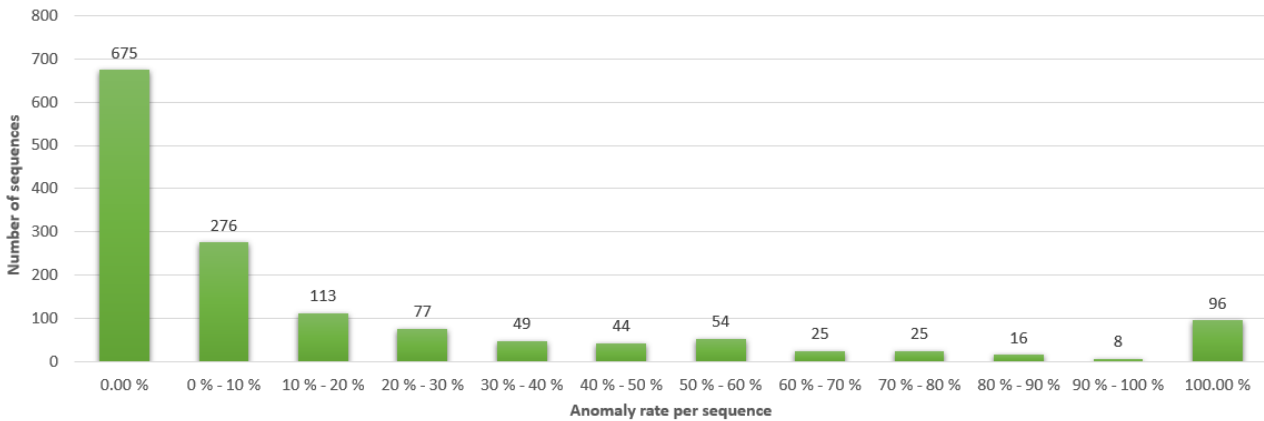


Figure 9-7: Number of sequences according to their inherent anomaly rate

In parallel, further tests are undertaken focusing on the sequences involved in the expertise process. The first test (Test B in Table 35) consists of including only those sequences where the expert has invalidated the submitted sequence, bearing in mind that the other sequence is already invalid according to the redundancy criterion. This avoids the bias introduced by the filtering phase. The second test (Test C in Table 35) examines cases of discrepancy between the labels of T1 and T2, giving priority in these situations to the invalidated turbidity (the highest), thus strengthening the anomaly database. The last test (Test D in Table 35), in the event of discrepancy, only takes into account sequences validated by the expert, thus avoiding false sequences that are invalid simply because they don't meet the redundancy criterion. Table 35 summarizes these different tests, specifying the number of valid and invalid sequences considered in each scenario.

Table 35: List of tests applied to input data

Test A	Test B	Test C	Test D
<ul style="list-style-type: none"> <li>• Only 100% valid and invalid sequences</li> <li>• Number of valid: 675</li> <li>• Number of invalid: 96</li> </ul>	<ul style="list-style-type: none"> <li>• Only common 100% valid and invalid sequences</li> <li>• Number of valid: 428</li> <li>• Number of invalid: 78</li> </ul>	<ul style="list-style-type: none"> <li>• 100% valid and invalid sequences considering the invalid sequence if disagreement</li> <li>• Number of valid: 428</li> <li>• Number of invalid: 92</li> </ul>	<ul style="list-style-type: none"> <li>• 100% valid and invalid sequences considering the valid sequence if disagreement</li> <li>• Number of valid: 442</li> <li>• Number of invalid: 78</li> </ul>

Table 36 summarizes the results obtained using the different inputs. The results show a slight superiority of Test C. Using an analysis of variance (ANOVA), as described in Appendix K, the aim is to assess the significance of the differences observed between the various tests. With a p-value significantly above the 0.05 threshold, we conclude that there is no statistically significant difference between the means of the tests considered. Nevertheless, it is important

to note that overall, our results are superior to those previously obtained with the full data set. Therefore, for subsequent steps, we adopt the conditions of test A to evaluate our ResNet model. This evaluation is carried out using the raw data from T1 and T2 to ensure a sufficiently representative database size. It is crucial to take into account the potential bias introduced by this approach, underlining the need for a thorough analysis of the results at each stage of the test.

**Table 36: Training results of ResNet using different inputs and a 5-folds cross validation**

	Test A	Test B	Test C	Test D
Average Train F1 score	77.8%	76.9%	83.8%	78.8%
Std Train F1 score	4.4%	5.3%	4.7%	4.5%
Average Test F1 score	76.2%	74.6%	78.2%	77.4%
Std Test F1 score	6.5%	10.6%	3.2%	10.0%

⇒ The objective of this section is to assess the responsiveness of ResNet model to various inputs. The first results indicate a few notable distinctions between the various inputs: raw and reconstructed turbidity, which are mainly associated with the disparity in anomaly rates in the database. The precision is higher for Reconstructed T, suggesting the model's increased likelihood to minimize false positive predictions when applied to these data. Sensitivity tests focusing on sequences with 100% validity or 100% invalidity are conducted to neutralize biases. The overall superiority of the results compared to the full dataset is acknowledged. For subsequent steps, the conditions of Test A are adopted to evaluate the ResNet model, using raw data from T1 and T2.

## 9.2. Hyperparameters tuning

During these tests, the database was extended to July 2022. With this in mind, tests were undertaken to corroborate the conclusions of the input data sensitivity phase. For the subsequent tests, we consider the following prerequisites: raw data from T1 and T2 over the 18-month period, including only 100% valid or invalid sequences, standardized over a 24-hour window with a stride equal to half the window.

By default, it has been decided not to alter the architecture of the ResNet model. Thus, the adjustment of hyperparameters will not concern the architecture itself, but rather other elements that impact the results. The first aspect taken into consideration concerns the use of a 24-hour sequence, hence the question of whether this is the optimum window size remains open. Furthermore, the second point of interest is the classification threshold. Although the

objective of the output is rather a sequence label, the model generates, for each sequence, a probability of belonging to each class. The final class is therefore determined by a probability in excess of 0.5. The relevance of this threshold is called into question.

### 9.2.1. Sensitivity to the input window size

The aim of this test is to evaluate the model's response to variable input sequence sizes. The results are summarized in **Figure 9-8**, while the window size refers to number of hours. The metrics Train\_F1 and Val\_F1 respectively represent the average F1 score between the different folds on the training and validation data, using the 5-folds cross-validation approach. Then, for each window size, the best model among the 5 folds is saved and evaluated on the complete data set (and not just on sequences with exclusively valid or invalid labels, used for training). The results of this evaluation are referred to as Eval\_F1.

Analyzing the graph of results, we can see that optimum performance is achieved with a window size of 36 hours, giving an F1 score of 0.65. However, for window sizes above this limit, the standard deviation of results between folds becomes very large, indicating significant instability. Looking at the 24-hour sequence size, we observe an F1 score of 0.52 on the evaluation data, with interesting stability. This window has the advantage of practical interpretation, whereas the decision to validate or invalidate a 36-hour window seems tricky. Given the absence of a clear trend or a window size that stands out significantly from the others, we are maintaining a window size of 24 hours for subsequent tests.

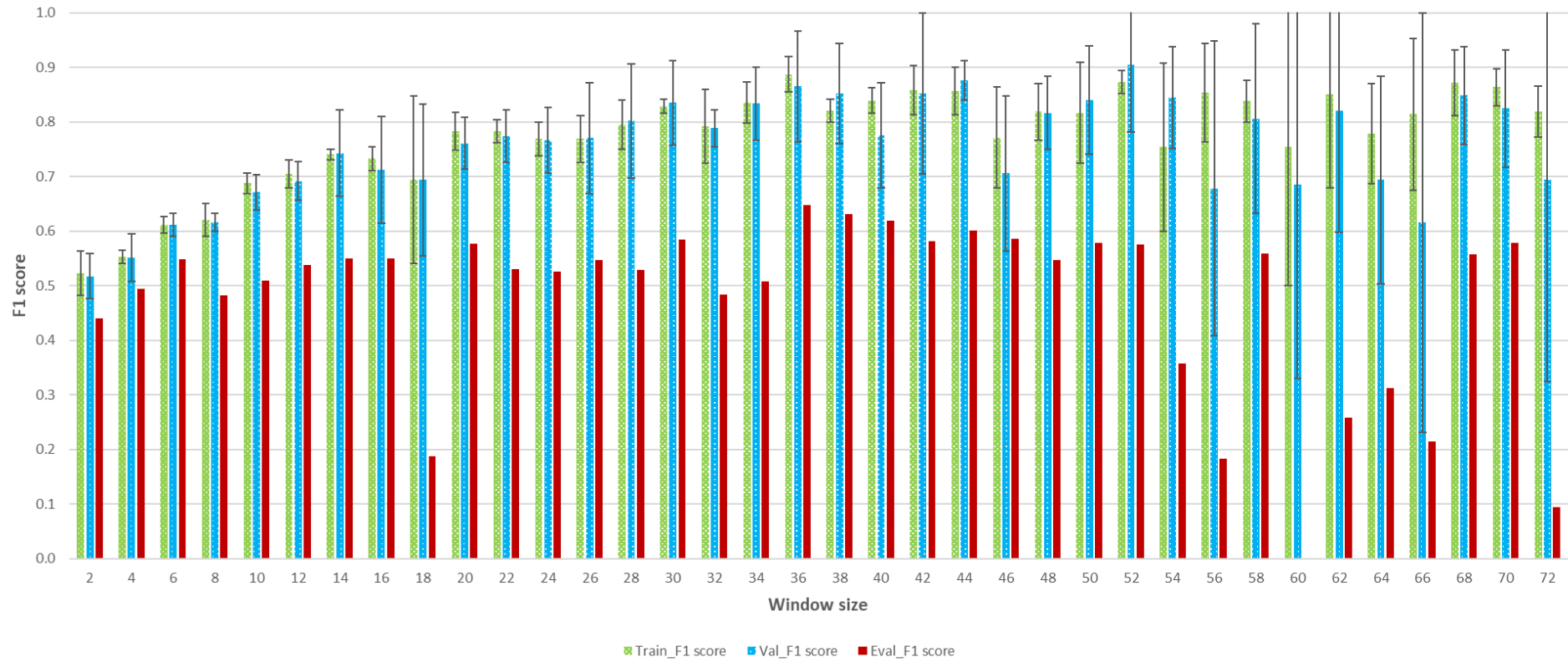


Figure 9-8: Results of sensitivity tests to the input sequence size

## 9.2.2. From probabilities to sequence label

### 9.2.2.1. Adapting the classification metric

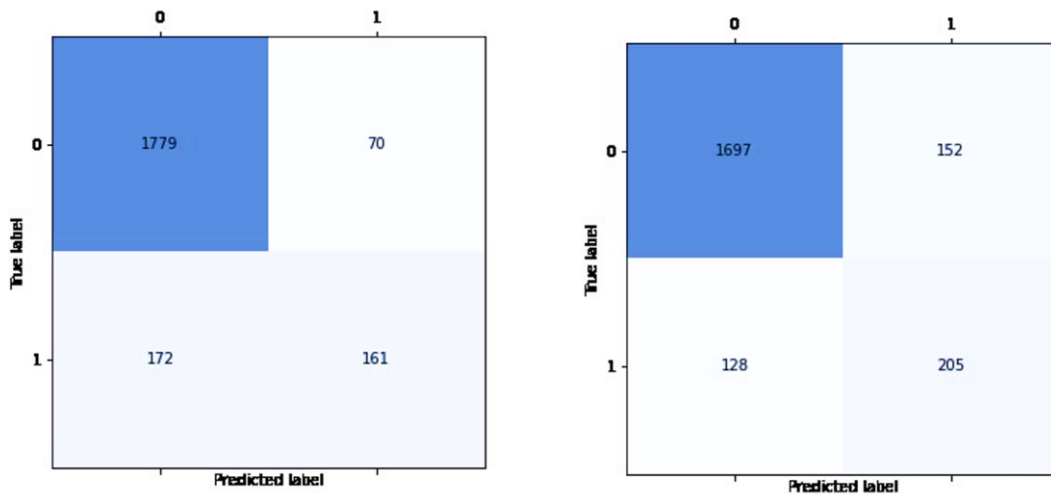
The tracking metric used during the training of the ResNet model and that allows to select the best model is the F1 score on the validation set. However, in our context, while false positives (FPs) can be submitted to a subsequent expert opinion for revalidation, false negatives (FNs) risk being lumped in with valid sequences with no possibility of re-evaluation. A preliminary approach would therefore be to adjust the weight of FNs in relation to FPs by adjusting the beta parameter of the F score (see [Equation 12](#)). [Table 37](#) summarizes the results obtained for different  $\beta$  values, where the latter is the weight attributed to the recall.

**Table 37: Evaluation results of ResNet model using F score as a loss function**

$\beta$	Only 100% valid and invalid sequences		All sequences	
	$F_\beta$	$F_1$	$F_\beta$	$F_1$
1	-	0.760	-	0.571
1.5	0.744	0.792	0.557	0.574
2	0.659	0.747	0.499	0.560
2.5	<b>0.768</b>	<b>0.812</b>	<b>0.609</b>	<b>0.594</b>
3	0.688	0.771	0.536	0.555

Analysis of the results reveals that focusing on the  $F_{2.5}$  score as a tracking metric improves the performance of sequences with only valid or invalid 100% tags. However, despite this improvement, overall performance remains modest. A comparison of the confusion matrices for both models, using respectively the  $F_1$  score and the  $F_{2.5}$  score, is presented in [Figure 9-9](#). Hence, we observe that there is a reduction in false negatives (FN), but this improvement is accompanied by a significant increase in false positives (FP), the latter having more than doubled.





Loss function	F1 score (Left)	F2.5 score (Right)
Precision	0.6970	0.5742
Recall	0.4835	0.6156
F <sub>1</sub> score	0.5707	0.5942
F <sub>2.5</sub> score	0.5048	0.5742
MCC	0.5209	0.5185

**Figure 9-9: Classification results of all sequences using the model issued according to different loss functions**

### 9.2.2.2. Adjusting the classification threshold

The best model is determined by taking into account the one that generates the best F1 score on the validation dataset, using 5-folds cross-validation. However, we consider it crucial to evaluate the model's performance on the complete data set. This is done using the ROC curve and/or the PR curve. Given our objective of maximizing the F1 score, we retroactively use the PR curve to identify the threshold for achieving the optimal F1 score over the entire training database. Thus, by applying a threshold of 0.4283 to the output of the ResNet model (which represents the probability of belonging to the class of interest, with a default threshold of 0.5), we manage to improve metric performance, as shown in [Table 38](#).

**Table 38: Results obtained with the a posteriori adjusted threshold**

	Precision	Recall	F1 score
Training database	0.8725	0.7355	0.7982
Complete database	0.5867	0.6096	0.5979

Once this threshold has been defined, it is interesting to evaluate the model's response by replacing the default classification threshold with the adjusted threshold. In this approach, a sequence is considered invalid as soon as its probability of belonging to the invalid class exceeds the adjusted threshold. To achieve this, a new learning phase is initiated, taking this adjusted threshold into account. The results obtained are summarized in [Table 39](#). We observe an improvement in the classification of sequences containing exclusively valid or invalid labels (the training database), while the impact on the whole database remains limited.

**Table 39: F1 score obtained with the adjusted threshold after re-training**

	Training phase		Evaluation phase	
	Average on training sets	Average on test sets	100% valid and invalid sequences	All subsequences
F1 score	0.813	0.779	0.839	0.597

Furthermore, in these tests, the adjusted threshold is provided with a precision of up to four decimal places. Thus, we need to assess the sensitivity of the model's response to this threshold, and to determine whether such a high level of precision is necessary to improve results. [Table 40](#) therefore proposes a sensitivity test to this threshold with a margin of +/- 10% and a threshold rounded to two decimal places.

**Table 40: F1 score obtained with approximated adjusted thresholds**

	Training phase		Evaluation phase	
	Average on training sets	Average on test sets	100% valid and invalid sequences	All subsequences
Adjusted threshold	0.813 ( $\pm 0.02^{12}$ )	0.780 ( $\pm 0.05^{12}$ )	0.839	0.597
Threshold + 10%	0.811	0.805	0.798	0.605
Threshold - 10%	0.825	0.808	0.738	0.545
Rounded threshold	0.828	0.825	0.784	0.590

We thus conclude that adjusting the classification threshold improves the results, particularly for sequences containing exclusively valid or invalid labels, without requiring great precision in its establishment. However, this approach requires a two-stage learning process: the first involves calibrating the model using the F1 score as an objective metric on the validation set. Next, this model is evaluated on the full data set, and the PR curve is analyzed to identify the score that maximizes the area under the curve, and hence the F1 score. Finally, new learning

<sup>12</sup> Refers to the standard deviation between the 5 folds during the training phase

is initiated by imposing this threshold as the classification threshold applied to the model output, which represents the probability of belonging to a particular class.

### 9.2.2.3. Combining both approaches

Thus, both strategies applied to the classification output show improved results. It is therefore legitimate to ask whether their combination could further benefit the model. To explore this possibility, we launch a training run using the F2.5 score as a tracking metric. Next, we exploit the PR curve on the whole data set, seeking to optimize the F1 score. Finally, we run a new training while maintaining the F2.5 score as the tracking metric. The final result on the training data is an F1 score of 0.759 and a score of 0.587 on the whole database. These results are less encouraging than those obtained with threshold adjustment alone. Thus, we conclude that the best approach remains the latter, even if it involves a tedious set-up with its two-phase learning. Nevertheless, this approach achieves an F1 score of 0.84, compared with the initial 0.76.

## 9.3. How can we improve the results ?

After establishing the input database (raw data from T1 and T2 over an 18-month period, including only 100% valid or invalid sequences, standardized over a 24-hour window with a stride equal to half the window) and determining the best classification strategy (two learning phases with an adjustment of the classification threshold based on the analysis of the PR curve), it is now pertinent to consider potential enhancements in results through the exploration of new approaches. Three specific issues warrant careful consideration: How can we leverage the strengths of our model by implementing pre-validation processes? Is binary classification truly the most effective approach to address the problem in a supervised context? What should be done with intermediate sequences, and how can we capitalize on their presence in the database? Hence, the exploration of a multiclass classification approach becomes relevant. Furthermore, conventional classification approaches require defining a threshold beyond which a sequence is deemed valid or not. What if we were to dispense with this threshold and repurpose the ResNet model to tackle a regression problem, where the objective is to directly predict the anomaly rate per sequence? All these avenues will be explored in the following sections.

### 9.3.1. Implementing pre-validation approaches

In this section, we implement a first stage of pre-validation for the model. The aim is to provide a base identical to that submitted to the expert. This step automatically invalidates trivial anomalies such as missing data, data outside the range of [1,1000], blocking or saturation. On the other hand, it automatically validates sequences that meet the redundancy criterion (see

**Equation 3**). Unlike model validation, which takes place at the sequence level, pre-validation, as described above, takes place at the measurement time step level. The two approaches are combined a posteriori. Classification using a saved model is quite fast. We are therefore not seeking to optimize this calculation time by preselecting sequences in advance. Our main objective is to consolidate the final result.

Once the ResNet classification has been carried out, we transform the labels at a time step of 5 minutes, assigning the same label (that of the sequence) to all the points that make it up. Then, according to an order of priority, a final label is assigned to each measurement (see **Figure 9-10**).

```

if trivial anomaly: label = "invalid";
elif redundancy: label = "valid";
) else label = output of the model

```

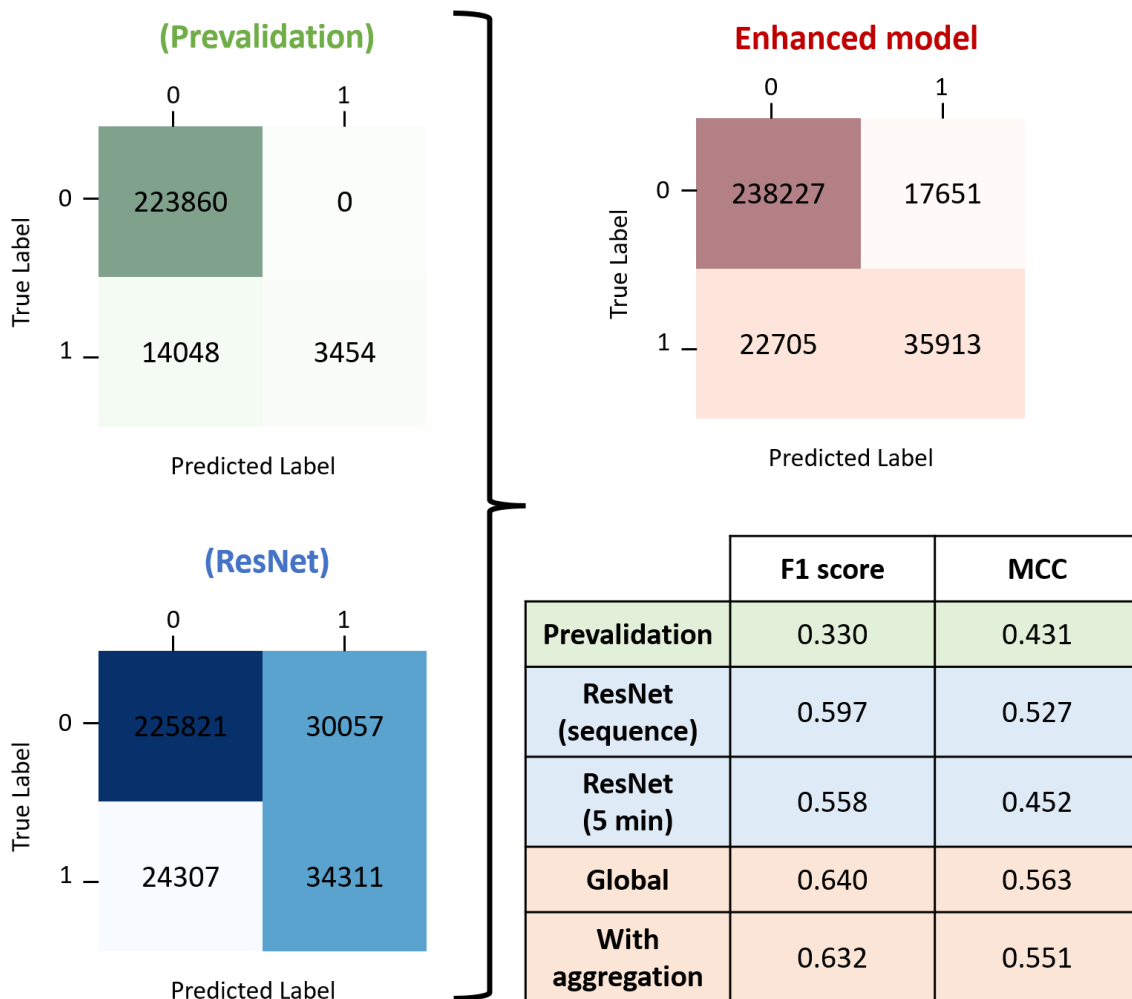
**Figure 9-10: Synopsis of the classification task enhanced with pre-validation**

**Figure 9-11** presents all the results. We already note that the expert invalidates several points that meet the redundancy criterion (FN of the pre-validation matrix). In practice, this results from the anomalies aggregation phase (see **Section 4.4.1**). Thus, even if data is valid according to the redundancy criterion, it can be invalidated depending on the context. Therefore, the presence of false negatives (FN) in the final result is to be anticipated. Hence, it is interesting to evaluate the outcome by implementing the same aggregation process (Row 5 of the table in **Figure 9-11**).

The ultimate outcome demonstrates mainly a decrease in the occurrence of false positives. These instances that were initially flagged as false positives by the ResNet model now align with the redundancy criterion, providing an opportunity to leverage the hardware redundancy that remains transparent for the ResNet model using a monivariate approach. Consequently, the final outcome enhances the ResNet model's performance, from an initial F1 score of 56% at the time step level to an F1 score of 64%. It should be noted that in this case, the subsequent merging of defects does not improve the results, and even degrades them by adding some false positives.

Nevertheless, from a practical standpoint, the focus may shift towards a more operational validation approach, allowing for the determination of whether a day (a 24-hour sequence) is valid or not without delving into individual time steps. **Figure 9-12** illustrates the results obtained based on the threshold applied to the global model outputs and the expert's results. It is observed that optimal performance is achieved with a threshold of 4.5 hours. In other

words, if the criterion is to invalidate a day with at least 4.5 hours of anomalies (either consecutive or not), an F1 score of 0.693 is attained. As indicated in [Section 4.4.2](#), the average duration of observed anomalies is approximately 4 hours. Thus, our threshold aligns closely with this average. While this approach may result in overlooking some shorter anomalies, it ensures the identification of anomalies exceeding this threshold, which could potentially have a more significant impact on the quality of the database. On the other hand, a threshold equal to 9 hours provides a better balance between precision and recall without degrading the final MCC, which remains around 0.589, and a slightly lower F1 score of 0.674.



**Figure 9-11: Enhanced model results combining the ResNet and a pre-validation phase**

To sum up, the implementation of a preliminary pre-validation indicates a notable reduction in false positives, improving the ResNet model's performance to an F1 score of 64% and an MCC of 56%. The focus on a daily validation approach achieves optimal performance with a threshold of 4.5 hours, leading to an F1 score of 0.69%. This aligns with the average duration of observed anomalies.

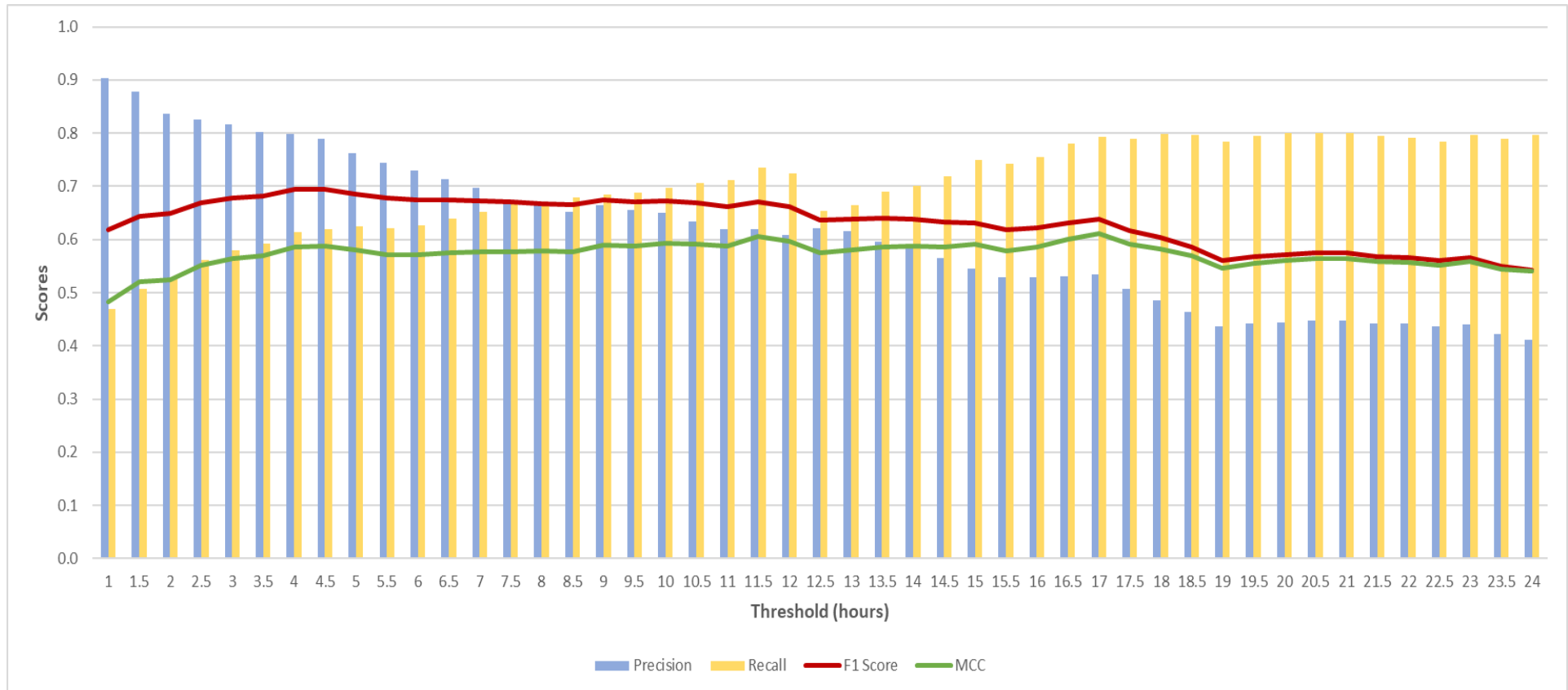


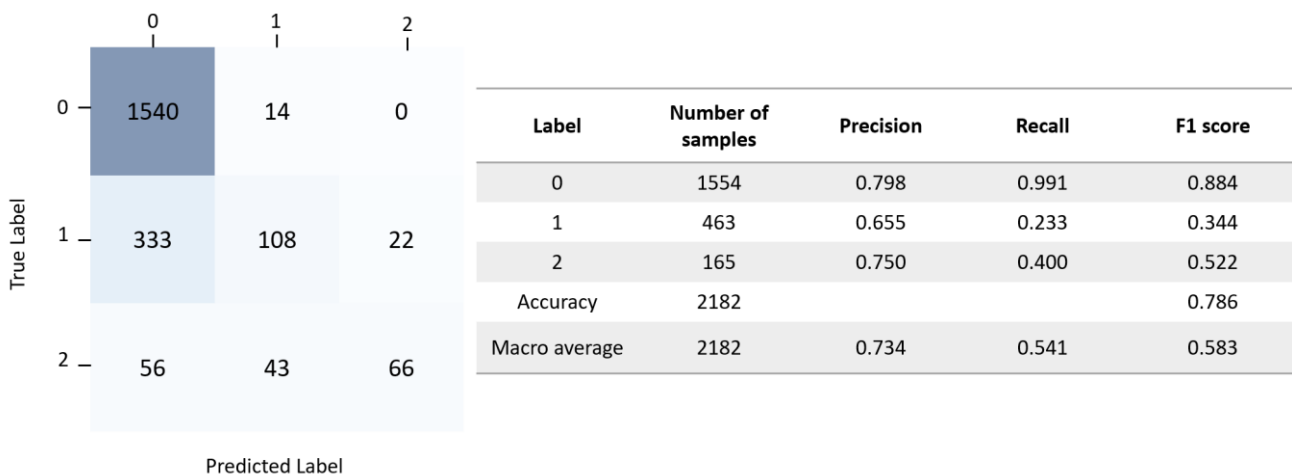
Figure 9-12: Performance metrics according to the classification threshold applied to the expert 5-minutes scale validation and the results of the enhanced ResNet model

### 9.3.2. Multiclass classification

Based on tests carried out on the binary classification model using ResNet, it was found that the most effective method is to train with either fully valid or fully invalid sequences. Then, when the model is deployed (after the two training phases), a sequence is presented to the model. Depending on its probability of belonging to each of the two classes and the adjusted classification threshold, the sequence is assigned a label. However, this approach has marked disadvantages for sequences whose probability is around the threshold limit. Thus, it is relevant to evaluate performance in the context of a multiclass classification, including sequences that are clearly valid, clearly invalid, as well as intermediate sequences whose classification is uncertain and may require further expertise. We aim to have relatively high confidence in the first two classes.

Figure 9-13 shows the results of classification using three classes, where:

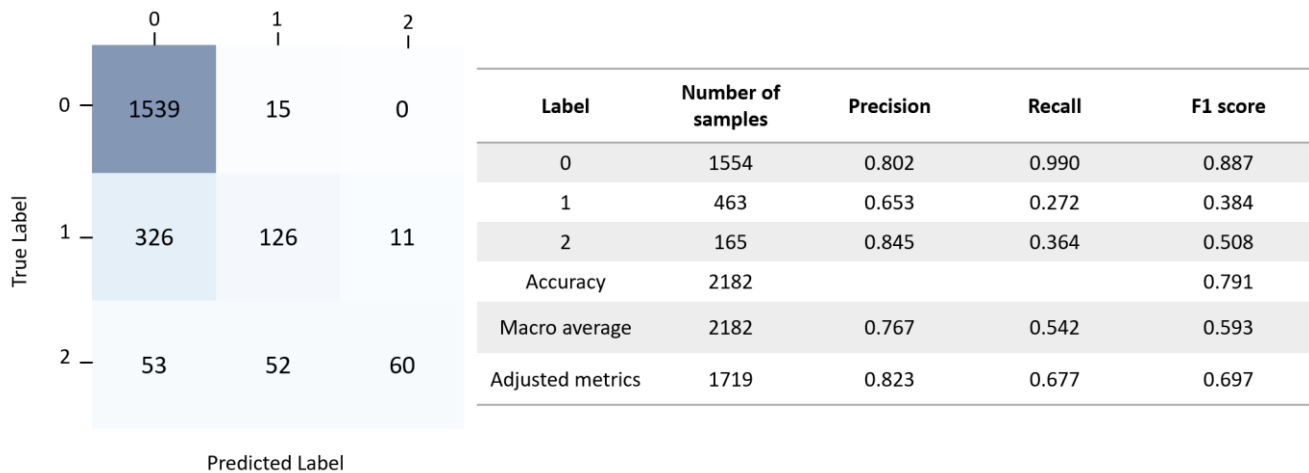
- Label 0: Valid class: sequences with an anomaly rate between 0% and 20%
- Label 1: Intermediate class: sequences with an anomaly rate between 20% and 80%
- Label 2: Invalid class: sequences with an anomaly rate between 80% and 100%



**Figure 9-13: Multiclass classification using ResNet and a threshold of 20%**

We observe that the F1 macro score in this scenario is 0.583. The score for class 1 is the lowest, as the model runs into difficulties with the intermediate sequences, which end up scattered across the three classes. What's more, the same problem is observed for the sequences of the invalid class. The model classifies a limited amount of data in this category (a total of 88), whereas the initial database contained 165 sequences in this class. Only the valid class presents consistent results, with a recall of 0.99%, where incorrectly classified sequences are grouped in the intermediate category. Consequently, the introduction of the third intermediate class seems to disrupt the learning process. So, instead of seeking to optimize the macro F1 score, which is the reference metric in multiclass classification, we will

guide the model by imposing the optimization of an adjusted F1 score equal to the average of the F1 scores of classes 0 and 2. The aim is to reduce the impact of the intermediate class by giving it less weight during training. The results are highlighted in [Figure 9-14](#), showing a slight improvement of the performance metrics.



**Figure 9-14: Multiclass classification using an adjusted score**

Indeed, this approach presents a problem, as even when conditioning the F1 score, the number of intermediate sequences is significant, exceeding that of the class of interest (invalid). Consequently, we re-evaluated the model's performance by adjusting the threshold to 40%. The three classes are redefined as follows:

- Label 0: Valid class: sequences with an anomaly rate between 0% and 40%.
- Label 1: Intermediate class: sequences with an anomaly rate between 40% and 60%.
- Label 2: Invalid class: sequences with an anomaly rate between 60% and 100%.

[Figure 9-15](#) summarizes the results obtained. Despite the increase in adjusted F1 score and accuracy, mainly attributed to an improvement in recall, it can be seen that the model does not classify any sequences in the intermediate class. Instead, it simply divides its sequences into valid and invalid classes. What's more, the model has difficulty in correctly identifying invalid sequences, with a recall of 0.615 (meaning it's missing 38.5%), in addition to the sequences it mistakenly invalidates.

In the end, it appears that this approach does not significantly improve the results, introducing instead a bias linked to the definition of the intermediate class. Furthermore, it does not meet our initial hypothesis / will of having two clearly defined classes, i.e. valid and invalid. In this context, the predictions merge the classes, making it difficult to select sequences with an intermediate anomaly rate.



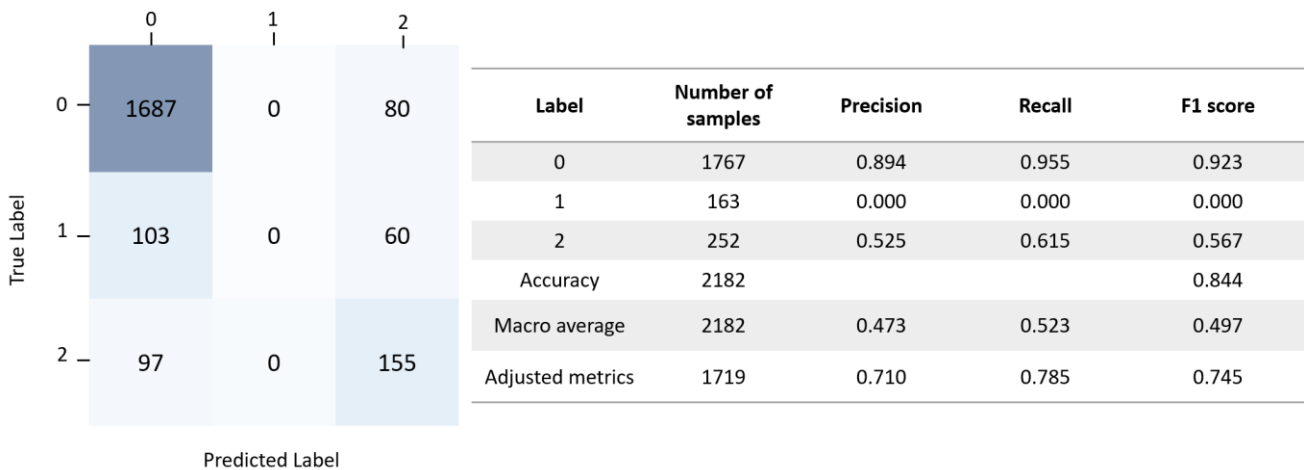


Figure 9-15: Multiclass classification using ResNet and a threshold of 40%

### 9.3.3. Predicting the anomaly rate per sequence

A third approach to investigating the use of the ResNet model involves modifying it to predict the anomaly rate of each input sequence, rather than providing a probability of being anomalous. The aim of this approach is to free ourselves from classification thresholds and the bias introduced by class definition when training the model. This modification is based on the model structure described in Section 5.4.4.4.

Examination of the anomaly rates generated by the model reveals an inability to categorically predict sequences as either fully valid or invalid (see Figure 9-16). This observation suggests an ambiguity in the model's response.

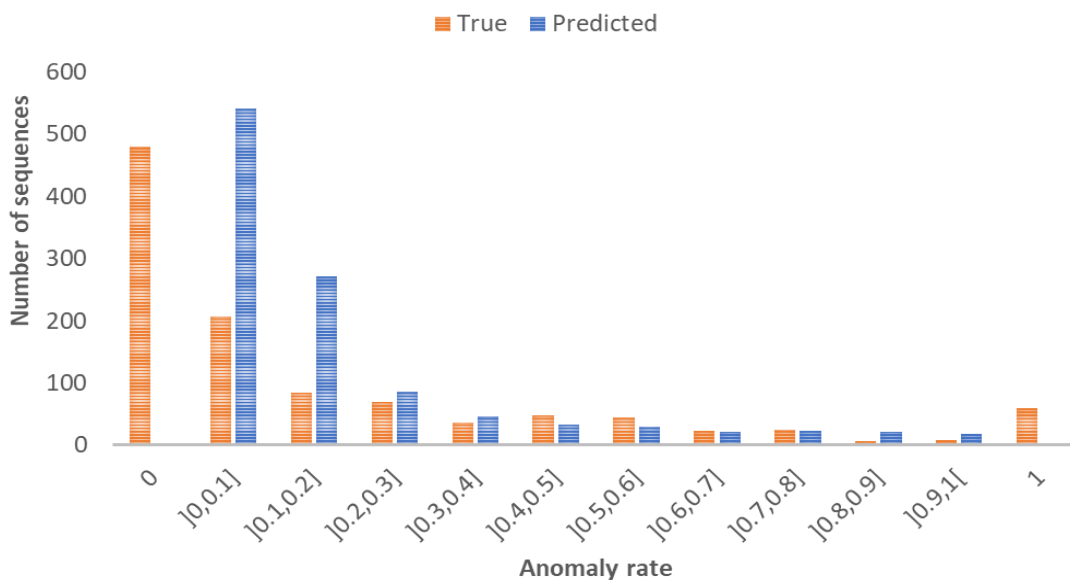
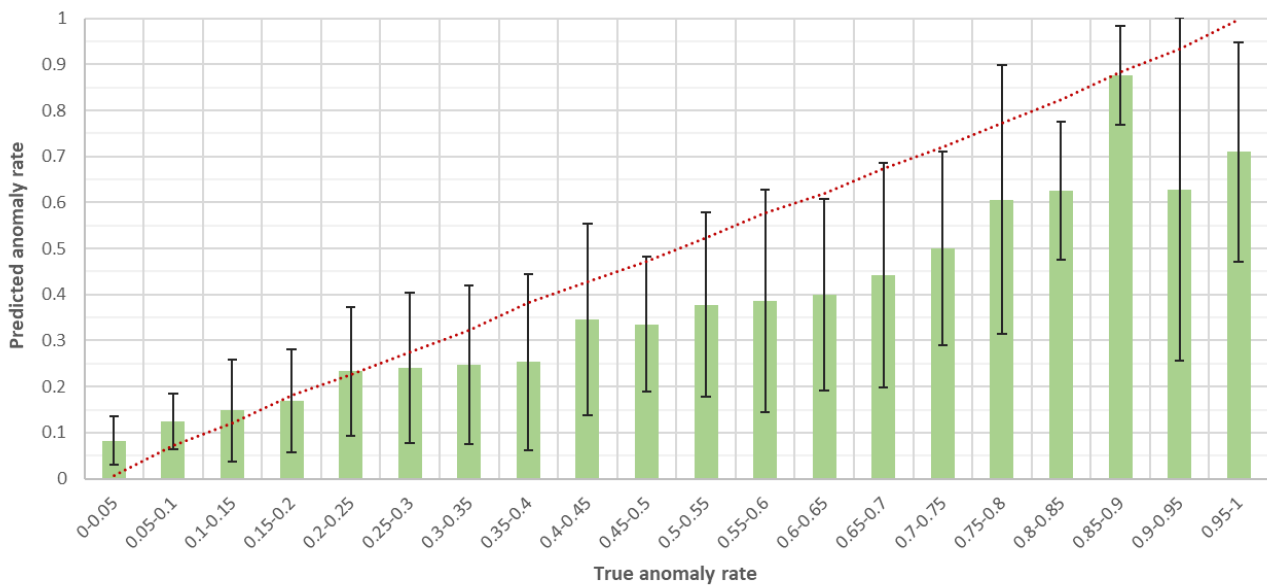


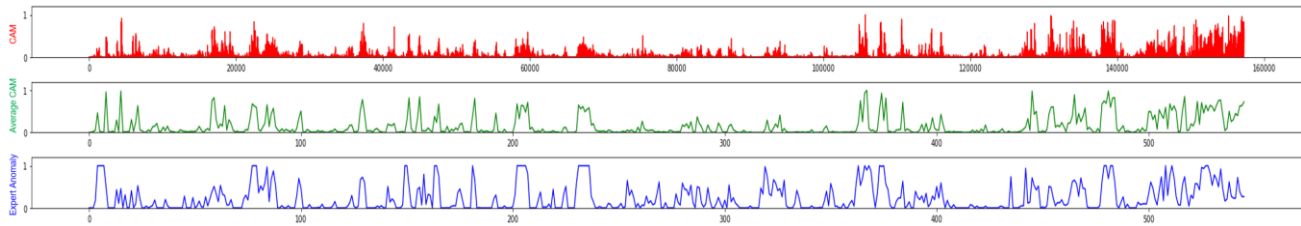
Figure 9-16: Histogram of anomaly rates predicted by the model versus true anomaly rates

**Figure 9-17** shows the comparison between the anomaly rate predicted by the model and the actual anomaly rate of the sequences. Significant variability in results is observed, particularly for sequences with high anomaly rates. A tendency to overestimate the anomaly rate is identified for sequences with low anomaly rates, systematically for sequences with anomaly rate reaching up to 10% and on average for those around 25%. On the other hand, a quasi-systematic underestimation is observed between 50% and 75%, with an average well below reality. This analysis highlights the complexity of accurately predicting anomaly rates for different sequence classes.



**Figure 9-17: Comparison of true anomaly rate per sequence and the predicted anomaly rate**

On the other hand, the prediction of an anomaly rate raises an additional problem: how can these anomalies be precisely located within the sequence itself? If a sequence has an anomaly rate of 25%, where exactly are these 25% invalid time points located? To resolve this question, we are exploring the Class Activation Maps (CAM) of the ResNet model. The principle behind this method is detailed in **Appendix H** for a more in-depth understanding. **Figure 9-18** shows the CAM output in red, its average per sequence in green, and the anomaly rate identified by the expert in blue.



**Figure 9-18: CAM results compared to true anomaly rate. the CAM output in red, its average per sequence in green, and the anomaly rate identified by the expert in blue**

Despite the seeming correlation between these different representations, evaluation of the correlation between CAM at the point scale and manual validation using the biserial point reveals a very low correlation coefficient, below 0.5. Even at the sequential scale, although the correlation coefficient reaches 0.7, it remains difficult to establish a bijective relationship between the CAM value and the actual anomaly rate of the sequence. This finding underlines the complexity of the task of assigning specific anomalies to precise locations in a sequence, despite the use of CAM as a visualization tool.

A final exploratory approach using the regression model is to consider it as a preliminary to the classification problem, rather than directly exploiting the anomaly rate as the final result. **Figure 9-19** shows the F1 classification score for different thresholds applied to both the model output and the manual validation result. Overall, the best results are close to the diagonal up to an anomaly rate of 0.7, as identified by the model, showing a good correlation between the anomaly rate identified by the model and that resulting from manual validation (redundancy + expertise + aggregation). Nevertheless, the best results are concentrated on low anomaly rates. In absolute terms, the best F1 score obtained is around 77% for an anomaly rate of about 10%. In other words, if we invalidate the days as soon as around 2.5 hours are identified as invalid, we obtain an F1 score of 0.77. This result surpasses the best score obtained with a binary classification approach, which was 0.69.

This finding suggests that using the regression model as a preset for classification can offer a significant improvement in performance, even if it means invalidating the whole day without accurately identifying the fault location, as is already the case with classification approaches where a sequence is invalidated without necessarily knowing how abnormal it may be.

2T	Prediction																			
Targets	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
0.05	0.683761	0.775746	0.752056	0.670143	0.613668	0.555066	0.503049	0.456693	0.419355	0.375415	0.348562	0.296684	0.269504	0.228675	0.205882	0.151515	0.11583	0.071146	0.036217	0
0.1	0.605602	0.746862	0.770833	0.72807	0.678233	0.632107	0.575916	0.525362	0.484171	0.435453	0.405512	0.346939	0.316008	0.269231	0.24295	0.179775	0.137931	0.085106	0.043478	0
0.15	0.558176	0.716648	0.770515	0.759055	0.71453	0.67031	0.622137	0.572565	0.528689	0.476596	0.444444	0.380952	0.347222	0.300716	0.271845	0.20202	0.15544	0.096257	0.049315	0
0.2	0.517799	0.679679	0.767204	0.767947	0.743169	0.709552	0.659836	0.61242	0.566372	0.511521	0.477541	0.409877	0.373737	0.328982	0.297872	0.222222	0.171429	0.106509	0.054711	0
0.25	0.479668	0.635714	0.739264	0.767606	0.752896	0.721992	0.678337	0.633028	0.589074	0.540943	0.505102	0.44385	0.405479	0.357955	0.324638	0.243161	0.188088	0.117264	0.060403	0
0.3	0.428816	0.586767	0.707993	0.752363	0.759916	0.740406	0.698565	0.654912	0.623037	0.582418	0.555241	0.489552	0.447853	0.396166	0.359477	0.275862	0.214286	0.134328	0.069498	0
0.35	0.392982	0.544516	0.688245	0.751491	0.768212	0.748201	0.709184	0.668464	0.646067	0.615385	0.587156	0.524272	0.48	0.432056	0.392857	0.30303	0.23622	0.14876	0.077253	0
0.4	0.377325	0.531414	0.677083	0.743902	0.764706	0.753695	0.713911	0.672222	0.649275	0.623853	0.601266	0.543624	0.49827	0.449275	0.408922	0.316206	0.246914	0.155844	0.081081	0
0.45	0.34358	0.493927	0.636528	0.712154	0.739857	0.73107	0.703911	0.676558	0.664596	0.644737	0.627986	0.567273	0.518797	0.466403	0.439024	0.347826	0.272727	0.173077	0.090452	0
0.5	0.308403	0.448468	0.584906	0.663677	0.707071	0.705556	0.698507	0.675159	0.675585	0.676157	0.666667	0.611111	0.567901	0.513043	0.484305	0.386473	0.304569	0.194595	0.102273	0
0.55	0.268431	0.392496	0.514851	0.612827	0.668464	0.674627	0.670968	0.67128	0.678832	0.6875	0.685714	0.643172	0.614679	0.565854	0.535354	0.428571	0.337209	0.2125	0.119205	0
0.6	0.233365	0.35119	0.466942	0.56	0.622857	0.649682	0.657439	0.671642	0.679842	0.689362	0.696429	0.669903	0.639594	0.608696	0.576271	0.459627	0.370861	0.230216	0.138462	0
0.65	0.212683	0.324242	0.436441	0.525773	0.591716	0.622517	0.635379	0.65625	0.680498	0.699552	0.707547	0.690722	0.67027	0.639535	0.606061	0.496644	0.402878	0.251969	0.152542	0
0.7	0.193294	0.29584	0.412148	0.503979	0.568807	0.597938	0.609023	0.628571	0.652174	0.688679	0.696517	0.688525	0.678161	0.658385	0.649351	0.536232	0.4375	0.275862	0.168224	0
0.75	0.168	0.258268	0.362416	0.446281	0.504792	0.541516	0.571429	0.597403	0.62963	0.676768	0.684492	0.686391	0.6875	0.680272	0.671429	0.564516	0.491228	0.313725	0.193548	0
0.8	0.147624	0.227564	0.325688	0.403409	0.456954	0.503759	0.539419	0.563636	0.595122	0.641711	0.647727	0.658228	0.657718	0.647059	0.651163	0.548673	0.466019	0.32967	0.219512	0
0.85	0.141988	0.219002	0.314088	0.389685	0.441472	0.486692	0.521008	0.543779	0.574257	0.630435	0.635838	0.658065	0.657534	0.646617	0.650794	0.563636	0.48	0.340909	0.227848	0
0.9	0.136317	0.210356	0.302326	0.375723	0.425676	0.469231	0.502128	0.523364	0.552764	0.607735	0.611765	0.631579	0.629371	0.615385	0.617886	0.542056	0.474227	0.329412	0.210526	0
0.95	0.130612	0.201626	0.290398	0.361516	0.416382	0.459144	0.491379	0.511848	0.540816	0.595506	0.598802	0.61745	0.614286	0.598425	0.6	0.538462	0.468085	0.317073	0.219178	0
1	0.121026	0.186885	0.270142	0.337278	0.395833	0.436508	0.46696	0.485437	0.513089	0.566474	0.567901	0.583333	0.577778	0.57377	0.591304	0.545455	0.47191	0.311688	0.235294	0

Figure 9-19: From anomaly rate per sequence to classification: F1 score results according to the invalidation threshold

## 9.4. Generalization to other sites

In this section, we'll look at various approaches to evaluate the best ResNet model assessed so far with a regression approach on other sites of SMA. We'll start by evaluating the performance of the best model on sites other than the one on which it was initially trained, namely Cottage. This analysis will enable us to determine to what extent the model is capable of generalizing its learning. We will then address the possibility of creating a generic model. This model, configured with the same parameters, will be trained on the dataset from all sites and evaluated both on the global dataset and individually for each site. Finally, we will explore the transfer learning strategy by building site-specific models. In doing so, we will assess how transfer learning can improve model performance by leveraging knowledge gained from other sites.

### 9.4.1. Direct evaluation on other sites

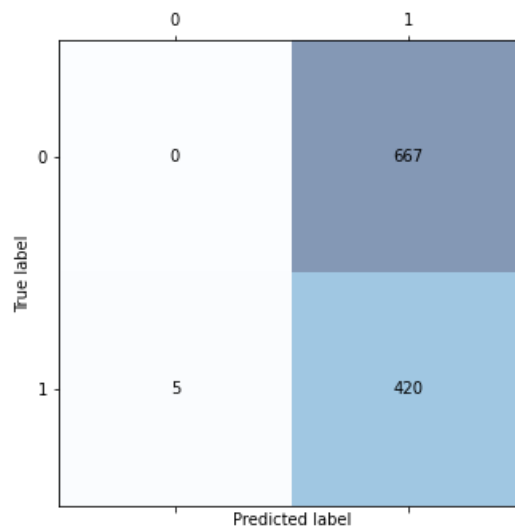
The aim of this phase is to directly evaluate the best model previously identified and saved for Cottage on turbidity data from other sites, without any prior adaptation, while maintaining the same pre-processing steps. These steps include the use of a 24-hour time window and data standardization. **Table 41** shows the results of the F1 score and the MCC for each site.

**Table 41: Direct evaluation of ResNet on other sites of SMA**

	Precision	Recall	F1 score	MCC
<b>Cottage</b>	0.763	0.778	0.770	0.658
<b>Antilles</b>	0.993	0.597	0.746	0.550
<b>Découverte</b>	0.841	0.748	0.792	0.616
<b>Goutte</b>	0.979	0.569	0.720	0.605
<b>Roosevelt</b>	0.386	0.988	0.556	-0.085
<b>All sites</b>	0.690	0.709	0.700	0.411

Examining the performance metrics, we observe that “Antilles” and “Goutte” sites exhibit similar results with a high precision (an average of 0.986) but a low recall slightly surpassing 0.55. This result indicates that the model is very accurate when predicting the positive class. However, the recall means that the model lacks a number of truly positive instances. This can be interpreted as a tendency of the model to be too conservative in predicting the positive class, hesitant to declare certain instances as positive, even if they are. On the other hand, “Découverte” demonstrates a well-equilibrated performance with an F1 score of 0.792, which is close to the results of the training site "Cottage".

Furthermore, the “Roosevelt” site displays a challenging scenario with a low precision of 0.386 and a high recall of 0.988, resulting in an F1 score of 0.556 and an MCC of -0.085. This pattern indicates that the model struggles to correctly identify positive instances while avoiding false positives. The low precision suggests that when it predicts a positive class, it tends to be incorrect in most cases. When we analyze the confusion matrix of this site, we can see that the model tends to predict almost a unique class; namely the invalid class (see [Figure 9-20](#)). This imbalance can be attributed to a variety of factors, such as site-specific features at “Roosevelt” that make it distinct from other sites and where the classification threshold seems very penalizing.



**Figure 9-20: Confusion Matrix of the ResNet model trained on Cottage and evaluated on Roosevelt**

In conclusion, the direct evaluation of the ResNet model on various sites within SMA presents a nuanced performance, revealing distinct patterns across different locations. In light of these findings, the next phase involves retraining the model using data from various sites, with a focus on addressing these specific challenges and enhancing overall performance.

## 9.4.2. Training the best model using data from different sites

### 9.4.2.1. Reinitiation the learning process

This phase involves reinitiating the learning process by using 24-hour time window and standardized sequences from various sites. The objective is to facilitate the model in comprehending the diverse dynamics of normal operations across different locations. Notably, the “Roosevelt” site is excluded from this phase due to notable differences compared to the other interceptors, as elaborated in [Section 4.3.3](#). [Table 42](#) summarizes the performance results.

**Table 42: Results using a ResNet model trained on the whole database from different sites**

	Precision	Recall	F1 score	MCC
<b>All sites</b>	0.599	0.950	0.734	0.372
<b>Antilles</b>	0.750	0.979	0.849	0.379
<b>Cottage</b>	0.360	0.986	0.528	0.211
<b>Découverte</b>	0.620	0.884	0.729	0.396
<b>Goutte</b>	0.683	0.950	0.795	0.530

Analyzing the results, the model exhibits an overall commendable performance when considering all sites collectively. The precision of 0.6 suggests a relatively accurate identification of positive instances, while the recall of 0.95 indicates a high capacity to capture the majority of true positive instances. The F1 score corroborates the model's effectiveness in maintaining a balanced classification. But the MCC shows a low correlation between expert validation and model output. These results are better than those obtained by evaluating the Cottage-specific model on the other sites if we consider the gain on recall, but the latter causes a loss of 10% of the precision.

Examining individual sites, we observe a general improvement of the recall, i.e., the capacity to detect most anomalies but the precisions tend to decrease globally. The "Cottage" site presents challenges, particularly with a notably lower precision of 0.36. This suggests that the model struggles with an increased number of false positives on the Cottage site.

In summary, the reinitiation of the learning process with data from various sites yields a global improvement of performance results but a nuanced performance across different locations. The model excels in capturing the dynamics of normal operations, yet site-specific variations still influence its performance, highlighting the importance of tailored approaches for different operational contexts.

#### 9.4.2.2. Total fine-tuning of the learning process

The aim of this phase is to exploit transfer learning approaches to improve and accelerate the learning process. The used technique is total fine-tuning, where the model is initialized with the best "Cottage" model, and learning is restarted for all layers using data from the other sites. Compared with the approach presented in [Section 9.4.2.1](#), the only difference lies in the initialization of the initial weights, aimed primarily at optimizing computation time. Analysis of the learning curves for both approaches (see [Figure 9-21](#)) reveals that this technique does indeed allow us to start with a lower initial loss value. However, the speed of convergence is slightly slower, as the model approaches the tangent of the performance curve. Despite this,

at the end of the 600 epochs, the loss functions are lower, confirming the benefits of this technique.

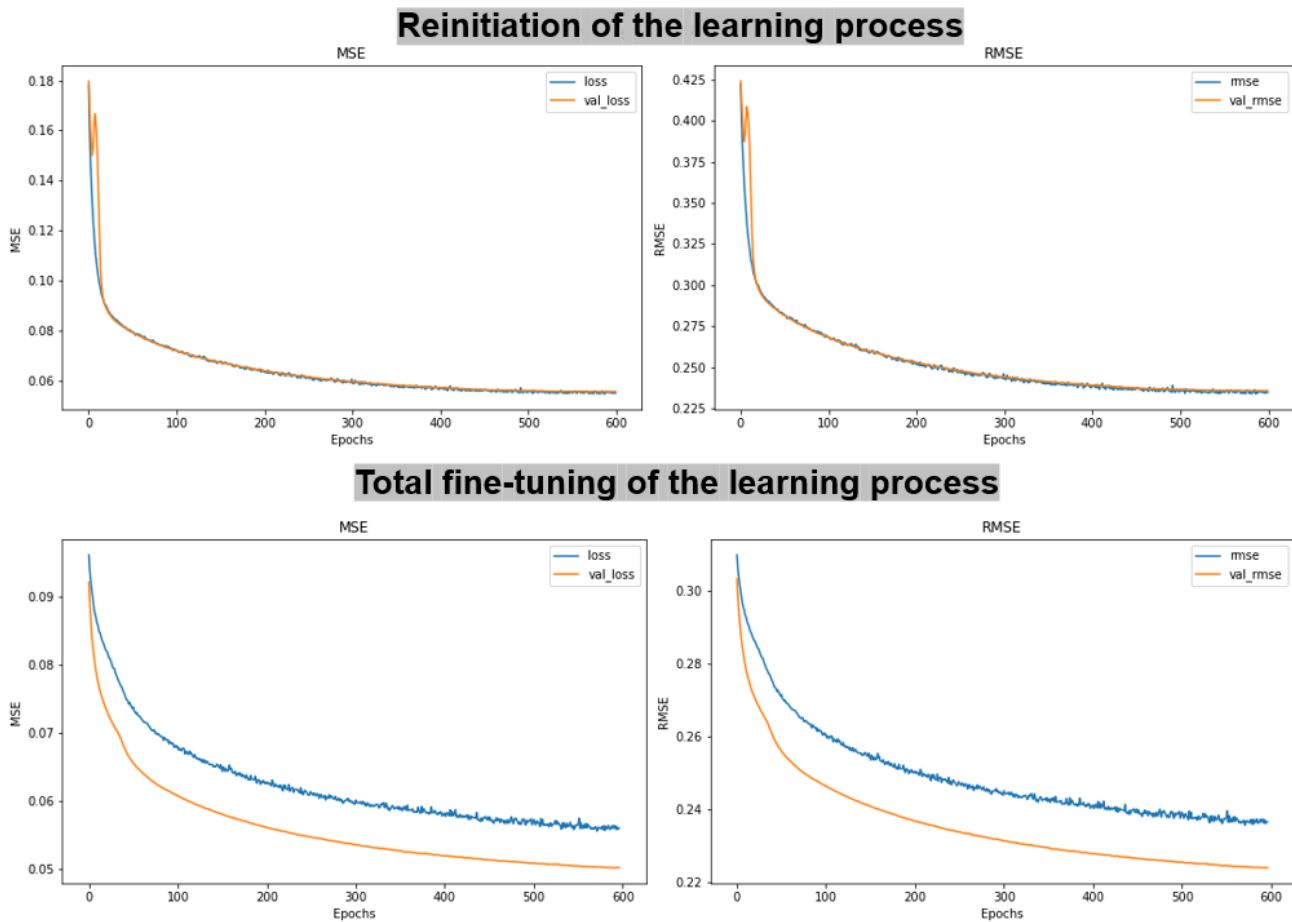


Figure 9-21: Comparison of the learning curves using different training strategies

The model performance results, trained on the whole data set with this approach, are summarized in Table 43. The performance obtained is of the same order of magnitude as that obtained previously, with a slight improvement. However, it is important to note that it is not possible to state statistically that this improvement is significant.

Table 43: Results using a total-fine tuning with data from all sites

	Precision	Recall	F1 score	MCC
All sites	0.626	0.921	0.745	0.407
Antilles	0.797	0.934	0.860	0.468
Cottage	0.378	0.972	0.544	0.253
Découverte	0.586	0.886	0.706	0.324
Goutte	0.789	0.905	0.843	0.655



### 9.4.3. Tuning a specific model for Roosevelt

The aim of this phase is to design a model specific to the “Roosevelt” site, whose hydraulic operation differs from that of the other sites. To this end, we adopt the same architecture as that of the best model described in [Section 9.3](#), while applying preprocessing steps similar to those used for Cottage. However, we evaluate different relearning techniques.

- **Test 1: Reinitiate the learning process**

This approach aims to reboot the model from scratch, keeping the same architecture but allowing specific adaptation to the characteristics of “Roosevelt”.

- **Test 2: Feature extraction learning**

This technique involves using the model previously trained on Cottage to extract features relevant to Roosevelt, without completely readjusting the model weights. For this, only the weights of the last layer are adjusted while freezing the rest.

- **Test 3: Partial fine-tuning of the learning process**

This approach focuses on the selective readjustment of model layers, according to the particularities of Roosevelt station, while preserving the knowledge acquired on Cottage. Concretely, we freeze the first 50% layers and adjust only the weights of the last layers

[Table 44](#) synthesizes the results of the different approaches.

**Table 44: Results of Roosevelt data validation using a specific model and different learning strategies**

	Precision	Recall	F1 score	MCC
<b>Test 1</b>	0.643	0.781	0.706	0.493
<b>Test 2</b>	0.362	0.753	0.489	-0.113
<b>Test 3</b>	0.653	0.741	0.695	0.482

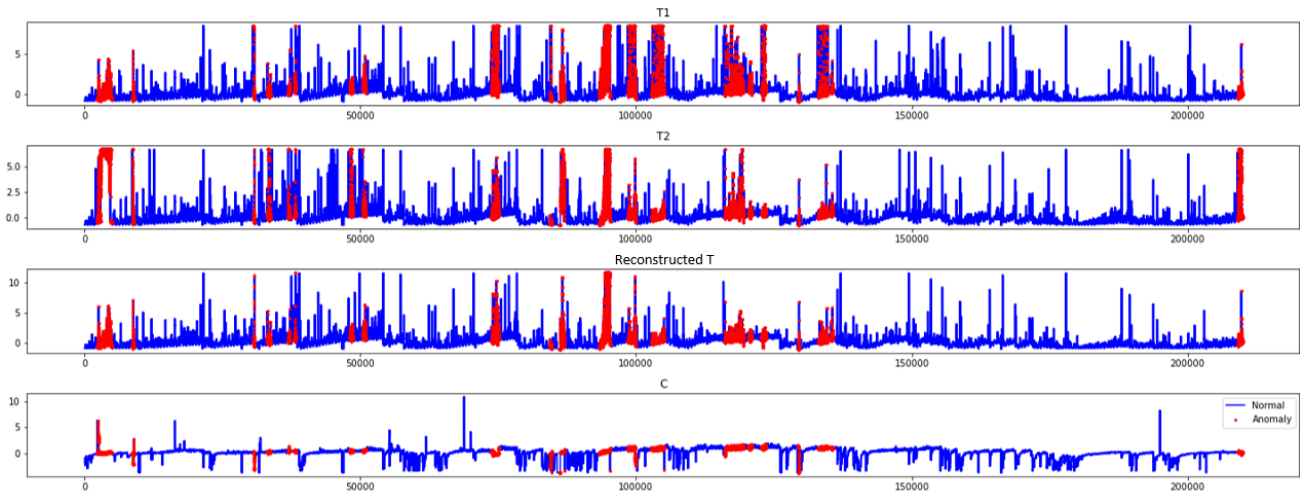
For the first test, we observe relatively balanced precision and recall, which suggests that starting the learning process anew allows the model to better capture positive instances, resulting in an improved overall F1 score and MCC compared to other approaches. Idem, the last approach demonstrates a balanced trade-off between precision and recall, contributing to a competitive F1 score and MCC. While the second test exhibits challenges, with a lower precision and a negative MCC. This suggests that relying solely on feature extraction may not be sufficient to effectively capture the specificities of the Roosevelt site, leading to a suboptimal overall model performance.

In conclusion, the analysis of the results suggests that starting the learning process anew (Test 1) and the partial fine-tuning (Test 3) are needed to capture the inherent data structure of “Roosevelt”. On the other hand, feature extraction learning (Test 2) exhibits limitations, indicating the importance of a more advanced training for optimizing model performance on site-specific dynamics.

⇒ In summary, the direct evaluation on various SMA sites reveals nuanced performance patterns. Retraining the model with data from different sites shows an overall improvement but with site-specific challenges. The overall performance is commendable, with improved recall but nuanced precision variations across sites. Transfer learning through total fine-tuning exhibits potential benefits, although the significance requires further investigation. Tailoring the model to the Roosevelt site, which is hydraulically different, demands careful consideration. The results indicate that rebooting the learning process and partial fine-tuning are effective strategies for capturing the site-specific data structure, while feature extraction learning exhibits limitations.

### 9.5. Multivariable anomaly detection

The aim of this section is to evaluate a multivariate approach. So far, the training of the ResNet model has been done on a univariate basis, thus losing any link between the two turbidimeters. To remedy this, our aim is to provide the model with different measures as a whole. When the model identifies a sequence as abnormal, it also assigns this label to each of its variables, as illustrated in [Figure 9-22](#). However, this approach can sometimes seem inappropriate, as the invalidity of one datum is often linked to a lack of concordance of the other variables, thus representing contextual anomalies. However, these contextual anomalies do not mean that all variables are invalid in themselves. Cases where all variables are invalid simultaneously are more likely to occur in the context of global faults, such as a connection problem or a power failure.



**Figure 9-22: Data validation results of Cottage data using a multivariable approach**

**Table 45** presents a summary of outcomes obtained through diverse approaches, incorporating distinct input data. The input sequences adhere to a 24-hour length with a half-window stride, and data standardization is applied to each variable. The output encompasses both the actual anomaly rate of the sequence and the predicted anomaly rate. Sensitivity tests are conducted for both the expert classification threshold and the model classification threshold. In **Appendix L**, detailed results, depicted through heatmaps representing various combinations, are provided.

With this in mind, the table provides, for each configuration, the thresholds for achieving the best performance, as well as the corresponding F1 score value. Despite variability from one test to another, a general trend emerges, indicating that optimum performance is generally obtained with a manual validation threshold (target) of 0.66 and a model threshold of 0.42 on average (see **Section 9.3.3**). It should be noted that the model threshold is lower than the expert threshold, suggesting that the model is potentially more tolerant than the expert. Therefore, with a threshold of 0.42, the sequence actually represents an anomaly rate of 0.66. Comparing the F1 scores of the different configurations, we see that "2TC" and "3TC" perform similarly, and slightly better than "2T" and "3T", suggesting that the introduction of conductivity has improved anomaly detection. These results, especially with the conductivity as input data, highlight an improvement of the classification scores compared to a monovariable approach.

**Table 45: Results of the multivariable approach according to the input data**

	2T	3T	2TC	3TC
<b>Prediction threshold</b>	0.4	0.35	0.45	0.5
<b>Target threshold</b>	0.65	0.6	0.7	0.7
<b>F1 score</b>	0.800	0.722	0.832	0.830

## 9.6. Synthesis of Chapter 9

The in-depth evaluation of the ResNet model, focusing on assessing its performance using Cottage turbidity dataset, has yielded several crucial results. Firstly, by looking at the model's sensitivity to input data, we highlighted the importance of carefully pre-processing this data, adjusting strides to avoid over-fitting. Data enhancement techniques, such as noise augmentation and up sampling, were explored to manage class imbalance. The use of data from several sites stabilized the results but did not lead to a significant improvement over the exclusive use of Cottage data. Sensitivity tests revealed nuances in model performance, notably increased accuracy with reconstructed turbidity data.

Next, hyperparameters tuning was undertaken without modifying the ResNet model architecture. A key question was the optimal size of the time window, with tests showing that optimal performance was achieved with a 36-hour window, although stability problems were encountered with longer windows. Maintaining a 24-hour window was preferred for further testing, due to its stability and practical interpretation. With regard to sequences validation, adjustments were made to the classification threshold, calling into question the relevance of the default threshold of 0.5. The results showed that adjusting the weight of false negatives versus false positives using the F-score beta parameter improved performance, but with a trade-off between reducing false negatives and increasing false positives. Classification threshold adjustment was also explored using the PR curve, significantly improving metrics and demonstrating the positive impact of this approach.

New approaches to improving results were also explored. The implementation of pre-validation showed promising results, reducing the ResNet model's false positives and improving its overall F1 score to 64%. The exploration of multiclass classification, while interesting, revealed challenges, particularly with the intermediate class. Finally, using the ResNet model to predict the anomaly rate per sequence showed encouraging results, even outperforming the classification model, with an F1 score of 77%.

Tests were also carried out to generalize the best model to other sites within SMA, with multiple objectives. Firstly, to assess the generalizability of the ResNet model initially trained on the "Cottage" site, by testing it directly on other sites. The results revealed nuanced performance, with significant variations in precision and recall between different sites, underlining the challenges of generalization. Then, the exploration of site-specific transfer learning techniques, particularly for the "Roosevelt" site, demonstrated the need for custom strategies to capture site-specific data structures. Finally, the multivariate approach was introduced, showing an improvement in anomaly detection with the inclusion of conductivity.

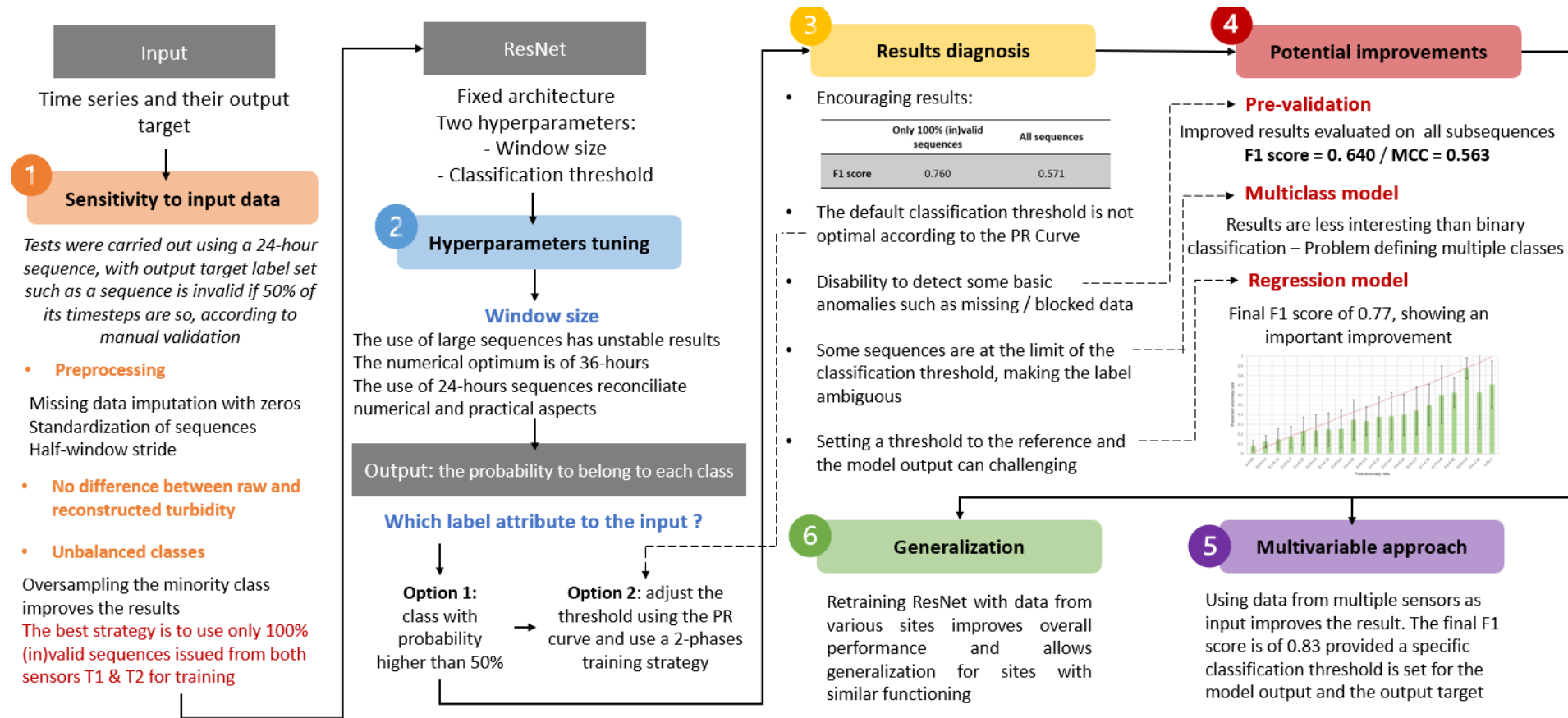


Figure 9-23: Overview of ResNet tests and results for anomaly detection using turbidity data

# Chapter 10. Autoencoder evaluation

---

The operating principle of the autoencoder (as described in [Section 5.5](#)), involves feeding the model with input sequences and training it to reproduce the same sequence on output. The sequences used for this purpose are taken from turbidity data from Cottage from February 2021 to July 2022. Further tests will be presented in terms of processing these sequences, their size and number (using the complete set of sequences or just a pre-selection) ([Section 10.1](#)). Subsequently, the model output is compared with the input by calculating a mean square error (MSE) per sequence. However, to perform a sequence classification task, an output classifier is applied to establish a threshold beyond which a sequence is considered valid or invalid.

The architecture of the model is not fixed at this stage and will be the subject of specific tests ([Section 10.2](#)). To guarantee the model's stability, several runs are launched, and an evaluation is carried out on the complete Cottage data set. The aim is to take account of model variability, which will be represented in the results graphs by error bars. For each run, the model producing the best results among the various folds is saved. In this way, when the model is deployed or subject to further improvements ([Section 10.3](#)), the best model is directly selected from all runs. In some cases, the confusion matrices illustrated, specific to a given model, may differ from the average performance, depending on the model's stability. The final tests, as with the other models, consist in evaluating the model's performance against data from other sites ([Section 10.4](#)) and then considering a multivariate approach ([Section 10.5](#)).

## 10.1. Sensitivity to input data

The aim of this section is to carry out input sensitivity tests on the AE model. Given the limited size of our dataset as previously highlighted during ResNet evaluation, all T1 and T2 sequences will be used as input to create an enhanced database. The default sequence size is set at 24 hours, and sensitivity tests on this parameter will be carried out later in this manuscript ([Section 10.2.3](#)). A sequence will be considered abnormal as soon as a time step is abnormal, which represents a significant constraint. Adjustments to this parameter will also be examined later ([Section 10.3.2](#)). The architecture of the autoencoder is basic, with a single hidden layer representing a latent space of 64 neurons. The aim of these tests is to assess the model's sensitivity to different **scaling approaches**, such as normalization or standardization.

### 10.1.1. Preprocessing

Within this framework, we conducted a series of varied tests to evaluate the performance of our autoencoder model under different input configurations (see [Table 46](#)). The tests include the use of the full input dataset with minmax normalization or standardization, as well as tests on the exclusive use of valid data. All these tests are carried out with a linear activation function. However, to complete the tests mentioned above, we have introduced an additional experiment designed to explore the model's sensitivity to the activation function. Thus, we included a combination of minmax normalization with sigmoid activation. It's important to note that sigmoid activation is incompatible with standardization, as the former constrains outputs between 0 and 1, while standardization has a wider evolution interval.

**Table 46: List of tests conducted to evaluate AE's sensitivity to input data preprocessing**

	Input	Normalisation	Activation
1	All	MinMax	Linear
2	All	StandardScaler	Linear
3	Only valid	StandardScaler	Linear
4	Only valid	MinMax	Linear
5	Only valid	MinMax	Sigmoid

To compare the results obtained in our experiments, we have exploited two distinct classification approaches: namely the 3-sigma rule and the PR curve approach. [Table 47](#) exhaustively summarizes the results obtained using these two approaches.

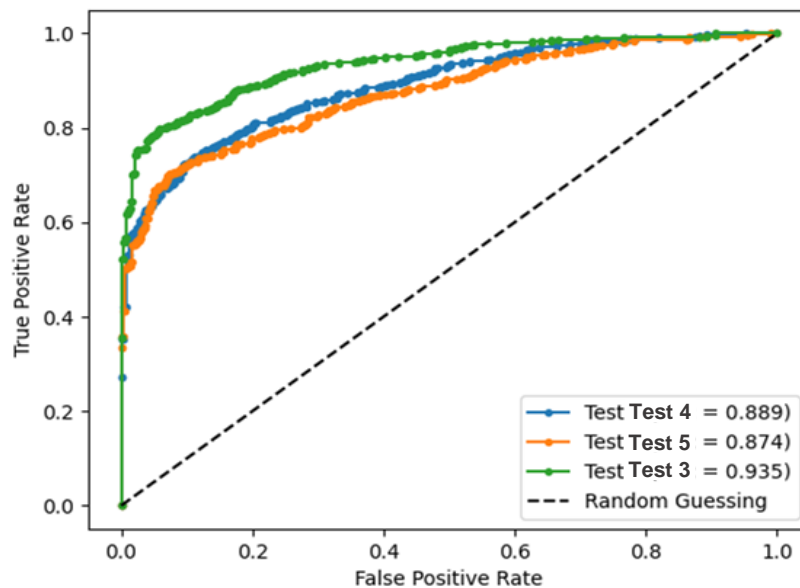
**Table 47: Results of input processing - autoencoder**

		Test 1	Test 2	Test 3	Test 4	Test 5
<b>PR Curve</b>	Precision	0.661	0.681	0.866	0.835	0.888
	Recall	0.943	0.910	0.877	0.810	0.735
	<b>F1 score</b>	<b>0.777</b>	<b>0.779</b>	<b>0.872</b>	<b>0.823</b>	<b>0.804</b>
	MCC	0.404	0.419	0.706	0.604	0.614
<b>3-sigma rule</b>	Precision	0.892	0.872	0.761	0.820	0.794
	Recall	0.457	0.466	0.944	0.812	0.799
	<b>F1 score</b>	<b>0.605</b>	<b>0.607</b>	<b>0.843</b>	<b>0.816</b>	<b>0.796</b>
	MCC	0.424	0.410	0.612	0.584	0.535

The results of Tests 1 and 2 highlight that using the complete set of sequences in the learning phase results in significantly lower scores. Although the F1 score does not seem to show a

significant difference between both tests, this is explained by the bias induced by the prevalence of abnormal sequences, here reaching 56%. The MCC review confirms this finding, pointing out that the very nature of the auto-encoder, focused on reconstruction by learning discriminatory elements, is hampered in the presence of anomalies in the training set. Generally, autoencoders excel in a semi-supervised approach. The use of the MSE distribution (3-sigma rule) in this configuration is also problematic, inducing a substantial gap of about 22% on the F1 score compared to PR curve results. Indeed, the presence of anomalies biases the mean and the standard deviation of the MSE, making the 3-sigmas rule inadequate.

For the other tests, the disparity between the results of the PR curve and the rule of the 3-sigmas is less marked. In particular, Test 3 stands out as the best performing, favoring the use of standardization. Tests 4 and 5, on the other hand, show no significant difference in terms of the impact of the activation function. This observation is corroborated by the comparison of ROC curves, where those of Tests 4 and 5 are similar, while that of Test 3 stands out more as the best model.



**Figure 10-1: ROC Curves for input sensitivity test on the autoencoder**

**These conclusions make it possible to recommend the adoption of a semi-supervised approach, with the exclusive use of 100% valid sequences in the training phase, associated with scaling using standardization.** Performance evaluation can be done using both approaches, with the PR Curve providing a maximum evaluation of performance to be reached after an adjustment of the x-sigma rule parameter, which can sometimes differ from 3.



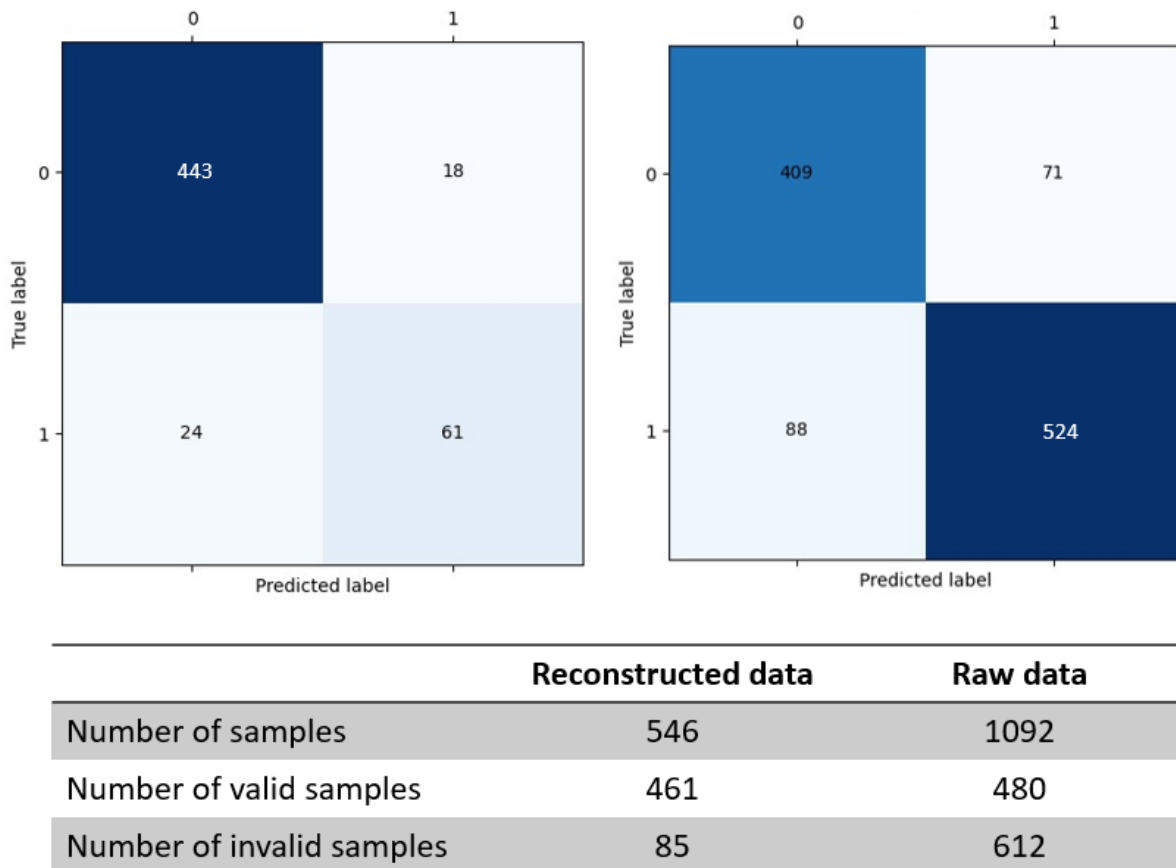
### 10.1.2. Input data

The aim of this test is to evaluate the model's response to various inputs, namely raw data and reconstructed turbidity. To this end, we fix the test hypotheses by applying the same pre-processing (standardization + linear) and training exclusively on fully valid sequences. In terms of architecture, a basic autoencoder (AE) is used, with a code composed of 64 neurons. An initial observation concerns the disparity in the size of the input databases, given that using raw data generates twice as much data. However, when focusing on 100% valid sequences, sample sizes are almost equivalent, with 480 samples for raw data and 461 for reconstructed data. The training databases are therefore considered equivalent in terms of size. The results of this test are compared in [Table 48](#).

**Table 48: Results for different input data using the PR approach**

	Precision	Recall	F1 score	MCC
Raw data	0.866	0.877	0.872	0.706
Reconstructed data	0.739	0.729	0.733	0.685

We observe that performance deteriorates when reconstructed data is used (see [Figure 10-2](#)). This can be explained by the fact that the number of invalid samples remains limited, and consequently the slightest error can have a significant impact on performance metrics. Although the number of false positives and negatives is lower in the case of reconstructed data, they represent an important ratio regarding the number of invalid samples.



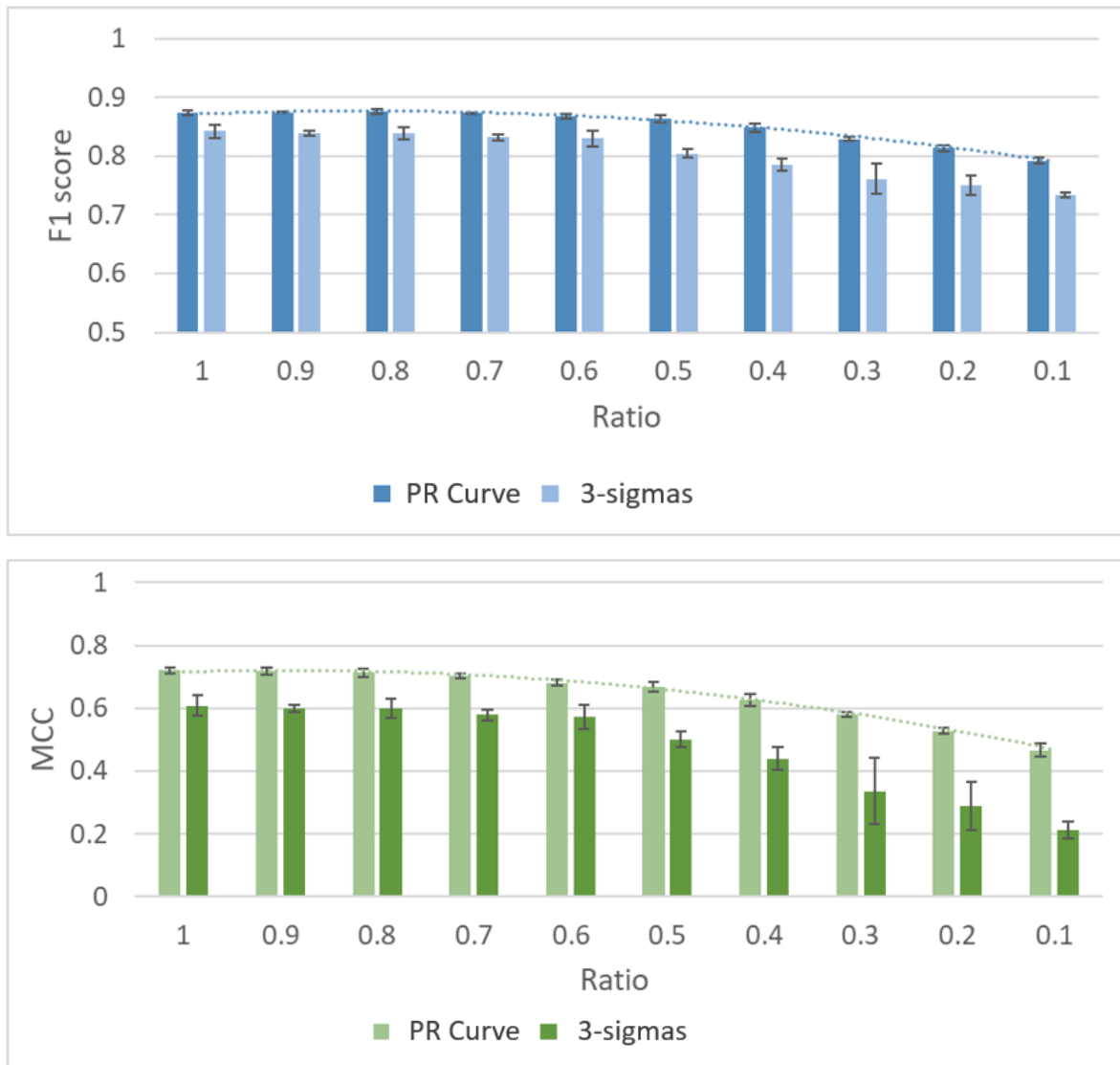
**Figure 10-2: Confusion matrix and performances depending on the input data – Left: reconstructed data, Right: raw data**

### 10.1.3. Size of the input database

The objective of this test is to assess the model's ability to learn more effectively as the size of the training database varies. Given the limitation of our data set and its complete use in the learning phase, a question arises: is it possible to improve the performance of the model by having more data? To answer this question, we conducted tests by gradually reducing the size of the training database and evaluating the overall F1 score at each stage. In practice, for each sub-database resulting from the progressive reduction of the size of the training set, we repeated the tests several times. This approach aims to ensure the stability of the results<sup>13</sup> and to overcome the random effect associated with the selection of the subset of data. The objective is to identify a potential trend in favor of improved performance with an increase in the size of the data. This exploratory approach aims to determine whether the model reaches a saturation threshold in terms of performance or whether it would actually benefit from an increase in the amount of data.

<sup>13</sup> The results provided are averages on the different runs

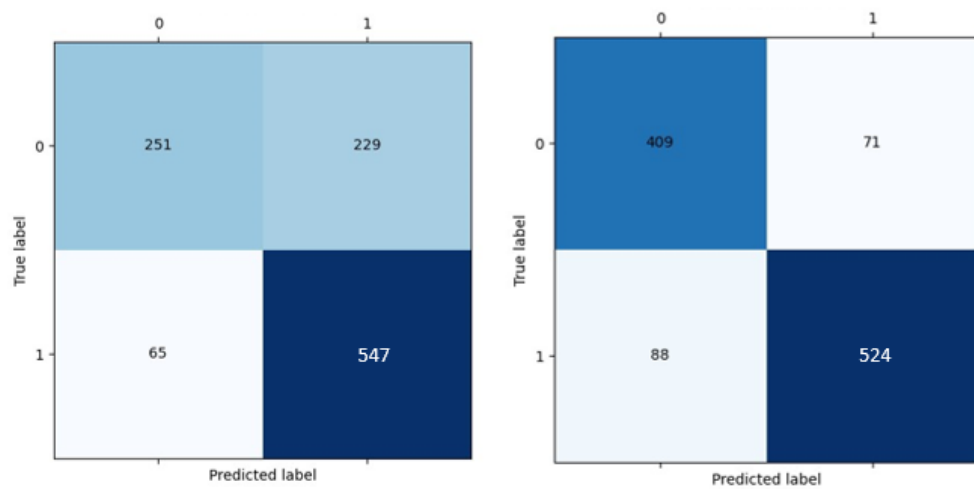
The analysis of the results reveals an interesting trend: the model seems to reach a performance plateau starting from a ratio of 0.7, indicating that using only 70% of our database is enough to stabilize the performance of the autoencoder model (see [Figure 10-3](#)).



**Figure 10-3: Performance metrics according to the ratio of input data used for training**

However, for lower data ratios, the MCC has a tendency to decrease significantly compared to the F1 score, while the latter remains relatively high. In order to elucidate this observation, let's consider the validation confusion matrix issued from the model that was trained on 0.1 of the database and that was established using the PR Curve approach. The analysis of the confusion matrix for a ratio = 0.1 highlights a notable ability to effectively identify anomalies, thus contributing to the observed high F1 score. However, there is a significant gap in the classification of valid data. When examining the confusion matrix with a ratio equal to 0.1 compared to that with a ratio = 1, the detection of valid sequences appears particularly

problematic, with the model assigning these sequences almost randomly between valid and invalid classes. This inefficiency in distinguishing valid sequences results in relatively low MCC.



	Ratio = 0.1	Ratio = 1
<b>Precision</b>	0.705	0.881
<b>Recall</b>	0.894	0.856
<b>F1 score</b>	0.788	0.868
<b>MCC</b>	0.456	0.706

**Figure 10-4: Confusion matrix and performances depending on the ratio of database**

In addition, the increasing gap between the two evaluation approaches (PR Curve and 3-sigmas), especially when MCC is taken as a reference, highlights an important challenge in the evaluation of the performance of the AE model. In fact, the decrease in the number of samples in the use of reduced percentages of the training database makes the establishment of an average MSE and a standard deviation less reliable, resulting in larger error bars for lower ratios. This challenge results in problems in establishing a precise and reliable threshold for classification, explaining the poor results with a significant deviation from the optimal performance obtained with the PR curve approach. This deviation tends to decrease as the database size increases, allowing a better statistical representativeness of the 3-sigma rule.

## 10.2. Hyperparameters tuning

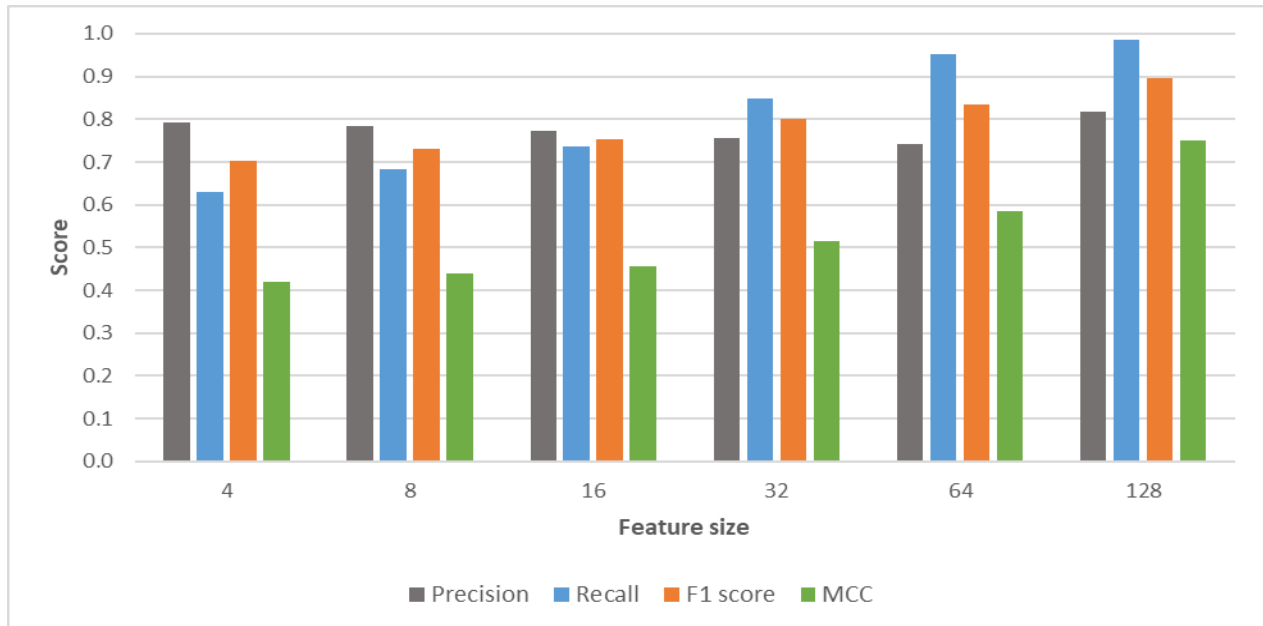
The judicious choice of the number of neurons and layers forms the backbone of neural network design, representing essential hyperparameters that shape its power and complexity. The number of neurons per layer influences the model's ability to capture complex patterns in

the data, while the number of layers determines the depth of the network, enabling abstract hierarchical features to be extracted. A network with a large number of neurons may be able to model complex relationships, but this can also lead to over-fitting problems, while a network with too few neurons may lack representational capacity. Similarly, an architecture with multiple layers enables the model to learn more abstract representations, but this may require a larger dataset to be effective. So, the delicate balance between the number of neurons and layers is crucial to optimizing the performance of a neural network and achieving an ideal match with the underlying complexity of the data.

### 10.2.1. Sensitivity to the feature size

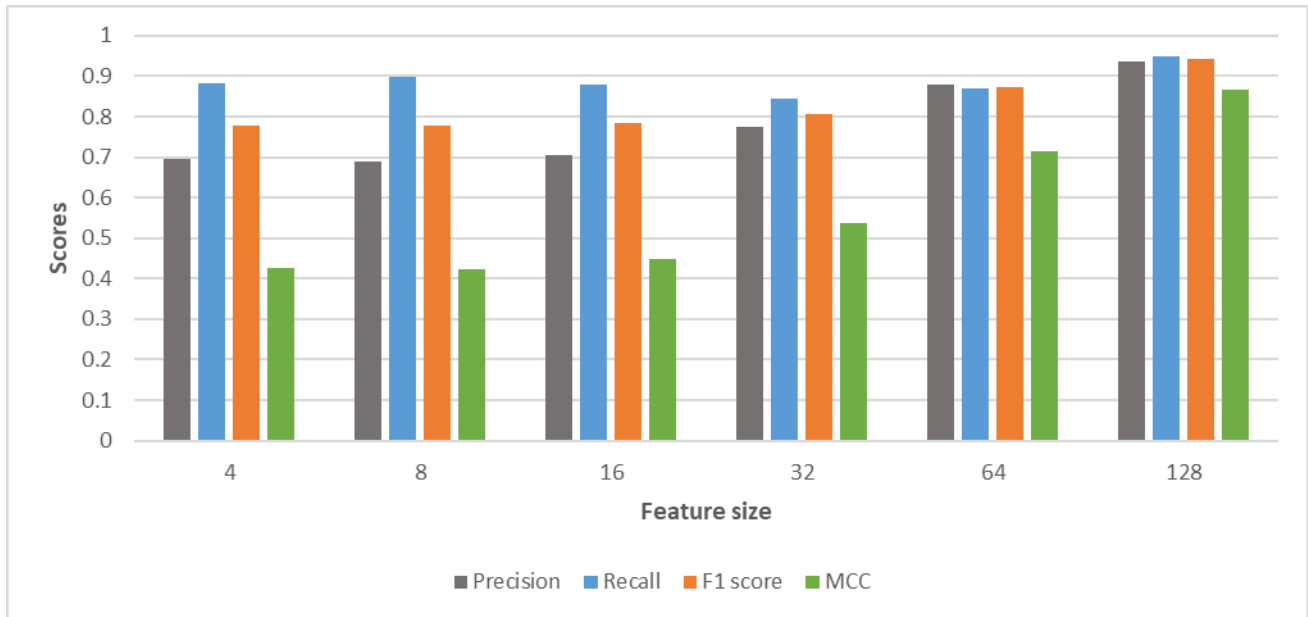
In the process of tuning hyperparameters for an AE, we start with a basic architecture that consists solely of latent space as a hidden layer (see Architecture 1 in [Table 10](#)). Using latent space as the only hidden layer initially simplifies the network structure, enabling the sensitivity of the model to this fundamental parameter to be explored. The size of the latent space, which represents the compression of data into a more compact representation, determines the autoencoder's ability to reconstruct inputs in a meaningful way. By progressively adjusting the size of the latent space, we can assess how the model reacts to different dimensions, thus determining the best configuration for the specific task. Having input sequences of 24 hours, i.e., an input shape = 288, and in order to respect the bottleneck configuration of the autoencoder, the size of the latent space is varied between 4 and 128.

The analysis of performance metrics such as precision, recall, F1 score and MCC as a function of latent space dimension reveals significant trends (see [Figure 10-5](#)). Using an approach based on the 3-sigma rule, the increasing of the latent space dimension is positively correlated with improved performance, particularly with regard to F1 score and MCC. The 3-sigma rule, here, demonstrates that higher latent space dimensions translate into higher performance, with an F1 score approaching 0.9 and an MCC of 0.75 for a latent space of 128 neurons. However, an interesting observation emerges: although recall increases, indicating an enhanced ability to identify anomalies, precision remains relatively stable. This stability suggests that increasing the size of the latent space enables more effective anomaly detection without generating a disproportionate number of false positives. Then, the question is: can we achieve even more interesting performances? Performance metrics using the PR curve approach are then analyzed.



**Figure 10-5: Results of sensitivity to the latent space size using the 3-sigma rule**

The results based on the PR curve as a function of latent space size within the autoencoder framework reveal significant trends. Model performance confirms a substantial improvement with increasing latent space size. In contrast to the results of the 3-sigma rule, precision exhibits continuous growth, rising from 0.695 to 0.935, indicating an increased ability to minimize false positives. Although recall shows a less regular variation, an overall increase is observed, rising from 0.843 to 0.948 as the dimension of the latent space increases. This observation suggests that the model becomes more effective in identifying anomalies. Consistently, the F1 score, representing the balance between precision and recall, follows a significant upward trajectory, rising from 0.778 to 0.942. Finally, the MCC shows a notable improvement, up to 0.866. These results are better than the 3-sigma approach, which suggests that further improvements can be achieved by optimizing the anomaly detection threshold.



**Figure 10-6: Results of sensitivity to the latent space size using the PR curve approach**

The analysis of the ROC curves of the different models consolidates these results (see [Figure 10-7](#)). In particular, the ROC curves of models with latent space dimensions of 4, 8, and 16 have marked similarities, all displaying AUCs around 0.8. These results suggest a relatively equivalent discrimination capacity between normal and abnormal classes for these three models. In contrast, the model with a latent space dimension of 128 stands out significantly, exhibiting a remarkable AUC of 0.97. The ROC curve of this model is considerably closer to the perfect model, indicating an exceptional ability to discriminate between the two classes. This disparity in performance between models can be attributed to the increased capacity of the model with a latent space dimension of 128 to capture subtle features in the data, thus reinforcing its ability to perform accurate classification.

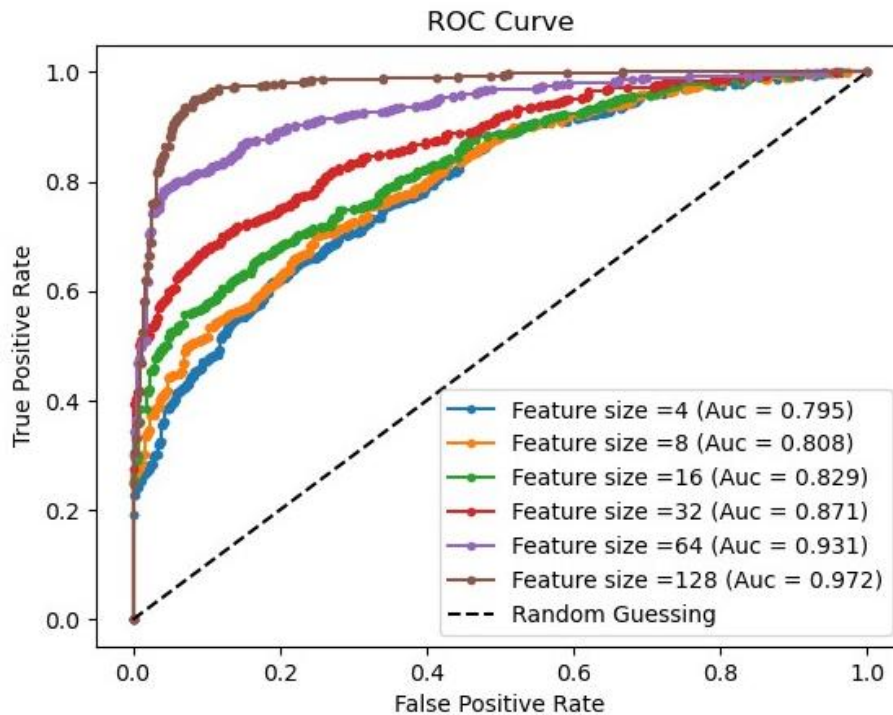


Figure 10-7: ROC Curves for the different models of architecture 1

With such remarkable performances, it is natural to question the occurrence of over-fitting, a condition that could compromise the generalization of the model to new data.

To do this, we differentiate two types of overfittings. The first is related to the number of epochs during learning. It occurs when the model continues to learn on the training data even after the performance on the validation data has reached an optimum and begun to deteriorate. This configuration can lead the model to memorize the inherent noise in the training data. The identification of this kind of overfitting can be carried out by examining the learning curves of the model. These curves provide a visualization of the evolution of model performance on training and validation sets over time. In case of overfitting related to the number of epochs, while performance on the training set may continue to improve (the MSE decrease), performance on the validation set may degrade. In our case, the model's learning is followed using callbacks in order to overcome this type of problem (see [Section 5.4.4.3](#)). [Figure 10-8](#) shows the learning curves for a latent space of 128. We observe that there is no degradation of the results on the validation data, even if the performances stabilize starting from 500 epochs.



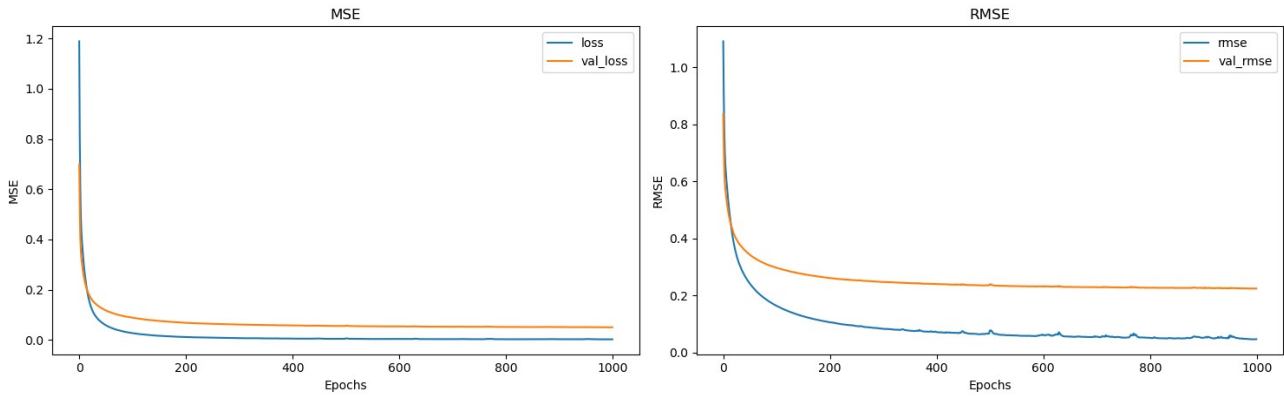


Figure 10-8: Learning curves for the model n°6

On the other hand, the second type of overfitting is related to the model architecture and can be caused by excessive complexity compared to the amount of data available. If the neural network has a large number of neurons for example, it can adjust too closely to the training data, even if they contain noise. Overfitting linked to the model architecture could be manifested by continuous improvement on the training set, but with stagnation or deterioration of performance on the validation set as a function of the model’s complexity. By observing these trends (see Figure 10-9), it can be stated that our best model, with a feature size of 128 and with the best performance, is not overfitting on the training data and that our results are accurate.

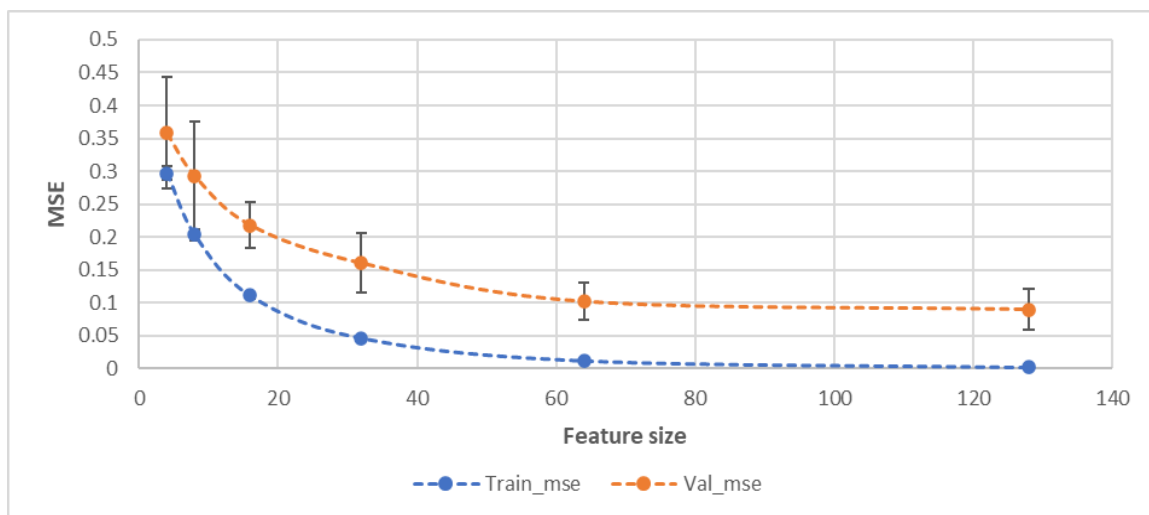
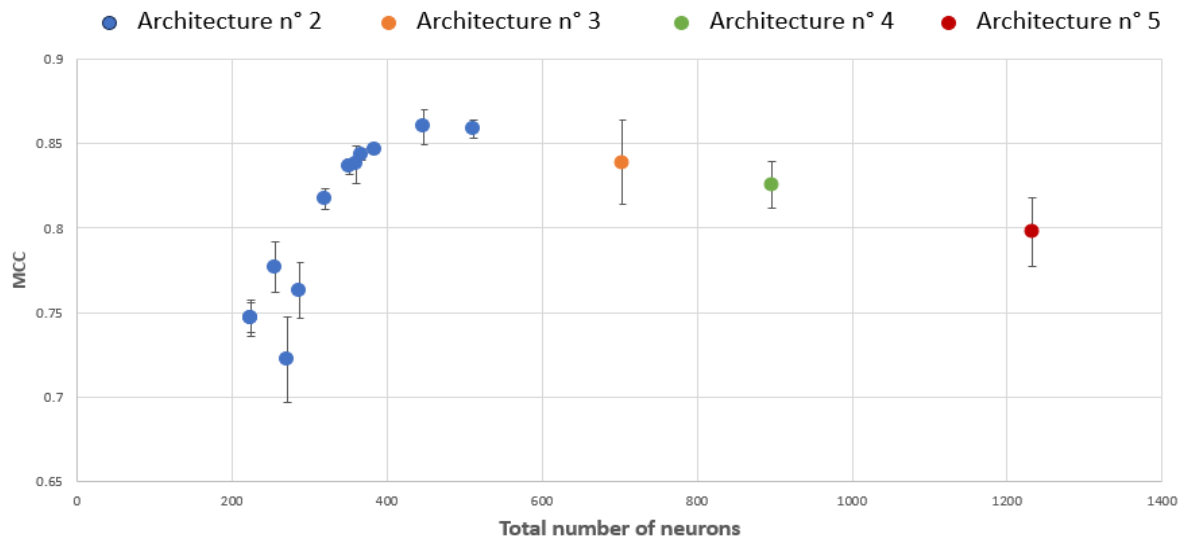


Figure 10-9: Tests of overfitting based on model's complexity

### 10.2.2. Testing different architectures

Figure 10-10 illustrates the evolution of the MCC for the various models described in Table 10, distinguishing the different architectures. The MCC results are obtained using the precision-recall (PR) curve approach, and a similar trend is observed for the F1 score results. It is notable that, beyond the use of three layers (Architecture n° 2), the model performances

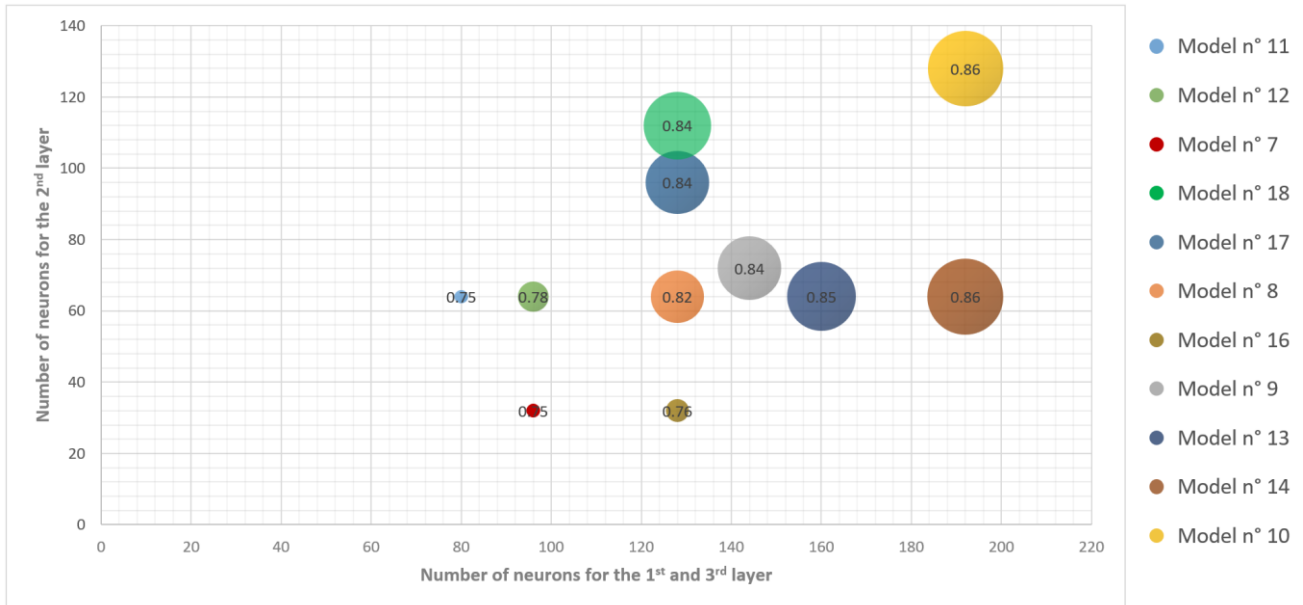
exhibit degradation and instability, with a significant increase in the standard deviation between runs. This observation leads to conclude that this specific type of architecture is not adapted to our application context. There are several reasons for these results, including the excessive complexity of the model in relation to the amount of data available, potentially leading to overfitting or increased sensitivity to random variations in training data. Thus, we will rather focus on the analysis of architecture models with 3 hidden layers.



**Figure 10-10: MCC results of Deep-AE architectures following the total number of neurons**

**Figure 10-11** provides an analysis of architectures n°2, highlighting the MCC as a function of the number of neurons in hidden layers and the size of the latent code. Each point on the graph represents a unique combination of these two parameters, with the MCC value indicated by the label and the size of each point visually reflecting this value. The aim of this visualization is to complement the observation made in

**Figure 10-10**, which indicates an improvement in performance as the total number of neurons increases. It thus becomes essential to observe whether there is a predominant direction between the size of the latent code and that of the adjacent hidden layers, in order to determine which parameter has the greatest impact on model performance. The aim of this analysis is to gain a more nuanced understanding of the dynamics between these two key components of the architecture.

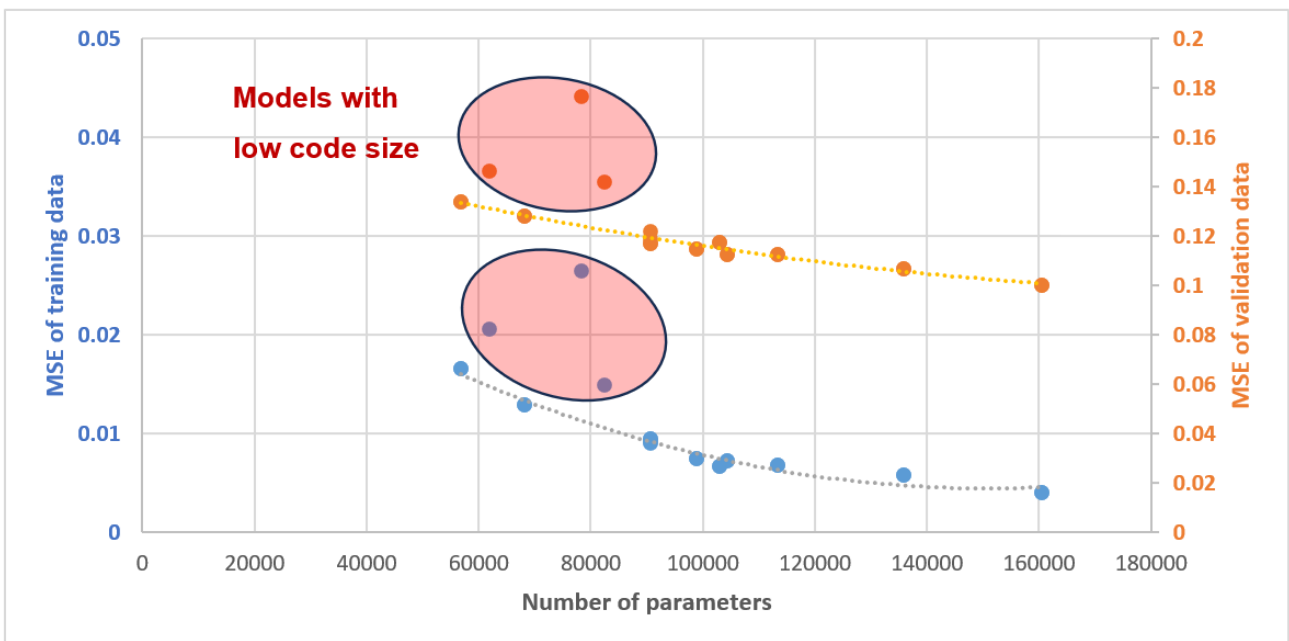


**Figure 10-11: MCC results of AE models with 3 hidden layers**

We observe that increasing the number of neurons in the hidden layers is generally associated with improved performance, as evidenced by the models 11, 12, 8, 13 and 14, which display MCCs ranging from 0.75 to 0.86. This suggests that the addition of neurons in the hidden layers contributes positively to the model's ability to reconstruct input data. In a similar way, we observe an improvement in performance with increasing latent code size. For example, the models 16, 8 and 17 have MCCs of 0.76, 0.82 and 0.84 respectively, demonstrating a positive relationship between latent code size and reconstruction quality.

Trend analysis considering a fixed code size (64) and a fixed hidden layer size (128) reveals significant observations. The stronger slope associated with a fixed hidden layer size suggests a potentially stronger impact on model performance. However, this assessment needs to be qualified by the fact that each neuron added to the first hidden layer is duplicated in the third layer. From this perspective, the slope linked to the size of the latent code becomes more important. Moreover, it remains crucial to consider both parameters simultaneously, as illustrated by the comparison of models 7 and 11, which show equivalent performance for an identical total number of neurons (224). It's important to stress that the generalizability of these findings is limited by the small number of evaluations. However, above a certain threshold, increasing the number of neurons does not guarantee improved results, as illustrated by models 17 vs. 18 and 14 vs. 10, where doubling the number of neurons in the code does not translate any improvement. This observation highlights the need to assess the risk of overfitting in such situations, where increasing the model's complexity does not translate into better performance.

Calculating the complexity of a model, here a deep-AE, is based on the number of network parameters. In the context of dense layers, each neuron is connected to all neurons in the previous and next layers. Thus, the number of parameters in a dense layer is determined by multiplying the number of neurons in the previous layer by the number of neurons in the current layer, plus the number of biases, one per neuron in the current layer. Generally, the complexity of the model is directly proportional to the total number of parameters, which is the sum of the parameters of all the layers. Adding additional layers or increasing the number of neurons in each layer increases the complexity of the model, which can lead to higher representational capacity, but also increases the risk of overfitting. For an equivalent total number of neurons, models can have different complexities: this is the case for models 7 and 11 for example. **Figure 10-12** shows that there is no overfitting in our case, as the loss function continues to decrease on both training and validation data, with a less significant curve for the latter. It should be noted that there are various points that deviate from the overall trend. These include models 7, 15 and 16, which have a reduced latent space size of no more than 32 neurons.



**Figure 10-12: Learning curves depending on the model's complexity**

After excluding models with small code sizes and focusing specifically on those with an MCC greater than 0.8, we aim to assess the additional impact of hidden layers compared with the basic architecture evaluated previously ([Section 10.2.1](#)). To do this, a comparison will be made between models with the same code size, enabling the contribution of hidden layers to be specifically isolated. Thus, we will examine model 5 in relation to model 8, as well as model 6 in relation to model 10, in order to quantify and analyze the added value provided by hidden layers in these specific configurations.

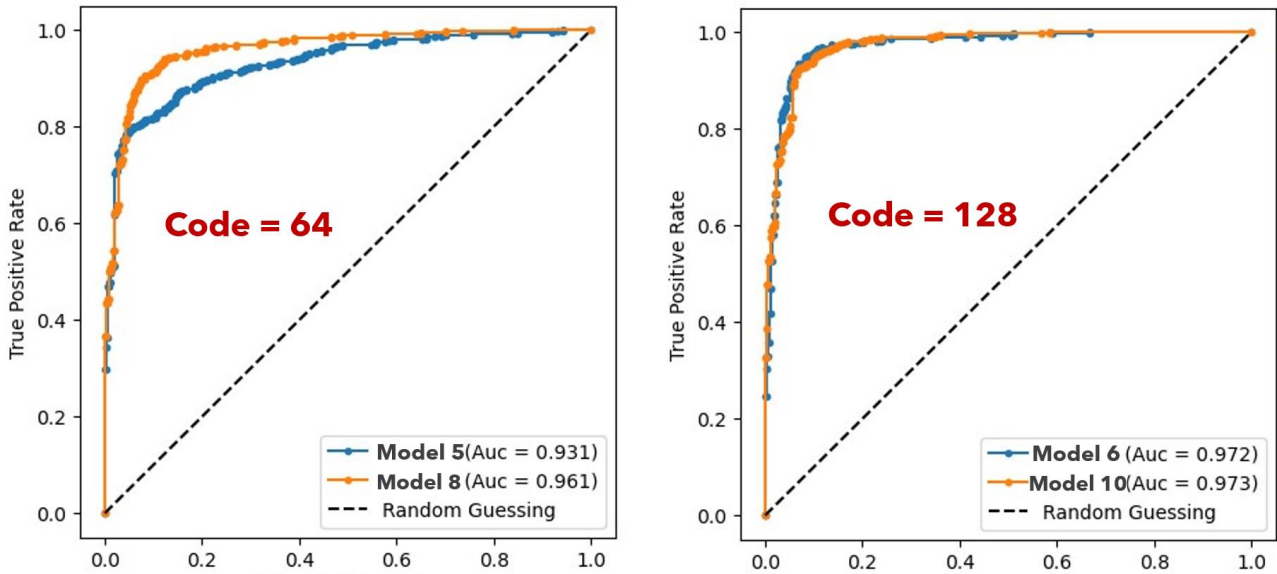


Figure 10-13: ROC curves comparing models with the same code size but different architectures

A closer look at the ROC curves reveals a significant improvement for models with a code dimension equal to 64, while this improvement is virtually absent for those with a code dimension equal to 128 (see Figure 10-13). This observation is corroborated by Figure 10-14, where an improvement in metrics such as F1 score and MCC is clearly perceptible on one side, but not on the other. In this context, the legitimate question that arises is whether models 6 and 10 are equivalent, especially as tests have been taken to exclude the possibility of an overfitting configuration.

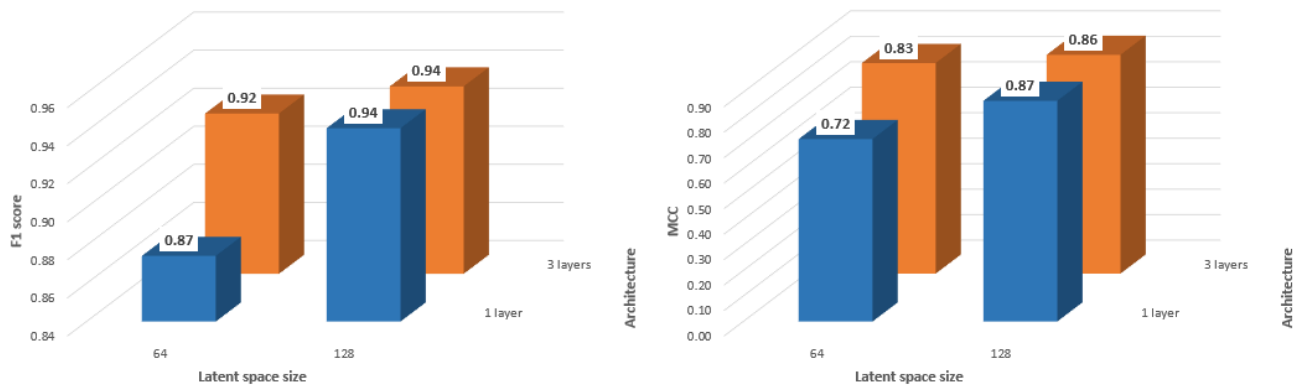
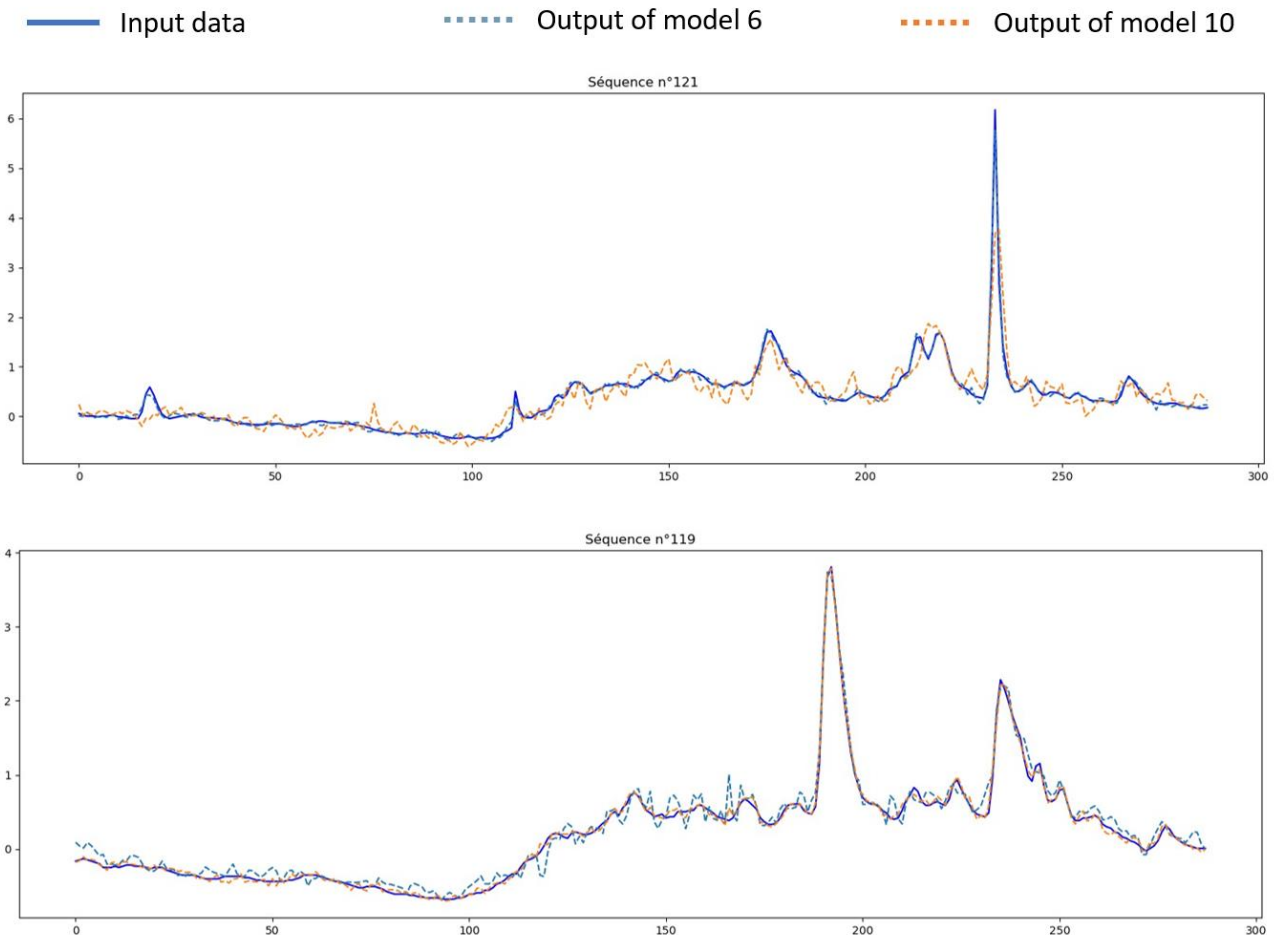


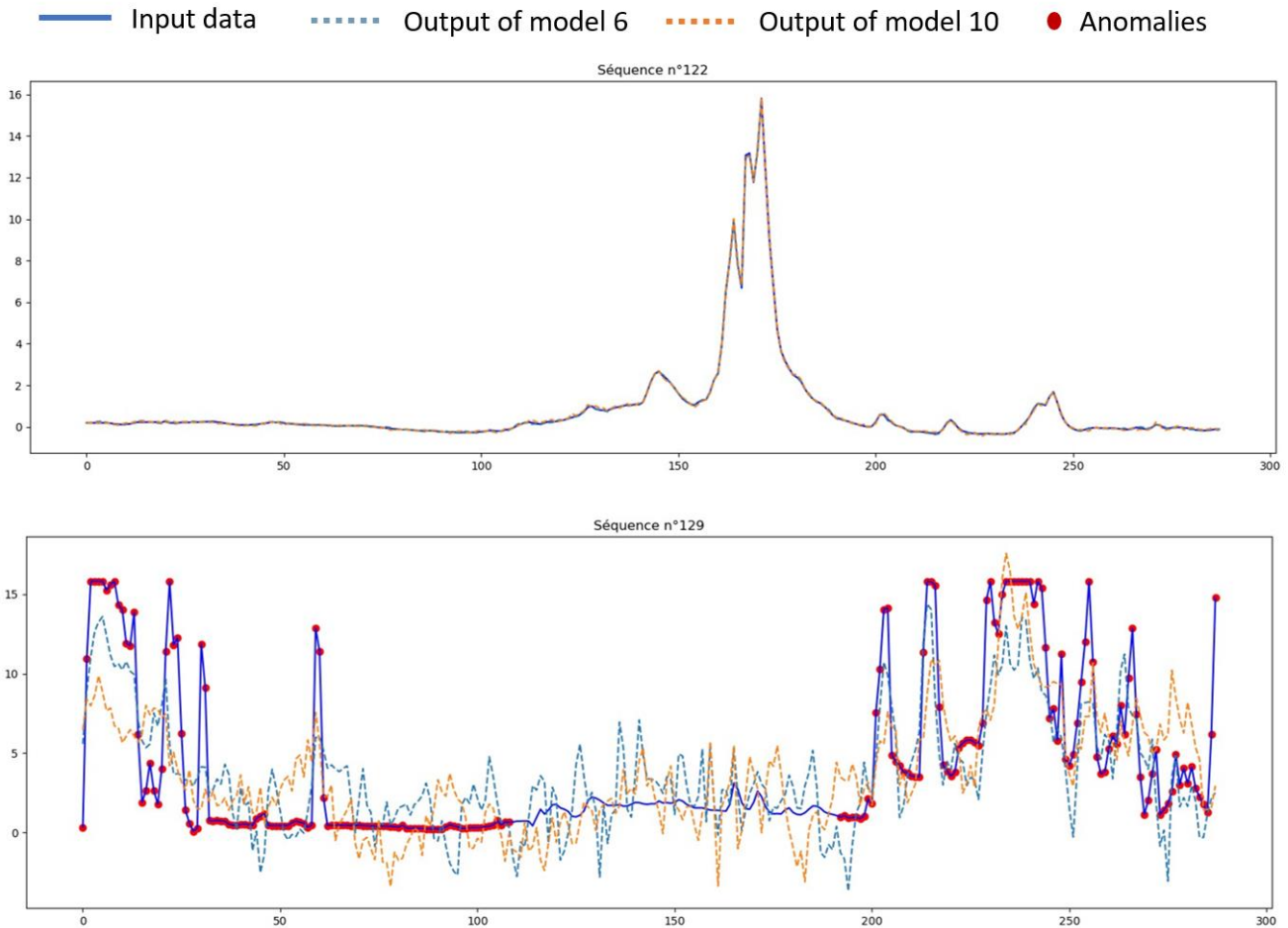
Figure 10-14: Performance metrics for models with the same code size but different architectures

To compare Models 6 and 10, we examine the reconstruction of the sequences generated by these models. Two illustrative examples are shown in **Figure 10-15**, where both models validate the input sequence but produce distinct reconstructions. It is also important to note that neither model generates noise systematically. This feature may be present in the reconstructions of both models independently.



**Figure 10-15: Reconstruction of valid sequence n°121 and valid sequence n°119 using the models 6 and 10**

However, specific sequences exist where the two models achieve a reconstruction in complete agreement. In addition, an analysis of invalid sequences reveals that the two models make errors separately and are mistaken differently (see **Figure 10-16**). Hence, it becomes clear that the two models do not produce exactly the same output for a given input sequence. So, although these models show similar performance, it is clear that they focus on different features, underlining the diversity in their approaches to sequence reconstruction.



**Figure 10-16: Reconstruction of valid sequence n°122 and invalid sequence n°129 using the models 6 and 10**

To sum up, following the various comparison tests between models and the assessment of the risks of over-fitting according to model complexity, it seems superfluous to exceed 3 hidden layers, given the limited number of examples and features available. Thus, the best-performing architectures are 1 and 2 (see [Table 10](#)), excluding models with a reduced code size of 32 or less. Best models in absolute terms include Model 6, characterized by a code consisting of just 128 neurons (subsequently referred to as **Model A**), as well as Models 10 and 14, which display equivalent performance with a 3-layers structure, where the first and third layers each comprise 192 neurons. In terms of code size, both sizes (64 and 128 neurons) deliver equivalent performance. Considering that the performance of a model is also linked to its complexity, which impacts training time, Model 14 is judged to be the second-best architecture (subsequently referred to as **Model B**).

### 10.2.3. Window size

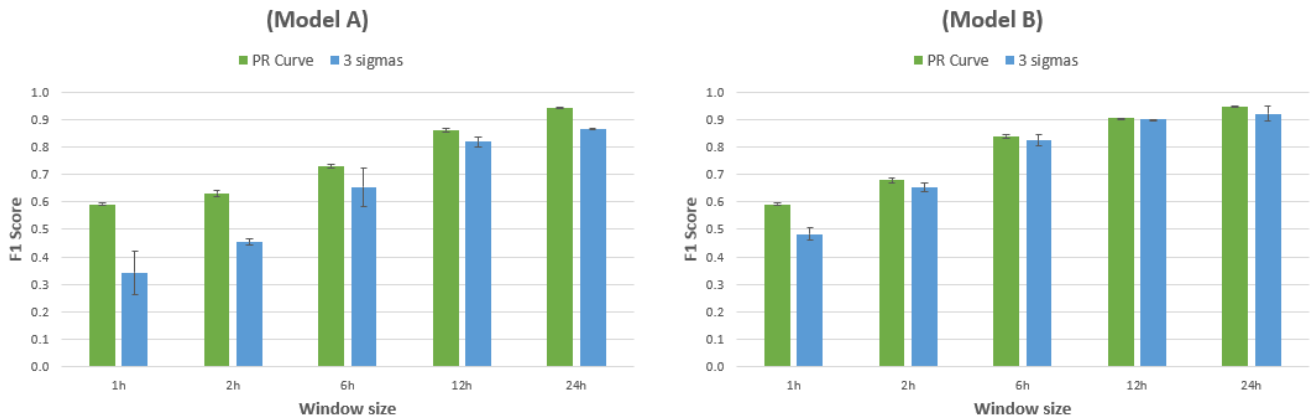
Since the start of the experiments, the size of the input sequences has been fixed at 24 hours in order to maintain an operational sequence characterized by well-structured dynamics, particularly under dry weather conditions with regular patterns. This sequence has also shown numerical promise in comparative evaluation with other models such as Matrix Profile and ResNet.

However, it is imperative to assess the sensitivity of the AE to this variable. For this, we will particularly consider shorter sequences. In fact, the use of excessively long sequences in dense models can generate problems linked to the **curse of dimensionality**. In general, it is recommended to have a number of samples significantly higher than the number of features. This favors better generalization of the model to new data, thus reducing the risk of overfitting. According to the literature, a rule of thumb suggests maintaining a ratio of 10 times more samples than features [239]. For classification problems, this ratio needs to be adjusted according to the number of classes.

However, it is important to note that this rule is not absolute. Determining the size of the database required must be the subject of specific tests, taking into account the model in place and overfitting evaluations. For some models, it may be possible to manage a large number of features with a consequent number of samples, depending on the nature of the problem and the quality of the data. Still, in our context, exceeding the 24-hour duration seems tricky given the limited number of samples available (1092). Thus, the tests on the window size mainly concern sequences with the following sizes: [1 hour - 2 hours - 6 hours - 12 hours and 24 hours].

Consequently, **Figure 10-17** presents classification results using the F1 score as a performance metric for different sequence sizes, using the two best models as stated in **Section 10.2.2**. The graphs provide a comparison between the two approaches, namely the PR curve and the 3-sigma rule.





**Figure 10-17: F1 score for different window sizes according to the trained model and the classification approach**

This result confirms equivalent performance between Model A and Model B when the window size is set to 24 hours. However, a discrepancy becomes apparent for smaller window sizes, with an outperformance of Model B. It should be noted that Model B stands out for its more stable results, characterized by a lower standard deviation between different runs. The use of Model B provides almost identical results between the two comparison approaches, which represents a significant advantage in the absence of manual validation to establish the PR curves.

As far as window sizes are concerned, performance improves as the input sequence size increases. Indeed, it is believed that the autoencoder requires large sequences to identify its distinctive features, thus promoting better sequence reconstruction. From a sequence size of 6 hours onwards, interesting results are observed, albeit slightly lower than the maximum obtained with 24-hour sequences. Moreover, given the constraint of invalidating a sequence as soon as a time step is invalidated, this sequence size may prove interesting as it is less constraining in absolute terms. Overall, the optimum sequence size lies between 6 hours and 24 hours, enabling us to reconcile operability while maximizing model performance.

Another assessment in this context is to analyze the sensitivity of the model to the value of the stride applied to the input data. So far, non-overlapping sequences have been considered, a choice that can be limiting, given that an invalid time step leads to the exclusion of 287 valid data that could contribute to model learning. This approach aims to increase the size of the input database by a mechanism similar to up-sampling, although the sequences are not identically duplicated, but rather partially superimposed.

Using model B with a window size set at  $w = 24$  hours, five different strides were tested, corresponding to the following ratios:  $w/2$ ,  $w/4$ ,  $w/8$ ,  $w/16$ , and  $w/32$ . [Figure 10-18](#) illustrates

the performance obtained using both the PR curve approach and the 3-sigma rule for classification. An apparent stability between the two approaches is observed, although simultaneously, a degradation of results is observed with increasing  $X$ , despite the effective increase in the volume of training data. Indeed, in certain situations, which seems to be the case here, data duplication can introduce redundancy, limiting the diversity of information relevant to learning. This redundancy compromises the autoencoder's ability to extract significant features, thus explaining the gradual degradation in performance observed.

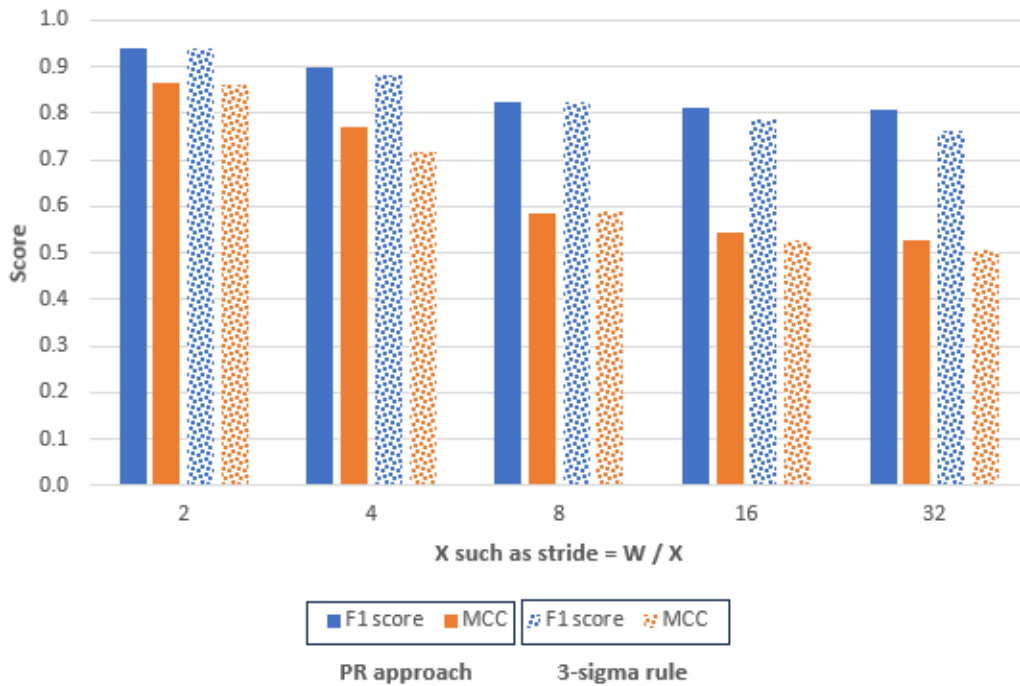
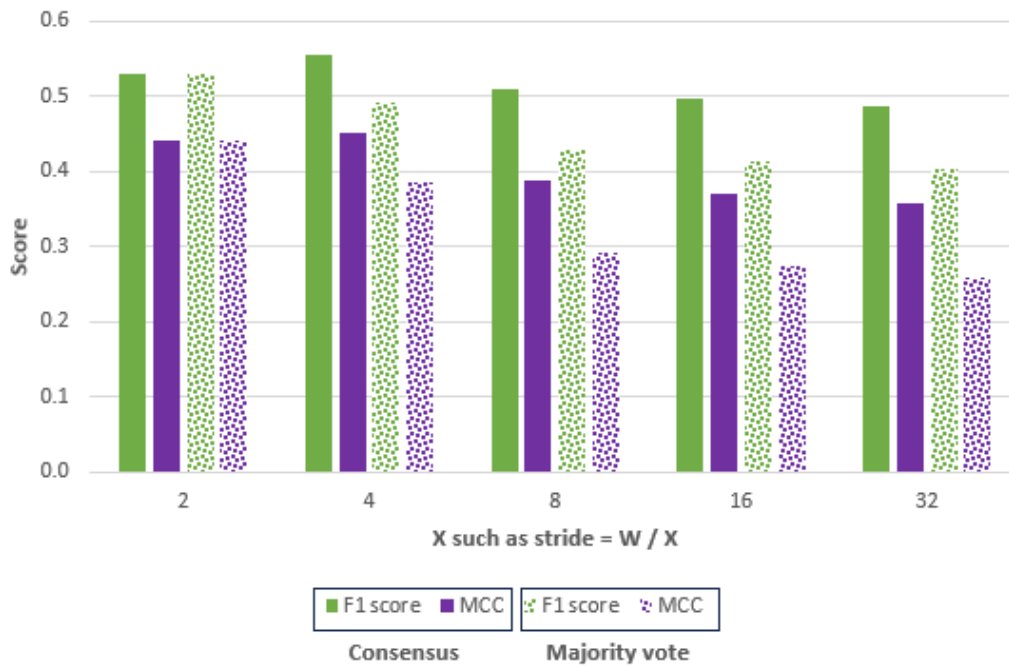


Figure 10-18: Performance metrics for different strides using Model B

However, the use of the overlay creates an additional problem related to the classification at the scale of each point. Indeed, each point can have  $X$  labels corresponding to the  $X$  sequences to which it belongs. In the case of divergent labels, the question arises as to the allocation of the appropriate label for the measure. Two approaches were examined: consensus and majority voting. Testing reveals an overall performance degradation, with a slight optimum achieved using the  $w/4$  stride. The use of consensus focuses on consolidated anomalies, eliminating any ambiguity of interpretation.



**Figure 10-19: Performance metrics at the measure scale for different strides using Model B**

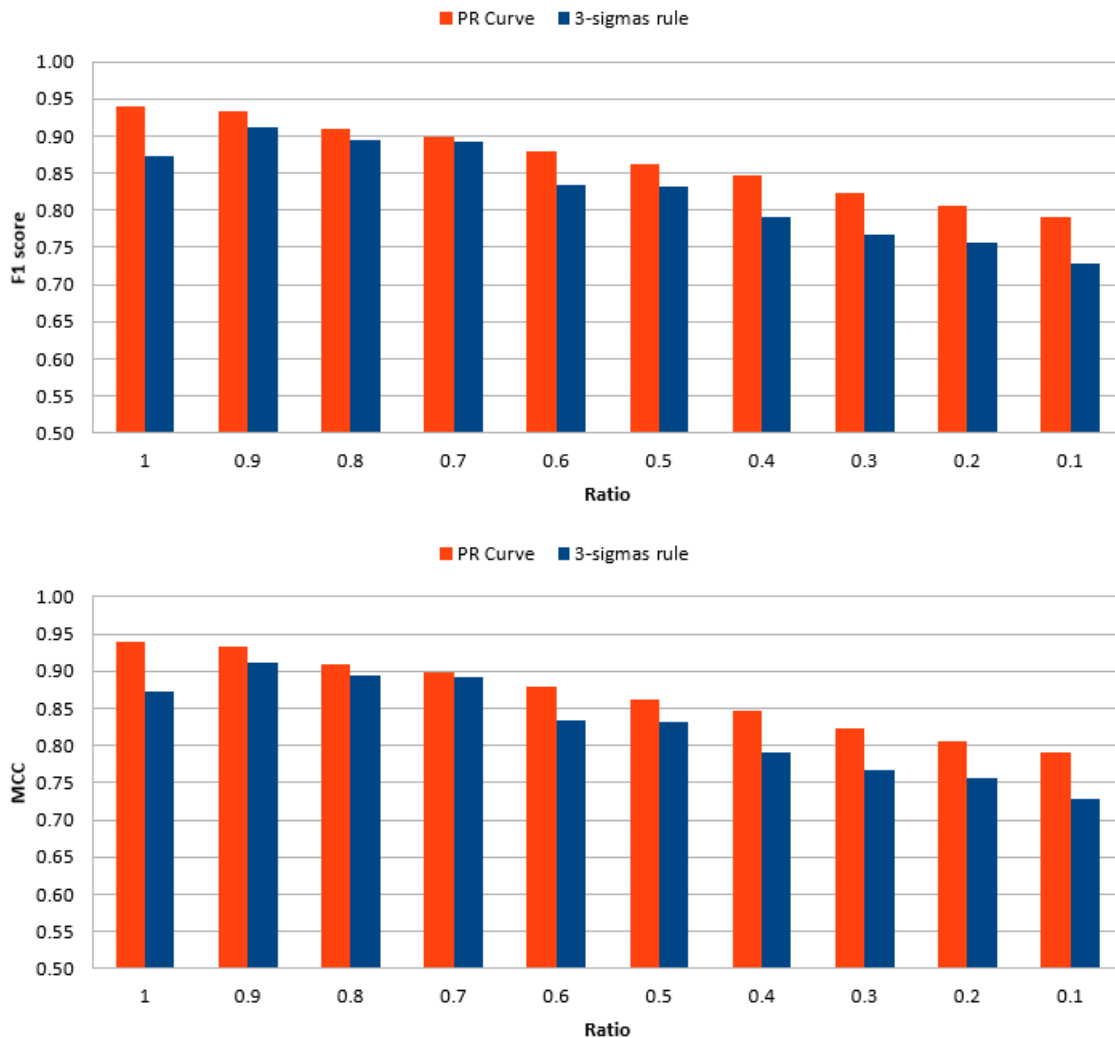
### 10.3. How can we improve the model's performance ?

Having defined the pre-processing steps (standardization) and identified the best architecture (Model A and Model B), with a non-overlapping 24-hour input window size, it's opportune at this stage to consider the potential improvement in results by adjusting the classification rules. Two issues deserve particular consideration: the relevance of the number 3 in the 3-sigma rule, and the judiciousness of the rule invalidating a sequence from an invalid time step. It is also possible to explore improvement approaches by merging the strengths of the best models and implementing pre-validation processes.

#### 10.3.1. Increasing database size

One of the first ways of improving our results is to increase the size of our database. Preliminary tests have revealed that the limitation of our database is particularly apparent when only 100% valid sequences are considered. Up-sampling approaches, using a different stride, do not lead to improved results and cause additional problems related to the final labels of point measurements. So, the question is, how much should we enhance our database, and what performance can we expect in this case? We have shown that the model with a single hidden layer, representing the code, reaches saturation, and therefore increasing the data will bring no benefit. However, the performance of the 3-layers model showed satisfactory results. To assess the impact of increasing the size of the database on this model, we adopt the same technique as presented in [Section 10.1.3](#), consisting of progressively reducing the size of the

training database and evaluating the overall F1 score at each stage. Specifically, for each sub-database resulting from the progressive reduction in the size of the training set, we repeat the tests several times. **Figure 10-20** illustrates the results obtained.



**Figure 10-20: Performance metrics according to the database size**

In contrast to the previous case, we do not observe a plateau, indicating the absence of training saturation. In fact, the increase in the database contributes to the improvement in results. Regarding the evolution of the performance metrics as a function of the size of the input data, we estimate that with a ratio of 1.30 (i.e. increasing the database by 30%), we could achieve maximum performance with an MCC of 0.90 and an F1 score of 0.96. However, it is crucial to ensure that the added data is relevant to guarantee such improvement.

### 10.3.2. Adjusting classification rules

Once the autoencoder model has been trained on sequences considered 100% valid, in accordance with its learning principle, the sequence classification phase is based on the

imposition of a threshold on the MSE of the reconstruction. This threshold determines from which MSE level a sequence is considered invalid. So far, two approaches have been explored for this purpose: the rule of 3-sigmas and the approach based on the PR curve. However, the objective here is to improve the latter and also to explore other potential methods for this classification.

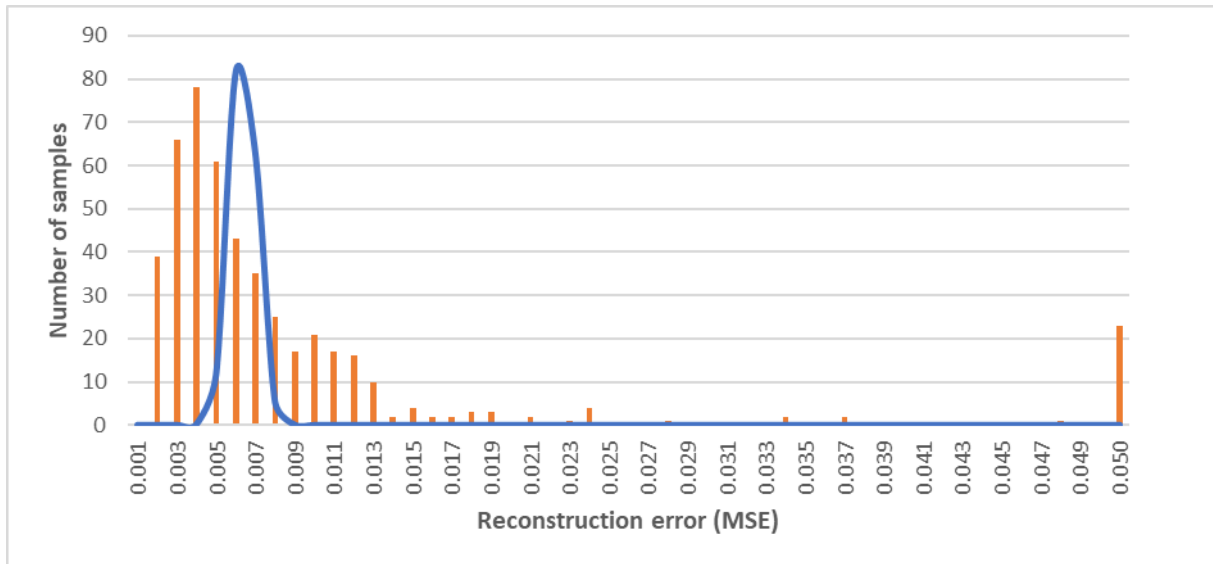
### 10.3.2.1. Three-sigma rule

The first method involves thresholding the MSE by considering a statistical basis determined by the mean plus three times the standard deviation. However, analyses show that this approach, based on the 3-sigma rule, does not always enable maximum model performance to be achieved. **Table 49** shows the x-sigma required to obtain the best performance for the two best models. It can be seen that the threshold defined by the 3-sigma rule is exceeded. By comparison, the threshold of the PR curve is higher. Consequently, the approach based on the 3-sigma rule may lead to excessive invalidation at the expense of satisfactory recall.

**Table 49: Evaluation of the x sigma required to achieve the best performances**

Architecture	mean MSE	std MSE	3-sigmas	threshold PR	x-sigma ?
Model A	2.34E-03	2.80E-04	0.003	0.007	17.00
Model B	5.88E-03	6.70E-04	0.008	0.015	13.89

The practical application of the 3-sigma rule approach raises important considerations in the context of the inherent difficulty of establishing the optimal value of x in an unsupervised approach. Indeed, the 3-sigma rule is widely used in classical statistical contexts where data follow a known normal distribution. However, in our context and in an unsupervised approach, the characteristics of the data may be less well defined. **Figure 10-21** represents the distribution of MSE in training data, showing that we are moving away from a normal distribution.

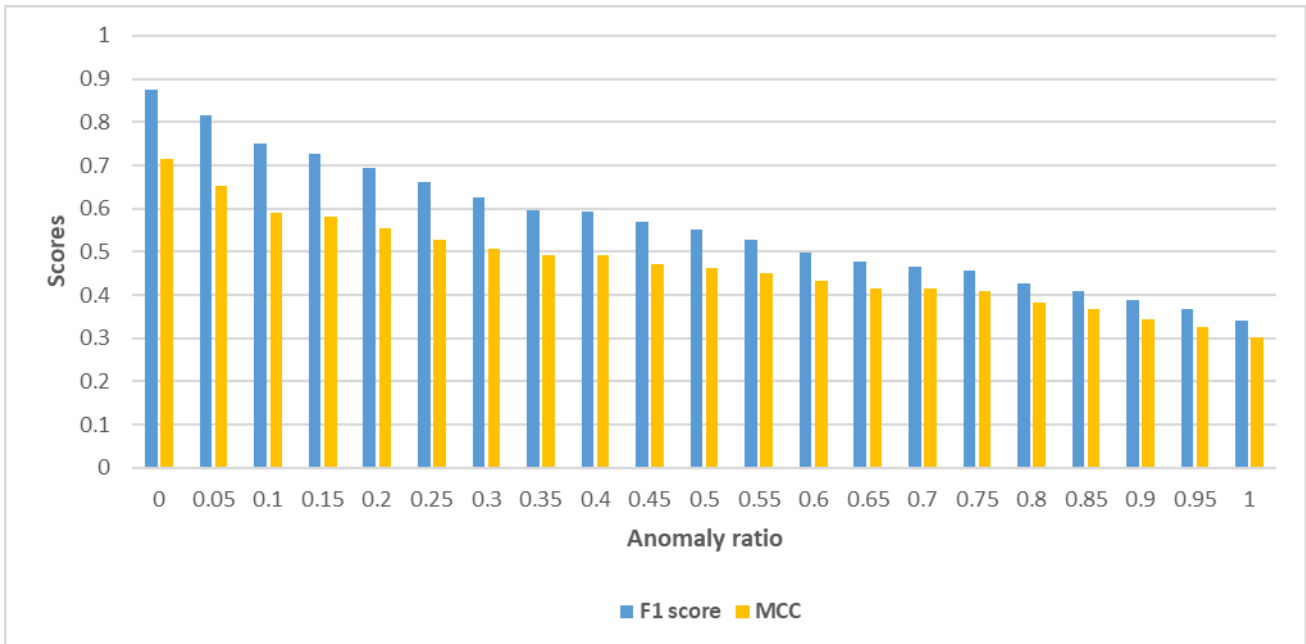


**Figure 10-21: Histogram of MSE on training data samples (in red) and the corresponding normal distribution (in blue) using Model B**

Thus, one of the difficulties lies in the intrinsically subjective nature of the choice of  $x$ . This subjectivity can be exacerbated by the absence of prior data labeling, limiting the ability to formally determine the appropriate threshold. As a result, the practical implementation of the 3-sigma rule in a semi-supervised setting can prove very challenging. Investigative options could involve adjusting the threshold using a more suitable distribution, such as a lognormal distribution, with a probability of exceeding the MSE of 1% for example on valid data, but due to time constraints this possibility was not examined.

### 10.3.2.2. Precision-Recall curve

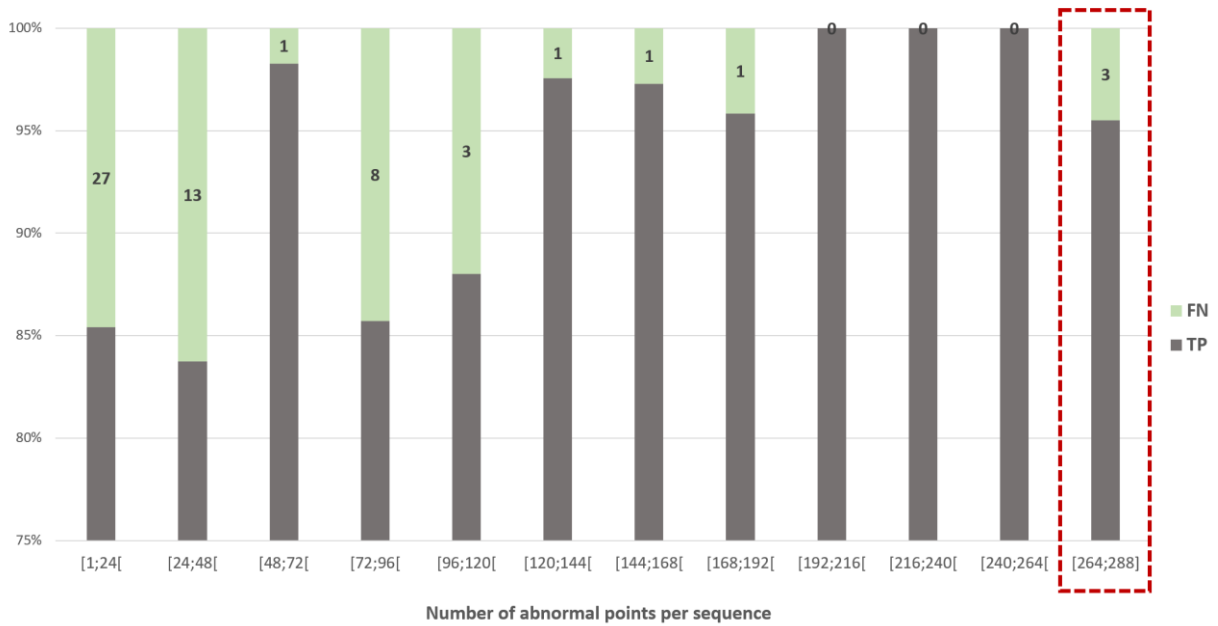
The second approach is based on the construction of the PR curve. However, to develop this curve, it is imperative to have a baseline, which compromises the semi-supervised nature of the AI model. In addition, it is necessary to convert the manual label developed at the time step scale into a sequence scale. So far, we have considered that a sequence is considered invalid as soon as a time step is, which is a significant constraint. To remedy this, we sought to evaluate the sensitivity of the model's performance to this classification threshold in post-processing. In other words, learning continues to be done with fully valid sequences, and it is the classification phase that is analyzed here. **Figure 10-22** illustrates the results, where the x-axis represents the relative threshold, that is, a sequence is considered invalid if  $(x\text{-axis} * \text{window size})$  points are invalid. The model used here is Model 5, chosen due to computation time constraints, given that it has demonstrated satisfactory performance.



**Figure 10-22: Performance metrics according to the classification threshold as an anomaly ratio per sequence. 0 refers to an invalid sequence from one invalid time step**

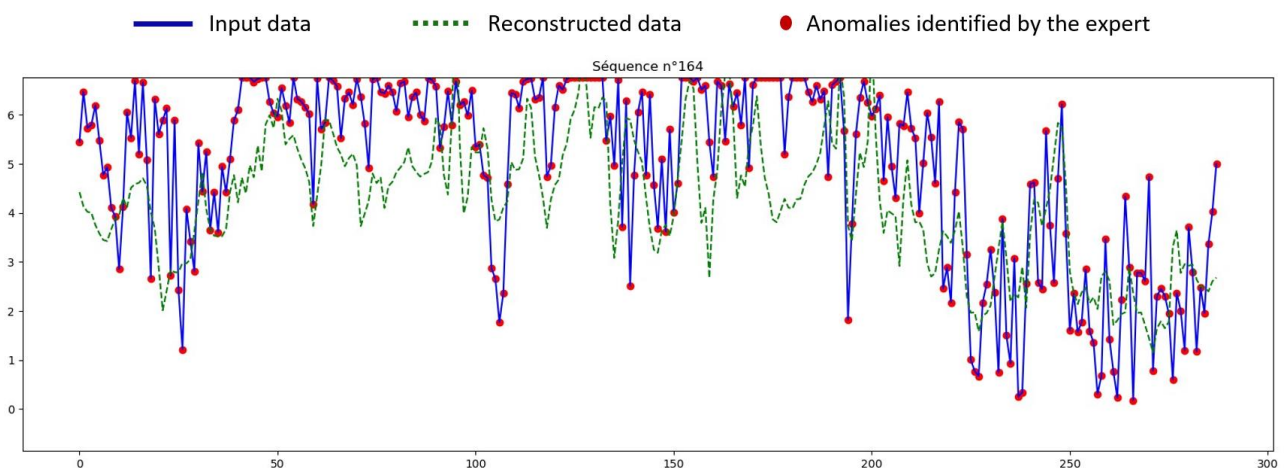
As the tolerance threshold for invalidating a sequence increases, the model's performance measures decrease, indicating a decline in the accuracy of its classification of sequences as invalid. Optimal model performance is observed with the lowest threshold, where every sequence is invalidated as soon as a time step is considered abnormal. Conversely, the least satisfactory performance is obtained when only sequences that are 100% invalid are rejected. Indeed, the trained model has not learned tolerance, so invalidating sequences containing a few abnormal time steps generates false negatives and reduces model recall. Up to a threshold of 0.05, performance is considered satisfactory, enabling a precision of 0.75 (accepting 25% of false alarms) and a recall of 0.87 (omitting 13% of abnormal sequences), while limiting errors to those exceeding one hour. False alarms require the inclusion of a margin of error and/or the further intervention of an expert. However, false negatives are drowned into valid data, underlining the importance of characterizing them. With this in mind, a fine analysis is carried out to diagnose sequences misclassified by the model.

**Figure 10-23** represents the number of validated sequences among the abnormal sequences identified by the manual validation in relation to the number of invalid points per sequence. It is observed that the majority of errors occur for sequences with a low number of invalid points. More precisely, 70% of errors concern sequences with an anomaly rate of less than 17%, equivalent to 4 hours of anomalies.



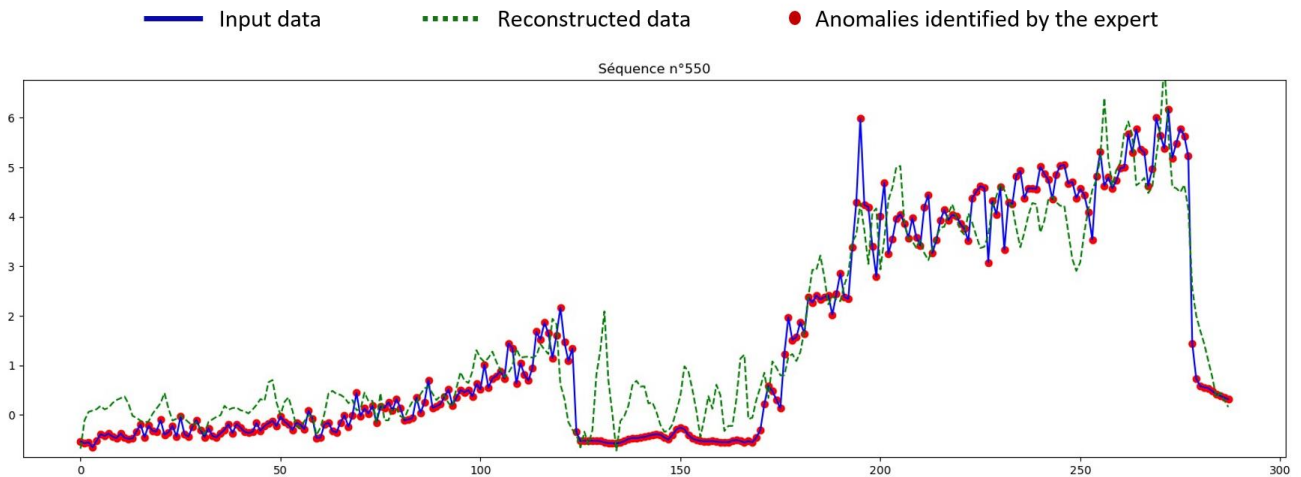
**Figure 10-23: Ratio of false negatives for different anomaly ratios such as identified by model 5 and using a classification threshold of 0.04**

However, a paradoxical situation arises in the case of sequences with an anomaly rate in excess of 90%, i.e. 23 hours of invalidity (framed in red in [Figure 10-23](#)). Of the 68 sequences in this category, the model manages to classify 65 correctly. These anomalies represent particularly complex anomaly detection challenges, as shown in [Figure 10-24](#) and [Figure 10-25](#), which presents two main examples, including very noisy saturations and drifts that are difficult to identify using simple rules, thus requiring the intervention of an expert.



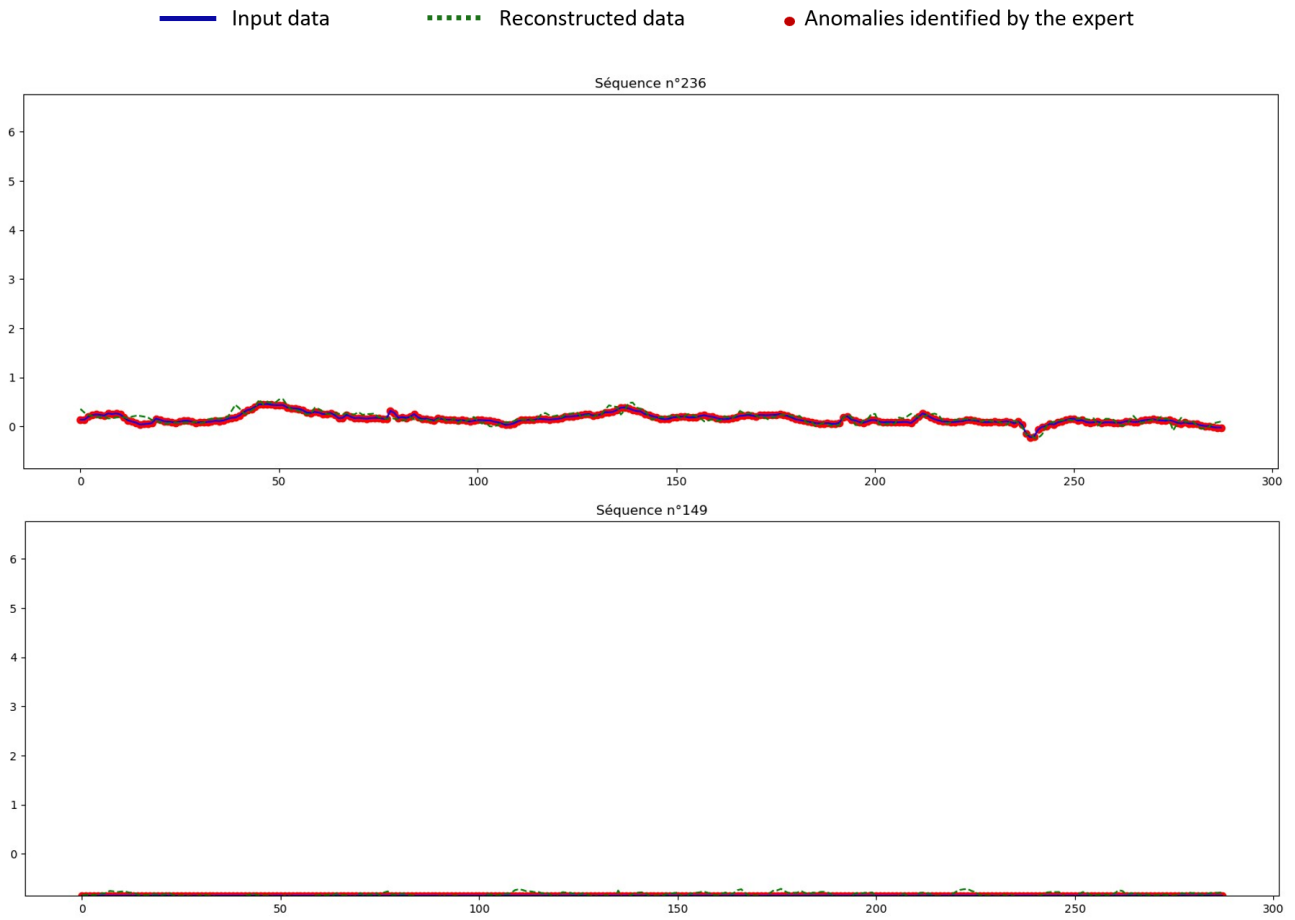
**Figure 10-24: Noisy saturation sequence invalidated by the expert and the AE.**





**Figure 10-25: Drift sequence invalidated by the expert and the AE**

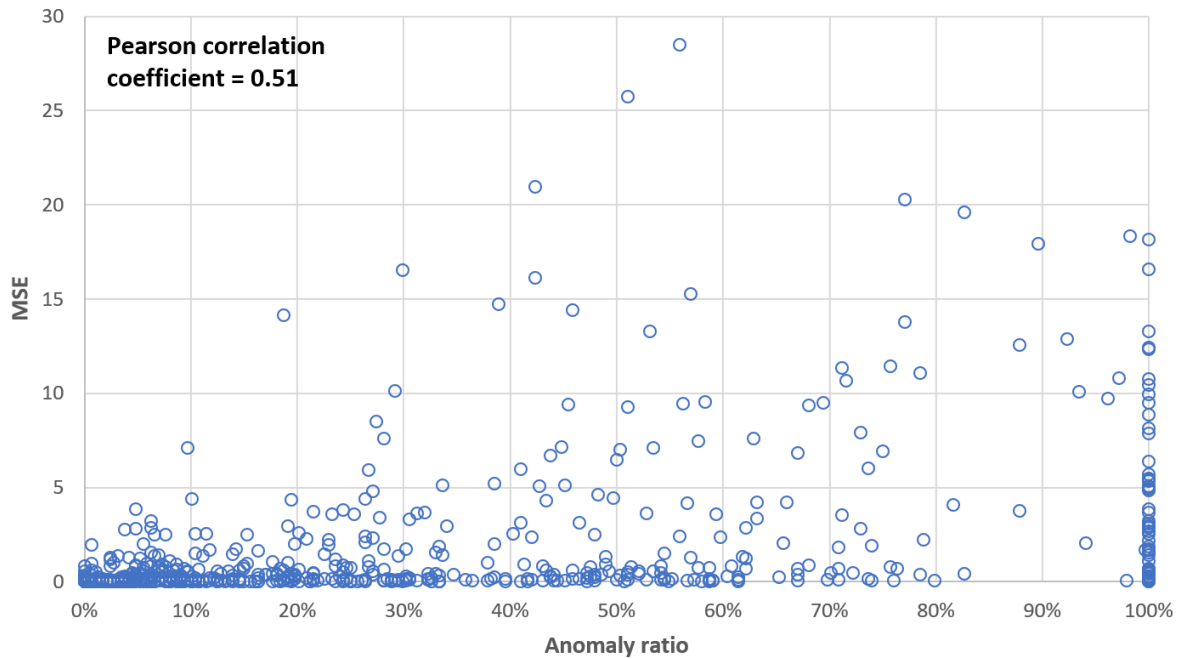
On the other hand, **Figure 10-26** shows the error cases where the model validated sequences considered completely invalid by the expert. These cases include two examples of null data (initially missing and replaced by 0). Despite the simplicity of this error, AE, like the other ML models tested, fails to identify this type of anomaly. The introduction of a pre-validation step would indeed make it easy to isolate such sequences. The second case concerns a sequence with very little variability, which the model manages to reconstruct satisfactorily, thereby validating it. However, a retrospective analysis of the pattern of this sequence reveals that it could have been validated by the expert, since it doesn't show any abnormal patterns. But in fact, this is an earlier sequence of two days of defects and the expert invalidated it for the sake of precise defect delimitation.



**Figure 10-26: Invalid sequences according to the expert, validated by the AE model**

### 10.3.2.3. Correlation between MSE and anomaly

An alternative approach is to examine the relationship between the MSE of the reconstruction and the anomaly rate per sequence. Analysis of the best model results (Model B) reveals weak correlations, measured by Pearson's coefficient, between these two variables. The scatterplot associated with this relationship is notably dispersed, making it difficult to establish a significant relationship between anomaly rate and MSE (see [Figure 10-27](#)).



**Figure 10-27: Correlation between reconstruction error (MSE) and anomaly rate per sequence**

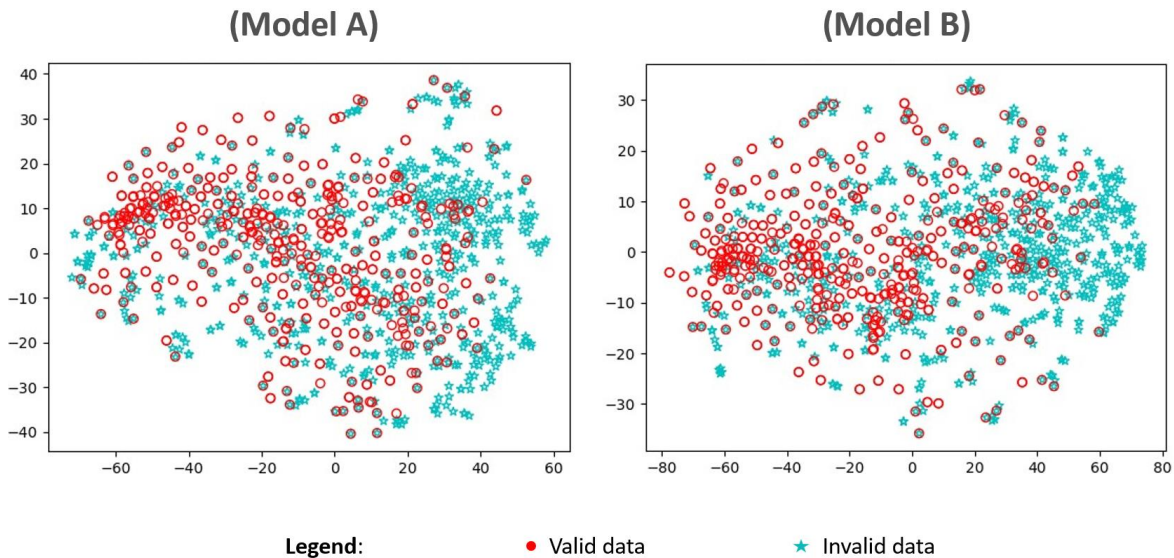
Another possibility would be to directly use the output of the AE, i.e. the reconstruction, and calculate the deviation from the step-by-step measurement. This would make it possible to assess whether the largest deviations are correlated with anomalies. For this purpose, the biserial point is used to measure the correlation between a continuous variable and a binary variable, here representing the validity or invalidity of the datapoint. To minimize the influence of extreme values, a normalized MSE is also calculated. However, the results obtained are relatively low. Consequently, it is concluded that there is no correlation between MSE and anomalies.

### 10.3.3. Ensemble model using the best architectures

The objective now is to compare the two best performing models in order to explore opportunities to combine them and take advantage of their respective advantages.

By comparing the t-SNE visualizations of the codes generated by the two autoencoders (Model A and Model B), we can obtain indications of the similarity of the latent representations (codes) produced by these models. Observations of the graphical representations (see [Figure 10-28](#)) reveal differences in the t-SNE visualizations of the two codes, indicating that the two models capture distinct structures within the input data. Although the t-SNE shows some overlap between valid and invalid classes for both models, it is important to note that this does not necessarily imply a lack of informativeness of the latent codes, as confirmed by the performance metrics. It is plausible that the dimensions in which the classes overlap are not

of crucial importance for classification, and that the model has succeeded in learning discriminative features in other dimensions. The two dimensions shown in [Figure 10-28](#) are chosen rather in response to comprehensive visualization constraints, not necessarily reflecting the entire latent space learned by the models.



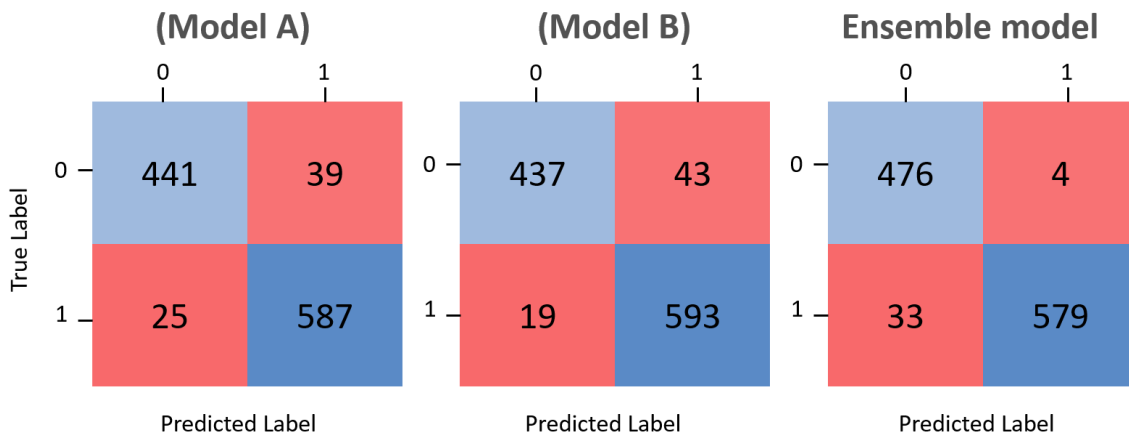
**Figure 10-28: t-SNE visualization of the code of Model A and Model B**

Now that we can state that the two best models show differences and learn distinct patterns, it would be interesting to explore the possibility of combining them to assess possible improvement. [Table 50](#) summarizes the results of models A and B using the two classification approaches, the PR curve and the 3-sigmas method. By adopting a strategy based on averaging the MSEs from the reconstruction of the two models, followed by applying the PR curve to find the optimal threshold, no significant improvement in results is observed. This approach therefore appears to have little relevance.

**Table 50: Ensemble model results using the average MSE combined to a PR curve approach**

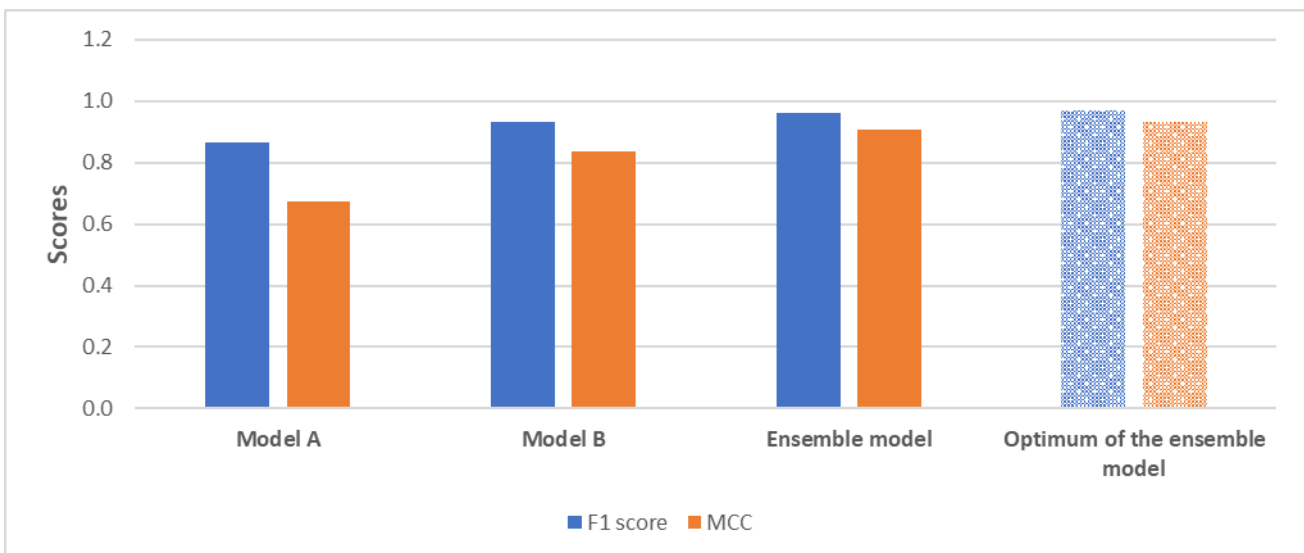
		Precision	Recall	F1 score	MCC
<b>Model A</b>	PR curve	0.938	0.959	0.948	0.881
	3-sigma	0.768	0.990	0.865	0.675
<b>Model B</b>	PR curve	0.932	0.969	0.950	0.885
	3-sigma	0.875	0.993	0.930	0.838
<b>Mean MSE</b>	PR curve	0.927	0.956	0.941	0.864

The second approach is to merge the results of the two classifications by consensus. **Figure 10-29** illustrates the evolution of the confusion matrices resulting from combining the results of the two models, using the PR curve as a reference. This shows a significant reduction in the number of false positives (FP) transformed into true negatives (TN), enabling a near-perfect precision of 0.99 to be achieved. Despite a slightly lower recall than the two models taken individually (0.95), the new model has a very advantageous MCC, equal to 0.93.



**Figure 10-29: Confusion matrices using consensus and based on the PR curve approach**

Analysis of the results using the 3-sigma rule reveals a considerable improvement, such that the overall model result approaches the optimum identified via the PR curve. The number of FNs remains very limited, and false positives represent only 6% of all anomalies detected.



**Figure 10-30: Results using the 3-sigmas rule (solid bars) compared to the results of the ensemble model using the PR approach (dotted bars)**

Consequently, combining the results of the two models shows promise, particularly through a consensus based on the 3-sigma rule. This approach maintains a semi-supervised approach, eliminating the need for a comparison baseline while achieving remarkable performance with a very limited number of false negatives (avoiding fault omitting) and 6% false alarms. The latter could be identified later by an expert.

#### 10.3.4. Implementing pre-validation approaches

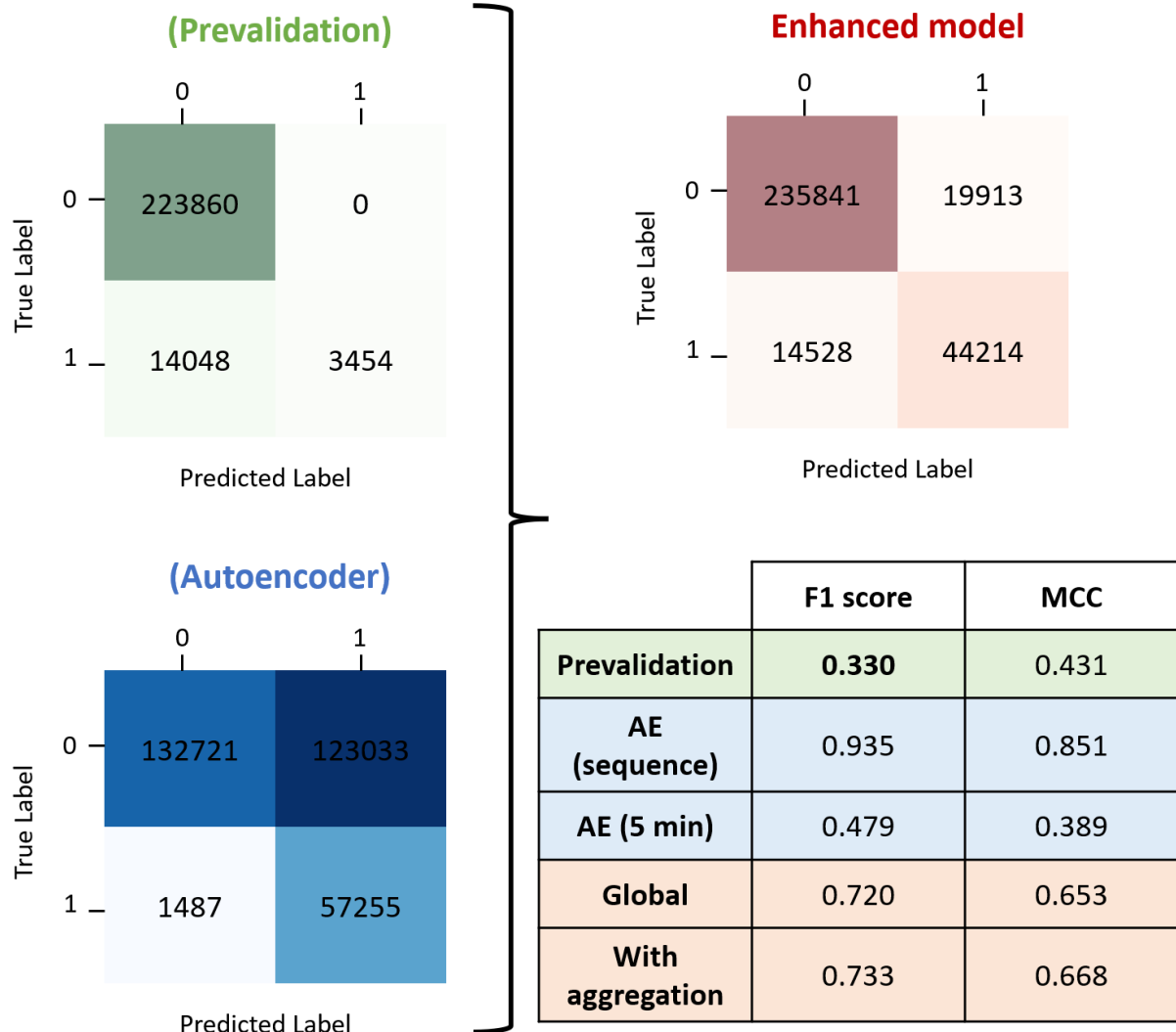
In this section, we implement a first stage of pre-validation for the model, following the example of the other models evaluated. The aim is to provide a base identical to that submitted to the expert. This step automatically invalidates trivial anomalies such as missing data, data outside the range of [1,1000], blocking or saturation. On the other hand, it automatically validates sequences that meet the redundancy criterion (see [Equation 3](#)). Unlike AE validation, which takes place at the sequence level, pre-validation with our approach takes place at the measurement time step level. The two approaches are combined a posteriori. Classification using a saved model is fast, taking less than a minute for 1092 sequences. We are therefore not seeking to optimize this calculation time by preselecting sequences in advance. Our main objective is to consolidate the final result.

Once the AE classification has been carried out, we transform the labels at a time step of 5 minutes, assigning the same label (that of the sequence) to all the points that make it up. Then, according to an order of priority, a final label is assigned to each measurement (see [Figure 10-31](#)).

```
if trivial anomaly: label = "invalid";
elif redundancy: label = "valid";
else label = output of AE model
```

**Figure 10-31: Synopsis of the classification task enhanced with pre-validation**

[Figure 10-32](#) presents all the results. The final result remains interesting, with an improvement from an F1 score of 48% on the scale of the time step to an F1 score of 73%.



**Figure 10-32: Enhanced model results combining the AE and a pre-validation phase**

However, in certain situations, the interest can be oriented towards validation on a more global scale, making it possible to determine practically whether a day is valid or not, without having to examine the different time steps. **Figure 10-33** illustrates the results obtained as a function of the threshold applied to the model outputs, compared to that applied to the expert's results. We find that the optimal performance is generally around the diagonal and with low thresholds. Indeed, the most satisfactory results are obtained by imposing invalidation from the slightest error. A threshold of 5% (equivalent to about 1h30) generates maximum performance, confirming that the AE model developed is not tolerant and reacts sensitively to the slightest error. This observation also suggests that it is difficult to fool the model, even by injecting subtle anomalies.

F1 score		Model threshold																		
Expert threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
0.05	0.90	0.87	0.84	0.80	0.75	0.72	0.67	0.62	0.58	0.51	0.45	0.39	0.37	0.32	0.28	0.23	0.20	0.14	0.09	0.05
0.1	0.83	0.86	0.85	0.83	0.79	0.76	0.72	0.67	0.63	0.56	0.50	0.44	0.41	0.37	0.32	0.27	0.23	0.17	0.10	0.06
0.15	0.79	0.82	0.85	0.84	0.81	0.78	0.74	0.70	0.66	0.59	0.53	0.47	0.44	0.39	0.35	0.28	0.24	0.18	0.11	0.07
0.2	0.74	0.78	0.81	0.83	0.82	0.80	0.76	0.73	0.69	0.63	0.56	0.50	0.47	0.42	0.37	0.30	0.26	0.19	0.12	0.08
0.25	0.69	0.74	0.78	0.80	0.81	0.81	0.78	0.74	0.71	0.65	0.58	0.52	0.49	0.43	0.38	0.32	0.28	0.21	0.13	0.09
0.3	0.63	0.68	0.72	0.75	0.80	0.81	0.79	0.77	0.75	0.69	0.62	0.56	0.53	0.48	0.42	0.36	0.32	0.24	0.15	0.10
0.35	0.58	0.64	0.67	0.72	0.77	0.79	0.79	0.78	0.75	0.71	0.65	0.58	0.55	0.50	0.45	0.38	0.34	0.26	0.17	0.11
0.4	0.57	0.62	0.65	0.70	0.76	0.78	0.78	0.79	0.76	0.71	0.65	0.60	0.57	0.52	0.46	0.40	0.36	0.27	0.18	0.11
0.45	0.52	0.57	0.61	0.65	0.71	0.75	0.77	0.79	0.77	0.72	0.66	0.62	0.59	0.55	0.49	0.42	0.38	0.29	0.19	0.13
0.5	0.47	0.52	0.55	0.59	0.66	0.69	0.72	0.76	0.78	0.75	0.69	0.65	0.62	0.59	0.53	0.46	0.42	0.33	0.22	0.14
0.55	0.42	0.46	0.49	0.53	0.59	0.63	0.66	0.70	0.72	0.69	0.70	0.66	0.64	0.62	0.56	0.50	0.46	0.37	0.25	0.17
0.6	0.37	0.41	0.44	0.47	0.53	0.57	0.61	0.65	0.67	0.69	0.71	0.70	0.68	0.66	0.60	0.54	0.52	0.42	0.29	0.19
0.65	0.34	0.37	0.40	0.44	0.49	0.52	0.57	0.62	0.65	0.68	0.72	0.72	0.70	0.69	0.63	0.56	0.55	0.46	0.32	0.21
0.7	0.31	0.34	0.37	0.40	0.45	0.48	0.52	0.57	0.61	0.64	0.71	0.73	0.72	0.72	0.65	0.59	0.59	0.49	0.34	0.23
0.75	0.27	0.30	0.32	0.35	0.40	0.43	0.48	0.53	0.57	0.63	0.69	0.73	0.72	0.73	0.69	0.64	0.65	0.55	0.39	0.27
0.8	0.24	0.27	0.29	0.31	0.36	0.39	0.43	0.48	0.52	0.57	0.63	0.68	0.68	0.70	0.68	0.67	0.68	0.58	0.43	0.30
0.85	0.23	0.26	0.28	0.30	0.34	0.38	0.41	0.46	0.50	0.55	0.62	0.66	0.67	0.69	0.67	0.65	0.66	0.58	0.45	0.31
0.9	0.22	0.25	0.27	0.29	0.33	0.36	0.40	0.45	0.49	0.53	0.60	0.64	0.64	0.66	0.64	0.62	0.63	0.58	0.44	0.32
0.95	0.21	0.24	0.26	0.28	0.32	0.35	0.38	0.43	0.47	0.51	0.58	0.62	0.62	0.64	0.61	0.62	0.63	0.60	0.45	0.33
1	0.20	0.22	0.24	0.26	0.30	0.33	0.36	0.40	0.44	0.48	0.54	0.58	0.58	0.59	0.57	0.58	0.60	0.57	0.46	0.36

Figure 10-33: Heatmap of the F1 score according to the classification threshold applied to the expert 5-minutes scale validation and the results of the enhanced model



## 10.4. Generalization to other sites

### 10.4.1. Direct evaluation of the best models

The aim of this phase is to directly exploit the two best models previously saved and evaluate them on turbidity data from other sites, without any prior adaptation, while maintaining the same pre-processing steps. These steps include the use of a 24-hour time window and data standardization.

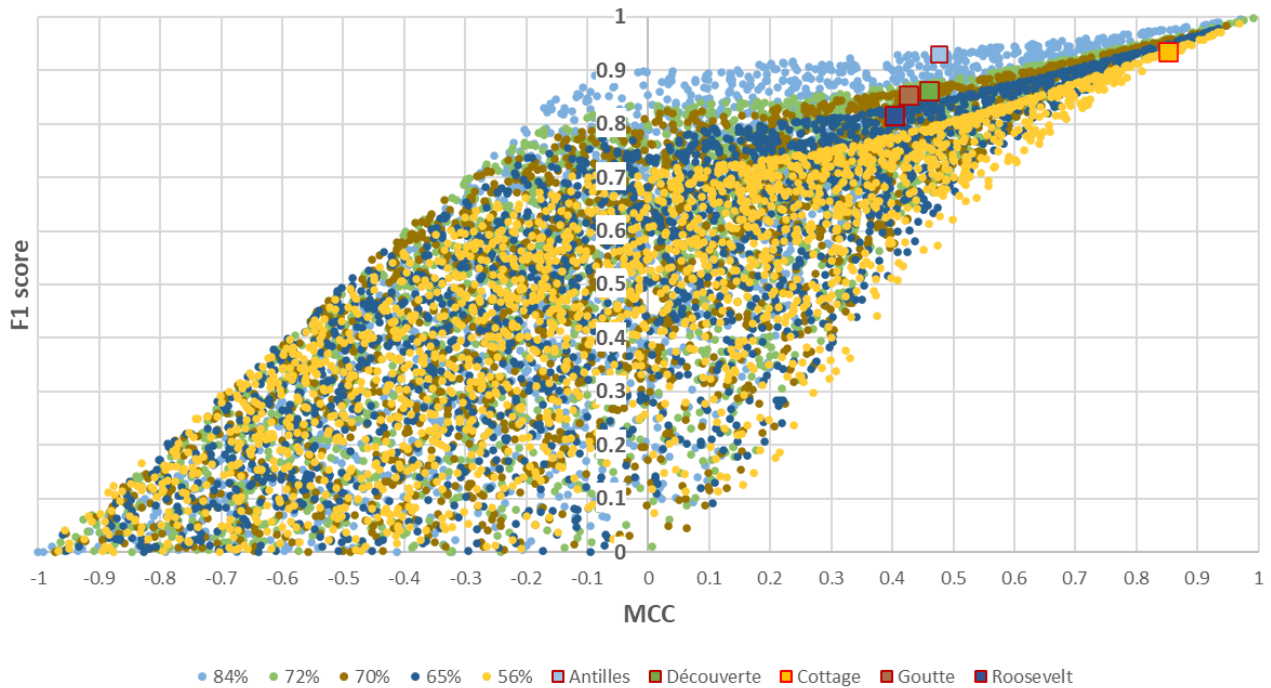
**Table 51** shows the results of the F1 score and the MCC as a function of site and model. The F1 score results are notably high, all exceeding 0.8. However, MCC results in this context remain relatively low, not exceeding 0.5.

**Table 51: Performance metric of direct evaluation of the best models on other sites data**

		Cottage	Antilles	Découverte	Goutte	Roosevelt
Model A	F1 score	0.948	0.930	0.865	0.859	0.815
	MCC	0.881	0.471	0.429	0.507	0.365
Model B	F1 score	0.950	0.929	0.859	0.854	0.812
	MCC	0.885	0.493	0.461	0.435	0.389

As discussed in [Section 6.1.2](#), it is important to highlight a potential bias between the F1 score and the MCC depending on the anomaly rate inherent in the database analyzed. [Figure 10-34](#) provides a visualization of each site's performance in a scatter plot corresponding to its anomaly rate. It can be seen that performance is relatively far from the minimum expected level (lower delimitation of the scatter plot corresponding to the relative anomaly rate of each site, illustrated by similar colors between the site (framed square in red) and its anomaly rate). Although the results are not optimal, they remain interesting given the anomaly rate present in the database studied.

The application of the trained model on other sites effectively identifies a considerable number of anomalies, which results in a high F1 score. However, the model makes many errors on valid data, as evidenced by the low correlation between the model predictions and the expert, illustrated by the low MCC. This observation can be explained by the different dynamics of other sites, whose normal functioning differs from that of our reference site, the "Cottage". As a result, the model fails to recognize the normal patterns of these sites and tends to invalidate them if they do not have similarities with those observed at "Cottage".

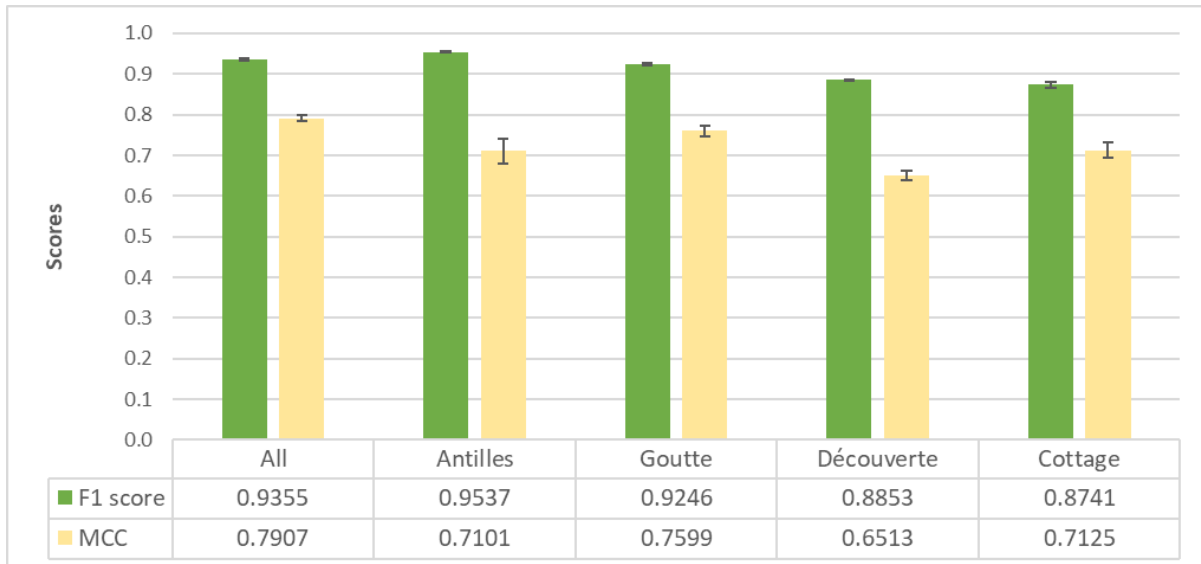


**Figure 10-34: Results at different sites and their comparison according to their anomaly rate**

#### 10.4.2. Training Model B using data from different sites

To remedy the above problem, a first approach is to reset the learning of the best architecture, in this case model B, because its higher number of neurons enables it to learn more. This reset is carried out using valid sequences from the different sites, with the aim of enabling the model to assimilate the different dynamics of normal operation. However, in this phase, the Roosevelt site is excluded due to its significant differences with other interceptors, as explained in [Section 4.3.3](#).

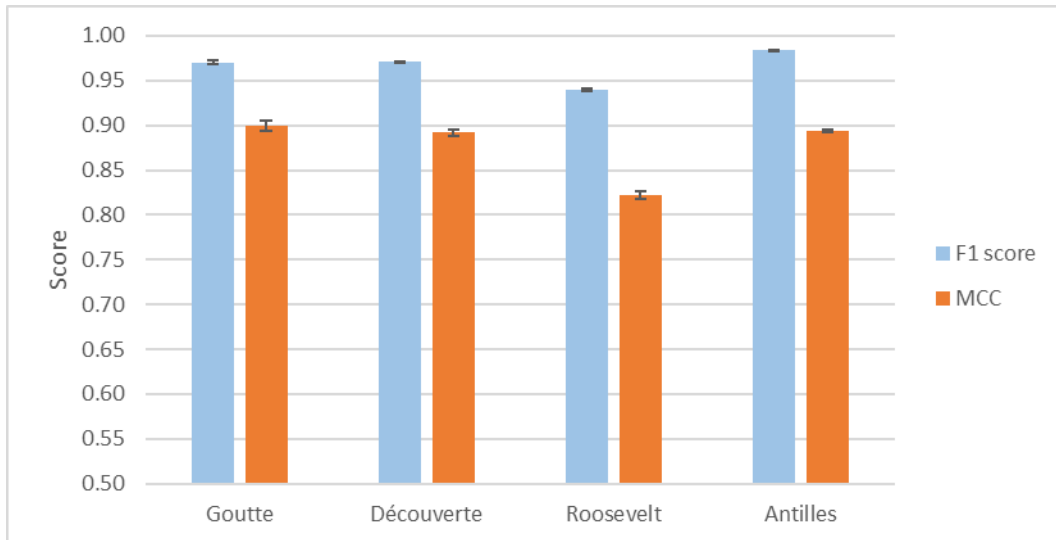
[Figure 10-35](#) shows the results of the evaluation of this model on data from all sites, as well as on each site independently. Globally, out of a total of 3082 invalid sequences, the model manages to identify 2841 of them, while generating 140 false positives. Overall, results improved, particularly in terms of the MCC, which now exceeds 0.65. It should be noted that there is a slight decrease in performance for the "Cottage" site, which is to be expected, given that the model adopts a generic approach and does not focus specifically on its operation. On the contrary, it seeks to identify anomaly characteristics in a global and common way across the different sites.



**Figure 10-35: Evaluation of the generic model on the whole dataset and on the specific data of each site**

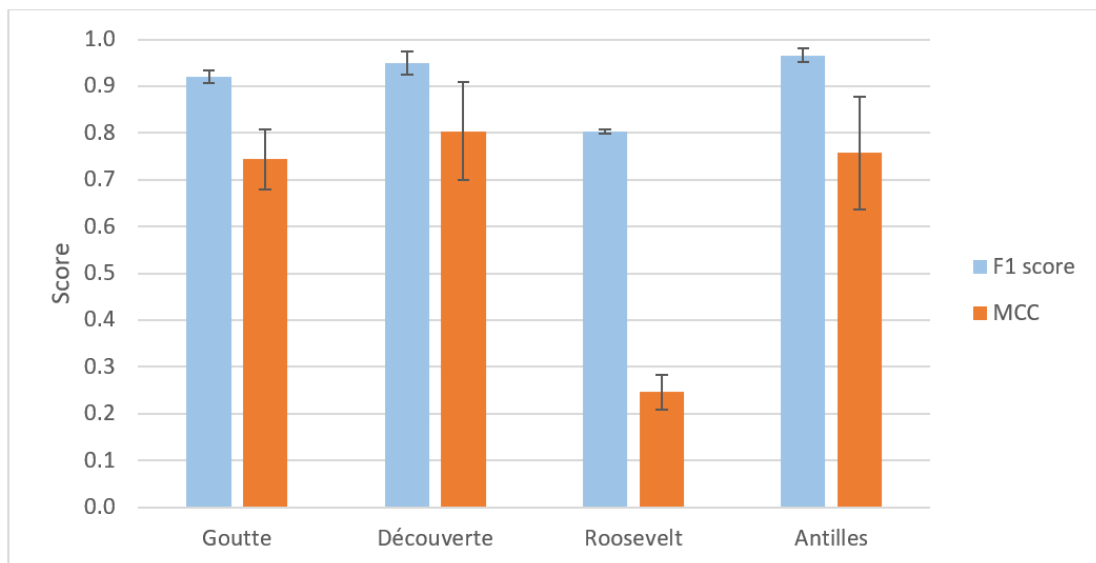
#### 10.4.3. Tuning a specific model for each site

Another approach to evaluate the model on sites other than Cottage is to train specific models for each site, in particular for Roosevelt, whose atypical operation requires a distinct approach. In this method, we keep the same architecture (that of Model B) and re-train the model site by site, evaluating each model on the database of the site in question. **Figure 10-36** shows the results obtained using an approach based on the PR curve, demonstrating a significant improvement in performance, with an F1 score approaching 0.95 and an MCC exceeding 0.8, as is the case for Cottage. We therefore conclude that the model's architecture is generalizable, although achieving scores similar to those of Cottage requires site-specific learning.



**Figure 10-36: Performance metrics of specific models, evaluated on their site database using the PR curve approach**

On the other hand, performance evaluation using the 3-sigma rule reveals a deterioration in results, particularly for the Roosevelt site. The F1 score for Roosevelt in the hypothetical scenario where the model predicts a single class (precision equals the anomaly rate and recall equal to 1), we obtain an F1 score of 0.79. In comparison, the F1 score obtained by the model is 0.8. So, we can see how close we are to this no-skill mode of operation. The 3-sigma rule turns out to be very penalizing for this site, and it would therefore be necessary to calibrate an optimal threshold using the PR curve if we wish to achieve satisfactory performance, or we may find another optimal architecture.



**Figure 10-37: Performance metrics of specific models, evaluated on their site database using the 3-sigma rule**

### 10.5. Multivariable approach for anomaly detection

The aim of this section is to evaluate a multivariate approach. So far, the simultaneous injection of the T1 and T2 sequences into the model has been done successively, thus losing any link between them under a monovariate approach. To remedy this, our aim is to provide the model with both sequences as a whole. Thus, for each timestamp, the model will simultaneously have the corresponding T1 and T2 measurements. We are also exploring additional multivariate approaches by adding the reconstructed turbidity and/or conductivity.

The autoencoder architecture used so far is mainly composed of dense layers. Using dense layers for a multidimensional input breaks the links between the input variables. Indeed, when a dense layer processes this input, it treats each element as a distinct feature, with an individually assigned weight. To treat the variables as a single set, it would be appropriate to turn to convolution layers. The latter uses filters sharing weights to detect similar patterns in different parts of the input, making the convolution layer very effective at extracting spatial features. We therefore carried out a sensitivity test to multivariate data using our base model (Model B) on the one hand, and the same architecture with convolution layers in place of dense layers on the other. To check the impact of the additional variables, we replaced each variable's data with white noise at each iteration. This allows us to assess the importance of each variable in the decision-making process, and to determine whether it actually brings a significant improvement.

**Figure 10-38** shows the results obtained using the model with dense layers. A first observation reveals that the model gives greater weight to the T1 variable, since deleting the latter leads to a significant degradation of the results. In contrast, the model with convolutional layers assigns relatively equivalent weights to the two variables T1 and T2, and the deleting of either leads to a similar degradation of results (see **Figure 10-39**). As for the model with dense layers, the best performance, albeit subtle, is observed in combinations with reconstructed turbidity. The addition of conductivity does not appear to bring any significant improvement. On the other hand, in the model with convolution layers, the best results are obtained when conductivity is included, while the addition of reconstructed turbidity tends to disrupt the model, as its removal does not noticeably affect the results. In conclusion, the two main variables carrying the most information are T1 and T2, while the impact of reconstructed turbidity and conductivity remains rather limited. Furthermore, comparison of these results with the monovariate approach highlights a deterioration in performance, leading to the assessment that this approach is not very promising in our case.

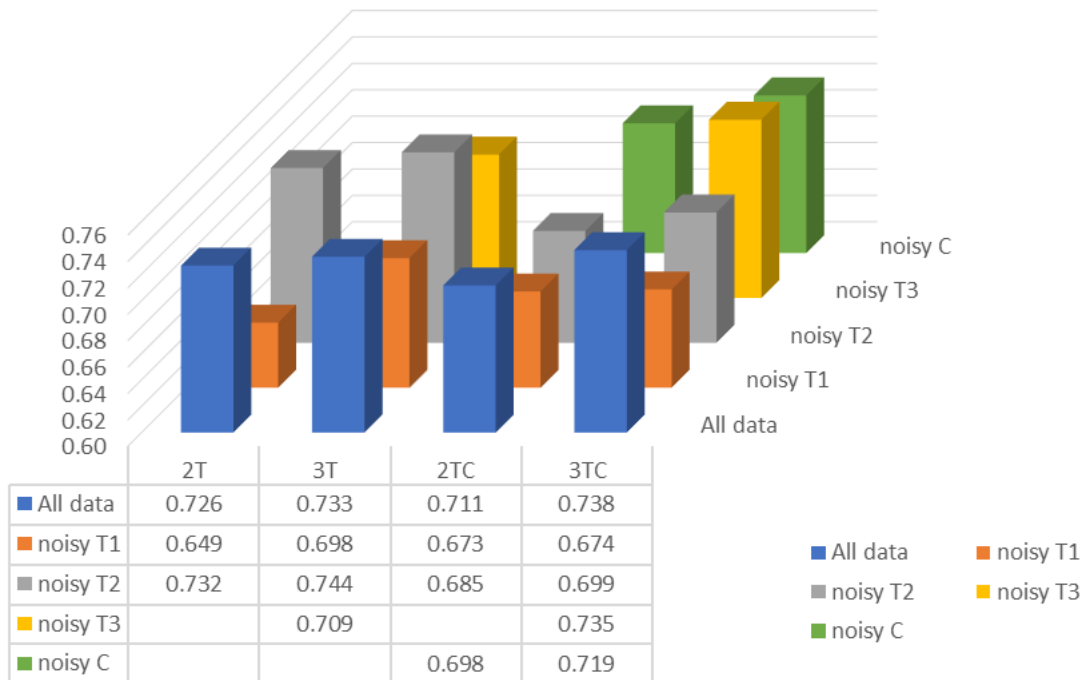


Figure 10-38: F1 score results using dense layers. T3 refers to the reconstructed turbidity

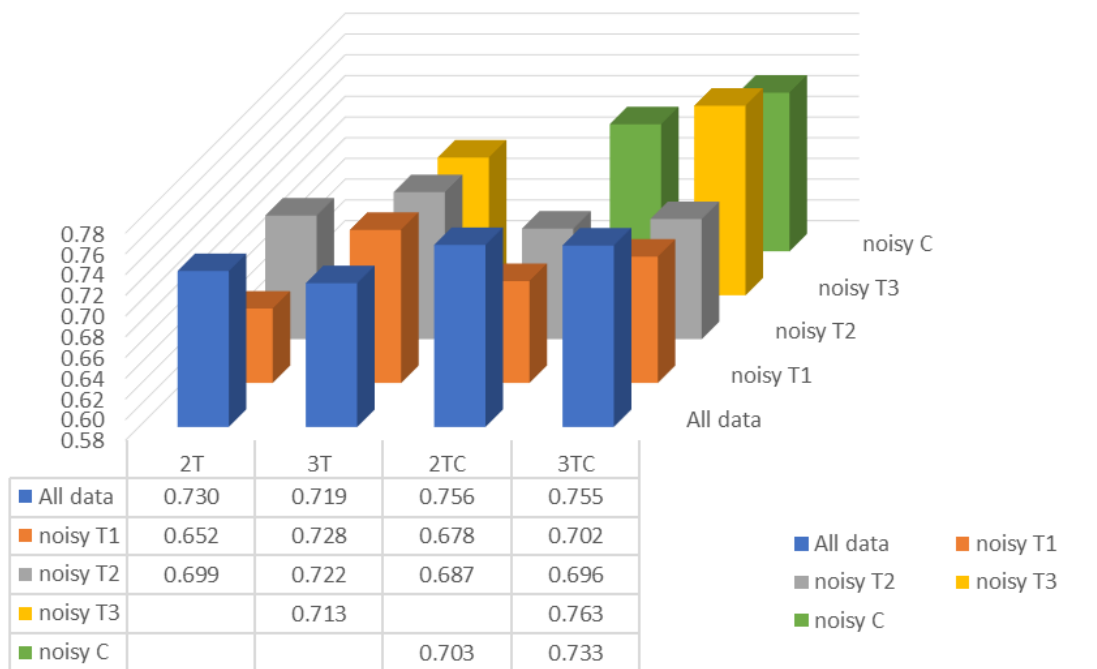


Figure 10-39: F1 score results using convolutional layers. T3 refers to the reconstructed turbidity

### 10.6. Synthesis of Chapter 10

The autoencoder is trained to reproduce input sequences, and its performance is assessed using MSE. Subsequent tests focus on sensitivity to input data, exploring preprocessing techniques, activation functions, and input variations. The results reveal that a semi-supervised approach with 100% valid sequences and standardization yields optimal performance. The impact of using raw versus reconstructed data is discussed, with a preference for raw data.

The second phase consists of the optimization of hyperparameters, with particular emphasis on setting the adequate architecture. In the context of an AE, sensitivity to the size of the latent space is explored. By progressively adjusting the size of this space, a positive correlation is observed between increasing the latent space dimension and improving performance. We also compare different architectures, highlighting optimal performance with three hidden layers. Finally, the influence of input window size on model performance is examined. Tests show equivalent performance between two models (A and B) for a 24-hour window.

In the quest to improve model performance, several strategies are explored. Firstly, increasing the size of the database reveals that a potential increase of 30% in the database size demonstrates optimal performance with an expected MCC of 0.9 and an F1 score of 0.96. Subsequently, attention is directed towards refining classification rules, with a critical evaluation of the 3-sigma rule and the Precision-Recall curve approach. While the former may lead to excessive invalidation, the latter involves converting manual labels and assessing the model's sensitivity to different classification thresholds. The analysis exposes challenges in handling sequences with high anomaly rates. Further enhancement is pursued through an ensemble model approach, combining the two best-performing models using consensus. This approach significantly reduces false positives, achieving a precision of 0.99. Lastly, the implementation of pre-validation approaches proves beneficial, automatically invalidating trivial anomalies and reducing false positives when combined with the autoencoder classification.

In an effort to extend anomaly detection to various sites, the direct evaluation of the best models on different sites' turbidity data reveals high F1 scores but relatively low MCC results. The models struggle to adapt to dynamics beyond the reference site, leading to errors on valid data. A reset of training of Model B using valid sequences from diverse sites improves overall results, with an MCC exceeding 0.65. Training specific models for each site further enhances performance, with F1 scores nearing 0.95 and MCC surpassing 0.8. Exploring a multivariable approach indicates T1 and T2 variables carry the most information, while the impact of reconstructed turbidity and conductivity remains limited.

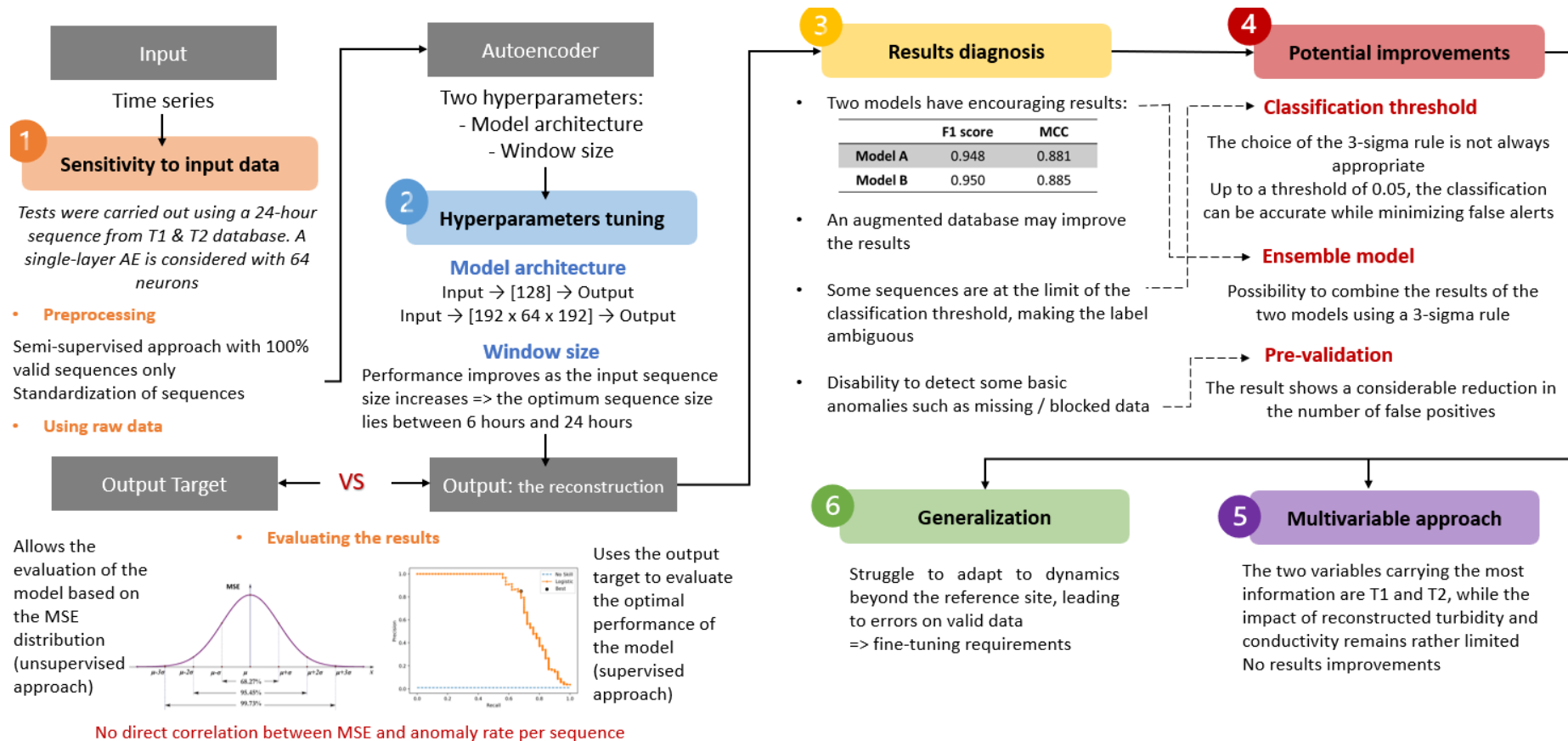


Figure 10-40: Overview of Autoencoder tests and results for anomaly detection using turbidity data





# Chapter 11. *Stretching Boundaries of Data Validation using AI*

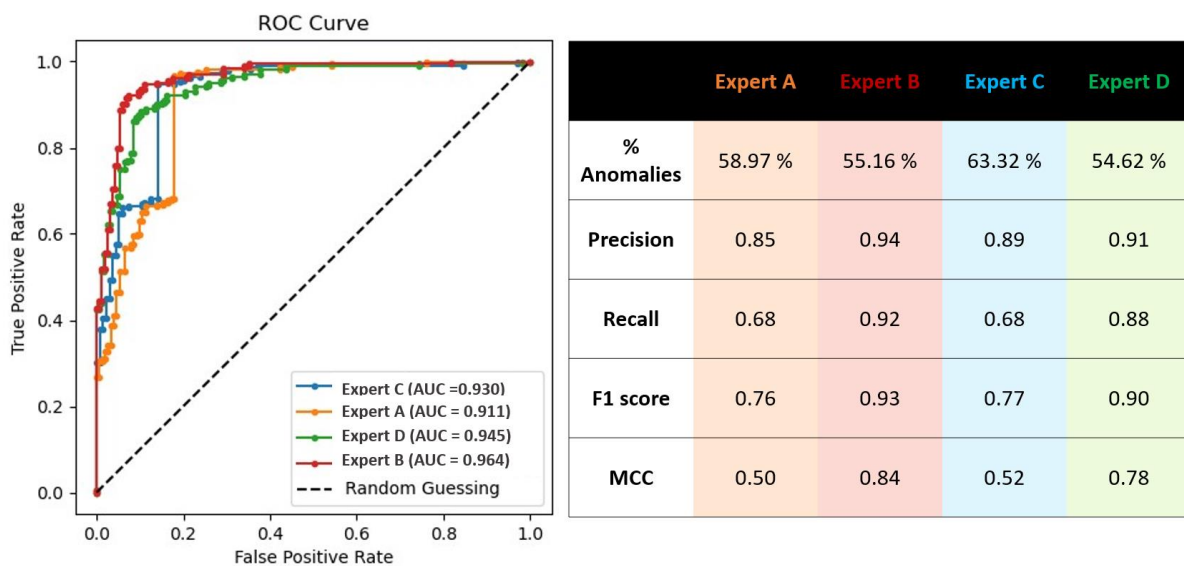
---

So far, our evaluation strategy has been centered on turbidity data from Saint Malo Agglomeration, selected for practicality and availability reasons. To establish a baseline for model comparison and provide input data for supervised approaches, we implemented a manual validation process. The validation pool exhibited a certain degree of subjectivity, albeit within an acceptable range to uphold the assumption that a single expert approximates the ground truth. Throughout our evaluation, a sole expert was retained as the reference for assessing model performance. However, to gauge the impact of using a different expert as a reference, we re-evaluated the performance of our identified best model, namely the autoencoder (see [Section 10.6](#)), considering the validation pool database. The objective is to evaluate the range of variability of the final performance ([Section 11.1](#)). Furthermore, it is crucial to assess the model's extrapolation capability to a completely external chronicle, as demonstrated by evaluating turbidity data from Cottage spanning August 1, 2022, to July 31, 2023 ([Section 11.2](#)). This aims to examine the model's stability over time and its potential use without requiring specific re-calibration. Expanding our focus beyond turbidity, the wide-ranging need for data validation in wastewater networks encompasses various sensor types. To validate this concept, we aimed to conduct proof of concept on other data types. Acquiring manually validated data poses challenges, making it impractical to repeat a specific validation process given time constraints. Consequently, we turned to unsupervised approaches. In view of the unsuitability of our best autoencoder model, which necessitates pre-selection of normal sequences for training, we performed our proof of concept using the Matrix Profile model. This unsupervised model does not require prior knowledge or training. The data studied in this context includes conductivity data from Saint Malo Agglomeration ([Section 11.3](#)) in addition to water level data from the Public Water Management Company (SPGE) of the Walloon Region - Belgium ([Section 11.4](#)).

## 11.1. Relation between annotator agreement and model's performance

After examining the characteristics of annotator agreement and considering their subjectivity, the next step involves exploring their relationship with the model's performance. Therefore, we evaluate our top-performing model, which is the pre-trained autoencoder in this context, using ground truths (GTs) provided by different experts. The central question addressed in this research is the extent of the impact of varying ground truths on the reported performance of an algorithm.

Analysis of the results presented in [Figure 11-1](#) reveals two distinct performance classes, Expert B and D on the one hand, and Expert A and C on the other. This classification differs from that highlighted in [Chapter 7](#), where a stronger correlation was observed between Experts A and D, then between B and C. This discrepancy is directly linked to the rate of anomalies identified by each expert, which varies between 55% and 63%. For the experts with the lowest rates, namely B and D, who selected the fewest anomalies, the recall is remarkably high, exceeding 0.88. For the other two experts, on the other hand, recall is lower, indicating a problem in the selection of anomalies identified. Precision, in all cases, remains of the same order of magnitude and satisfactory overall. However, if we compare these results with those of the agreement between the different experts (see [Figure 7-3](#)), we can see that the variability of the F1 score between the model and the different experts is similar between the different expert peers, with a peer-to-peer F1 score varying from 0.76 to 0.86. So, even taking into account the variability of the results, the model remains very interesting, freeing us from human bias and the heavy workload associated with daunting tasks. It should be noted that the upper limit of the model exceeds that of inter-expert agreement, without however being attributed to overfitting, given that the model has never been exposed to the expertise of at least one of the two experts, in this case expert A.



**Figure 11-1: Results of the AE model using different baselines issued from different experts**

Multiple ground truths are also generated at increasing levels of agreement, where  $\tau = 1/N$ , representing anomalies marked by any annotator;  $2/4$  and  $3/4$ , denoting consensus among half of the annotators or a majority vote; and  $4/4$ , indicating anomalies unanimously agreed upon by all annotators. The results for different rates of inter-annotator agreement are shown in [Figure 11-2](#). An important observation is the predictable increase in model performance

with inter-annotator agreement, especially in the higher recall ranges. These results indicate that the model is able to effectively identify the majority of anomalies without significant difficulty (recall = 0.94% and precision = 0.89%). However, the consensus method can sometimes focus on evaluating a model against the most obvious anomalies, thus providing overly optimistic performance estimates. However, we find that even the use of majority voting gives similar results. With a discord of 14% on 65% of anomalies, representing a relative deviation of 22%, the variance on the model's reported F1 score is 5%, which is still very advantageous.

Indeed, according to [190], one factor that stabilizes reported performance is low annotation variance. Here, we observe relatively low variation between annotations in terms of Smyth coefficient and pairwise F1 score, resulting in limited dispersion of performance curves. Thus, choosing any of the GTs to evaluate an algorithm would result in similar reported performance.

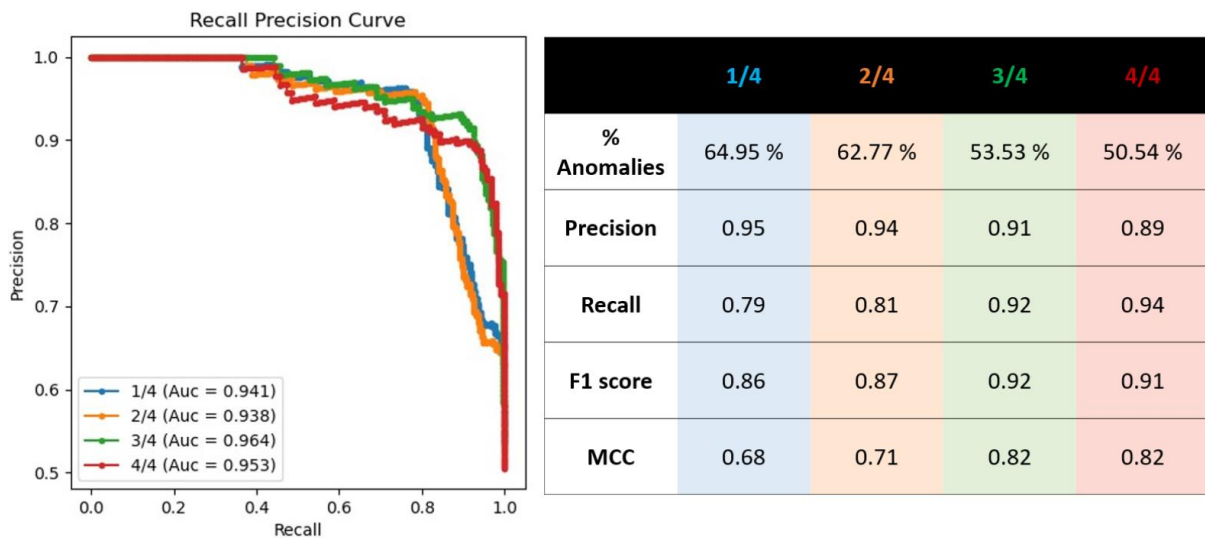


Figure 11-2: Results of the AE model using different annotator agreement's rates

In summary, beyond evaluating an algorithm on a dataset with diverse samples, it is crucial to assess the algorithm using multiple ground truths. The observed variance in performance across these diverse ground truths can serve to quantify the confidence in the model's performance. Therefore, it is advisable to evaluate the inherent uncertainties in annotator judgments prior to assessing detection algorithms, as measures of absolute performance may significantly differ based on the chosen ground truth. One approach to mitigate this bias is to have a reference with low variance. In our case, we evaluated this by employing a validation pool of multiple experts and assessing the model's performance across different experts and levels of annotator agreement. It appears that the variability in the model's performance is of a similar magnitude to what can be observed among experts, with the model exhibiting a closer

resemblance to certain experts than they do between each other. Moreover, we observe that the model's performance remains relatively stable across different annotator agreements, allowing for performance reporting with a precision of approximately 5%. It is worth noting that this configuration may not always be applicable and should undergo specific tests or include an error margin in the reported performances.

### 11.2. Model's extrapolation to new chronicle

Once the performance of the AE model has been validated on data from our test site, it becomes crucial to examine the extrapolation of this model on new data from the same site. To this end, we are using the Cottage turbidity chronicle from August 1, 2022, to July 31, 2023, which we will refer to as New Cottage. **Table 52** shows the evaluation results of the model trained with Cottage data (from February 1, 2021, to July 31, 2022) on New Cottage data. A significant deterioration in results is observed, particularly in the F1 score and the MCC. Indeed, the model has an increased likelihood of triggering a higher number of false alarms, indicating that it perceives certain situations as abnormal based on its training, whereas they are in fact normal.

**Table 52: Evaluation results of the Cottage-trained model on the Cottage and New Cottage databases**

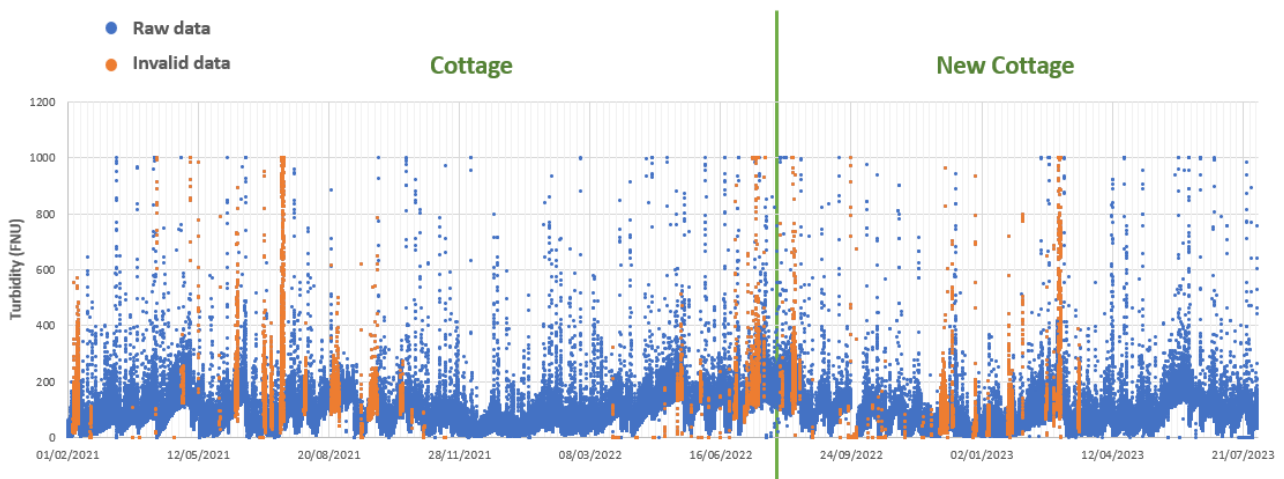
		Precision	Recall	F1 score	MCC
Train ↓ Evaluate	Cottage	0.9249	0.9461	0.9354	0.8511
	New Cottage	0.7106	0.8720	0.7831	0.4381

Given that overfitting and generalization tests on the other sites have demonstrated that the model is not overtrained, a legitimate question arises at this stage of the study: is this a new "normality" that the model previously failed to recognize? With this in mind, we took the opposite approach, training the model with New Cottage data and evaluating it on Cottage (see **Table 53**). Nevertheless, the result remains the same, with a deterioration in MCC and precision in the same fashion as the last test.

**Table 53: Evaluation results of the New Cottage-trained model on the Cottage and New Cottage databases**

		Precision	Recall	F1 score	MCC
Evaluate ↑ Train	Cottage	0.6806	0.9020	0.7758	0.4114
	New Cottage	0.9463	0.9783	0.9620	0.9110

Indeed, these results suggest a real disparity between the two chronicles. However, when we examine the average turbidity, whether for valid or invalid sequences, and the overall anomaly rate between the two, we observe similar orders of magnitude. No sensor or configuration changes were reported in the logbook, making this result particularly perplexing. When we explore the overall representation of data and anomalies in [Figure 11-3](#), we identify a distinct dispersion of valid sequences throughout the entire database.



**Figure 11-3: Manual validation results of Cottage turbidity data**

[Figure 11-4](#) depicts the ratio of fully normal sequences utilized for training both models (using Cottage and New Cottage databases respectively) across different months. Consequently, we observe that the majority of valid sequences for the first model are concentrated in the first quarter, whereas for the second model, they are more localized in the last quarter. Notably, these periods exhibit distinct seasonality, which could account for the decline in results as the learned normality differs between the two models.

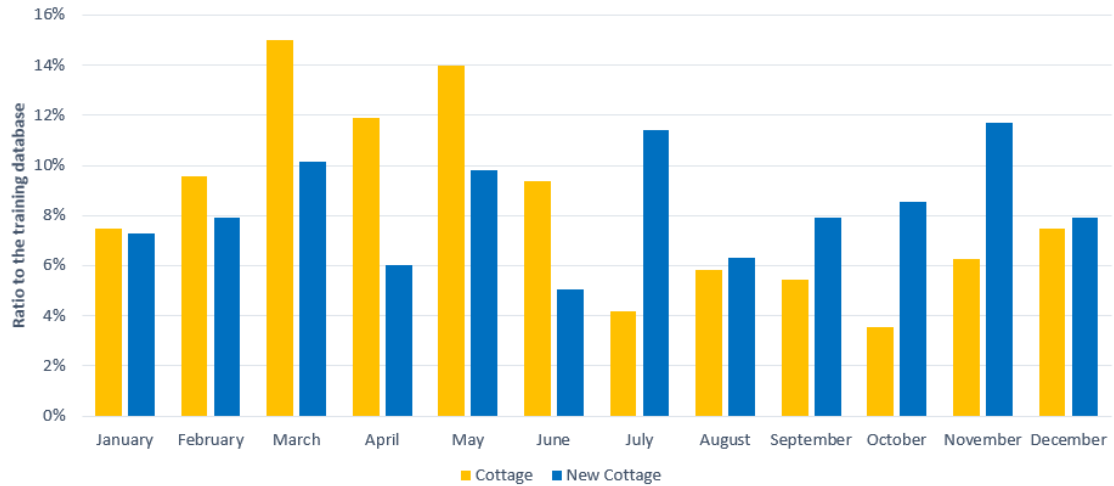


Figure 11-4: Ratio of normal training sequences

Learning with the full data set would free us from this bias. The results of this evaluation are presented in Figure 11-5, but in the absence of another validation set, we cannot state with certainty that the origin of the problem is indeed this one and that we can better generalize our model to new chronicles. So, although using 18 months of measurements as input, pre-selecting only 100% valid sequences could introduce a bias by providing more sequences of a specific configuration. Tests in this direction would require a larger database in order to draw conclusions about the extrapolation of our model to a new chronicle in the absence of any change in hydraulic configuration.

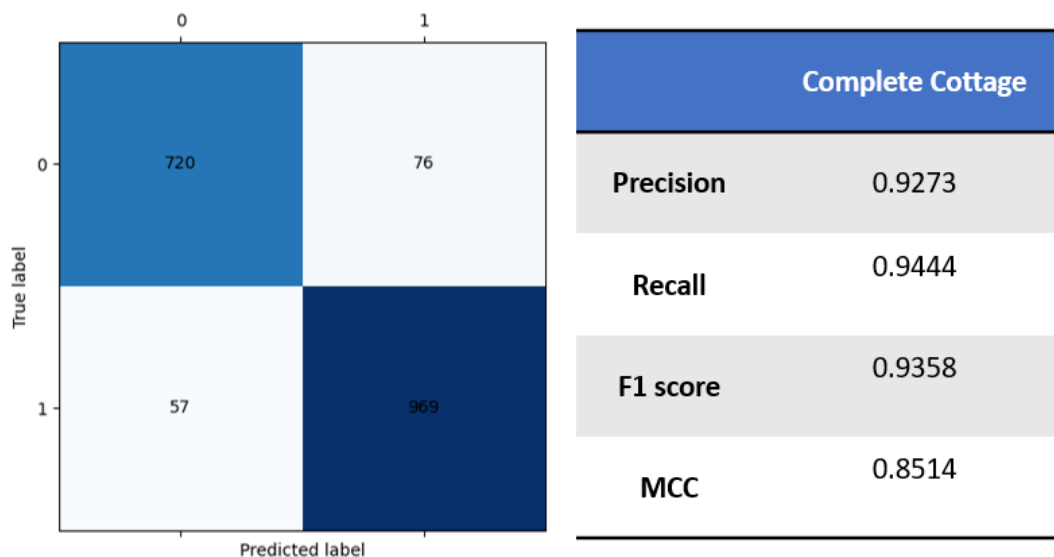
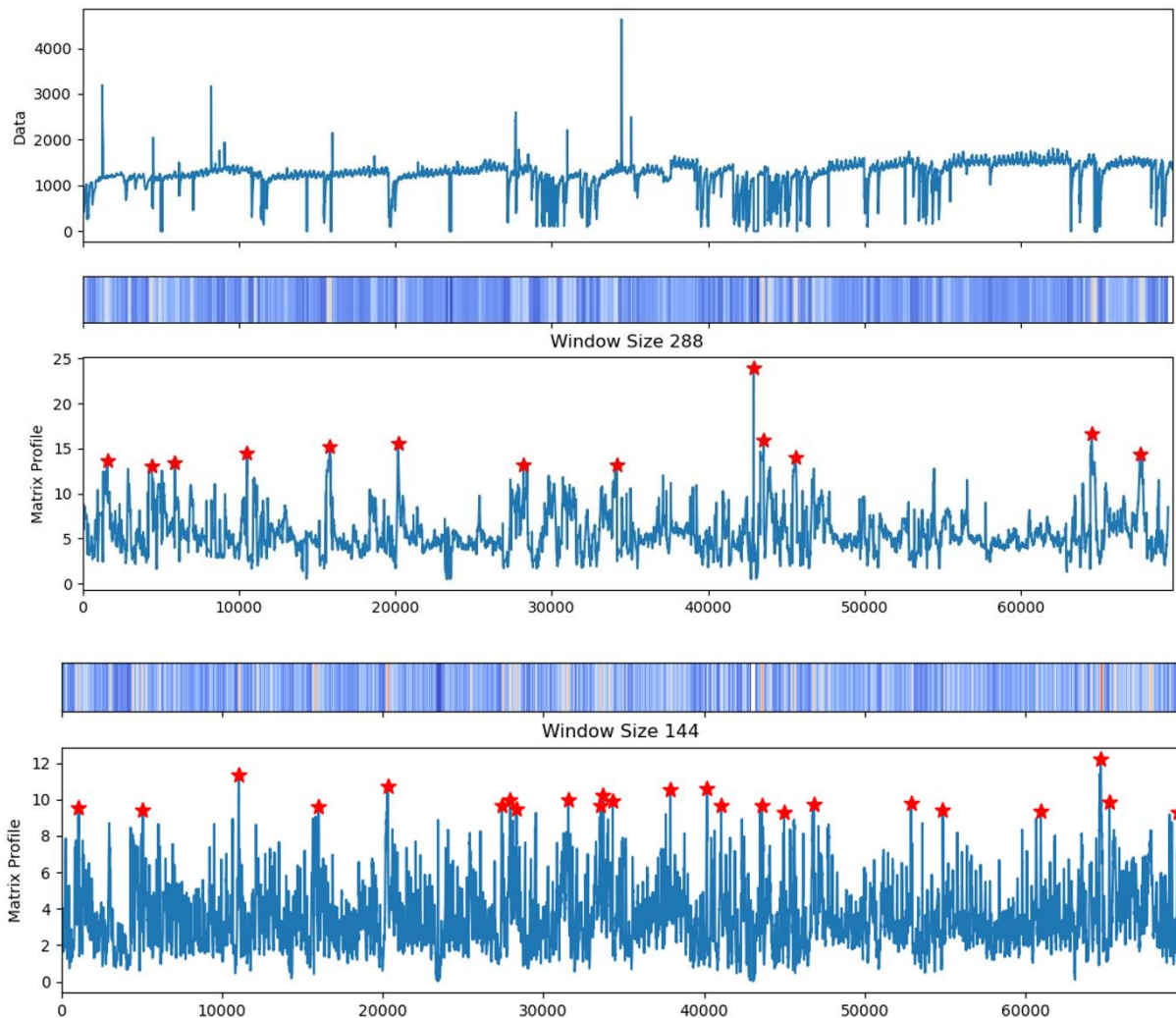


Figure 11-5: Evaluation results of the model trained on the complete Cottage database

### 11.3. Conductivity validation using Matrix Profile

The use of the matrix profile on conductivity data serves as a qualitative assessment, aiming to evaluate the algorithm's performance on a different data type and to gauge the reliability of the measurements. Typically, measurements of conductivity in wastewater networks are deemed reliable, leading to a lack of domain validation for this specific data type. Consequently, the absence of a target output hinders the possibility of conducting a quantitative evaluation of the model. As a result, our analysis focuses on the model's ability to accurately capture and characterize the underlying patterns within the conductivity data. The choice of the window size is made considering a domain knowledge on the general dynamics of wastewater networks. Hence, two window lengths are evaluated; fixed at 12 and 24 hours. The anomaly ratio is evaluated at 5% since the chronicle is not supposed to have many discords (see [Figure 11-6](#)).



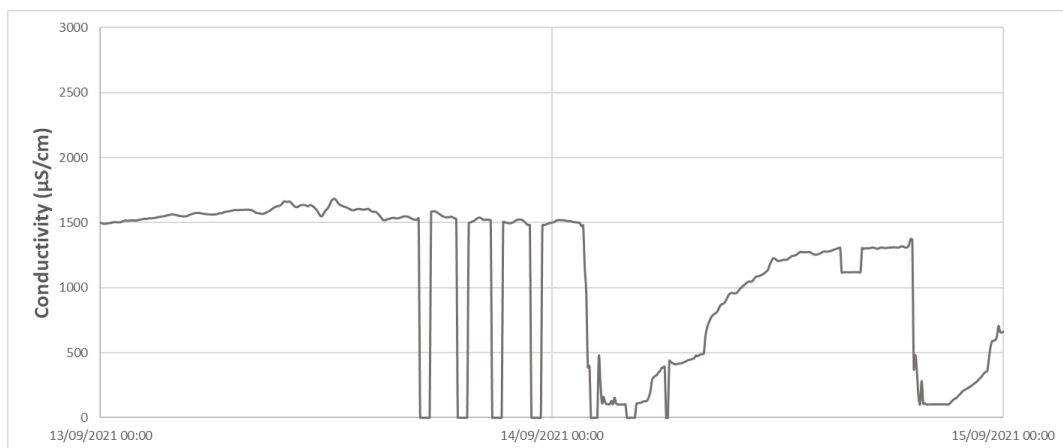
**Figure 11-6: Anomaly detection on conductivity data using matrix profile**



A close look at the matrix profile heat map reveals the absence of any outstanding anomalies. The range of variability in the profile is significantly reduced, indicating similarity in the data. We note that, for smaller window sizes, we identify an increased number of sequences considered anomalous. This increase is to be expected, given the reduction in the window size and the unchanged rate of anomalies to be detected. However, a comparative analysis of the two tests reveals an interesting trend: abnormal sequences appear to be aligned in a similar way. This convergence suggests a certain stability in anomaly detection, irrespective of the size of the window chosen.

Focusing on the sequences with the highest anomaly scores, we see that the day of June 30, 2021, stands out as the most anomalous when a 24-hour sequence is considered, presenting missing data over the entire period. However, this same sequence shows a significantly lower score when using a 12-hour time window. The reason for this is quite simple: dividing the sequence in two produces two totally similar sequences, which then become each other's closest neighbors. et this sequence is indeed abnormal.

A closer look at the largest anomaly detected with a 12-hour sequence, occurring around September 14, 2021 (also reported with a 24-hour sequence), reveals consecutive drops in data, with a significant drop at the start of the day (see [Figure 11-7](#)). This pattern is clearly anomalous, and the model makes the right decision in invalidating it.

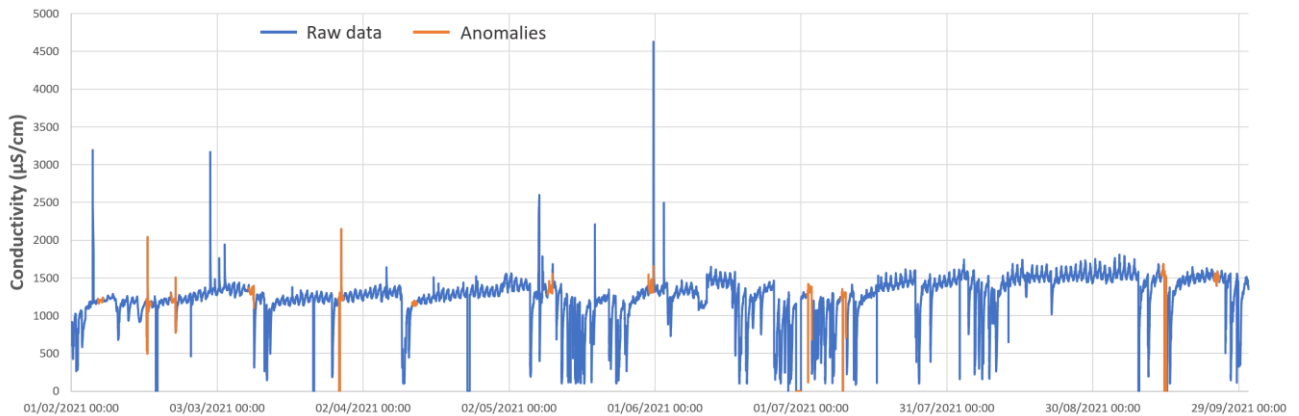


**Figure 11-7: Zoom in on the most abnormal sequence in the conductivity dataset using a 12-hours sequence**

To assess the reliability of overall results, it is essential to understand the dynamics of conductivity. Conductivity peaks may be associated with exceptional flow, possible seawater intrusion or special valve management. Conversely, a drop in conductivity may be linked to dilution caused by a rainfall event. In the case of Saint-Malo, all these scenarios are plausible.

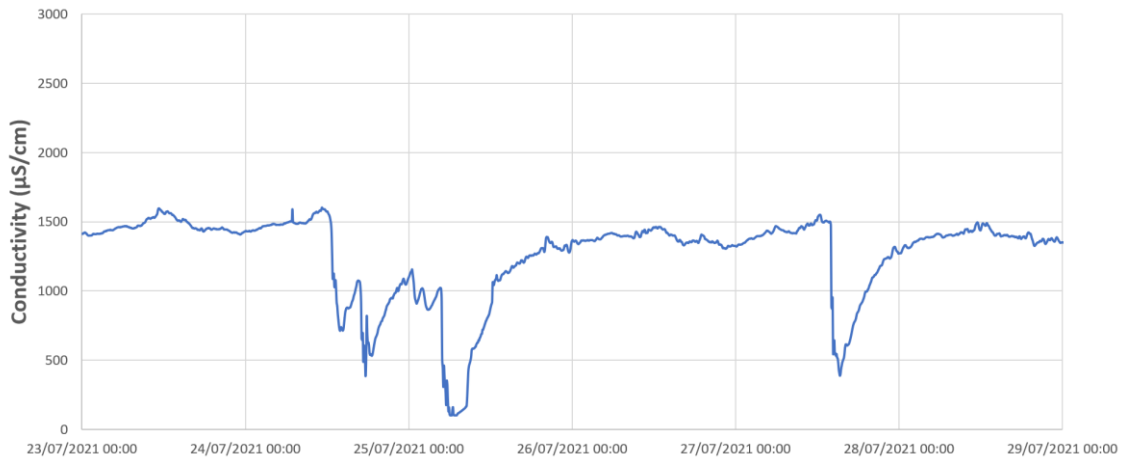
However, to confirm a particular scenario, it is necessary to consult the logbook on the one hand, and the rainfall data on the other.

It's important to note that the anomalies identified by Matrix Profile are not all of the same type and may concern both peaks and drops in conductivity (see [Figure 11-8](#)). Furthermore, not all drops or peaks are systematically invalidated, suggesting an apparent pre-selection of faults rather than automatic invalidation.



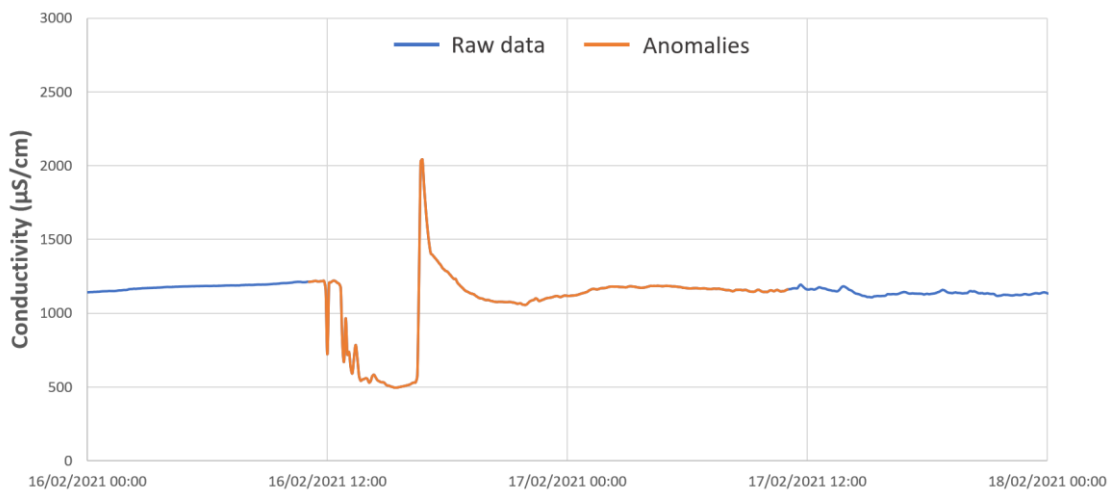
**Figure 11-8: Overall validation of conductivity data using Matrix Profile**

The general pattern of conductivity during a rainy event is illustrated in [Figure 11-9](#). There is a rapid drop at the start of the event, followed by a gradual recovery to normal levels. This dynamic aspect is characteristic of conductivity in wastewater networks during rainfall events. The July 24, 2021, event was marked by the occurrence of several rainy episodes (3.8 mm of rain in Saint-Malo), resulting in successive drops in conductivity. However, the process of returning to normal maintains a constant structure. On the other hand, the event of July 27, 2021, corresponds to a rainfall of 2 mm, with a faster but gradual return to normal.



**Figure 11-9: Typical conductivity pattern in wastewater networks during rainy events**

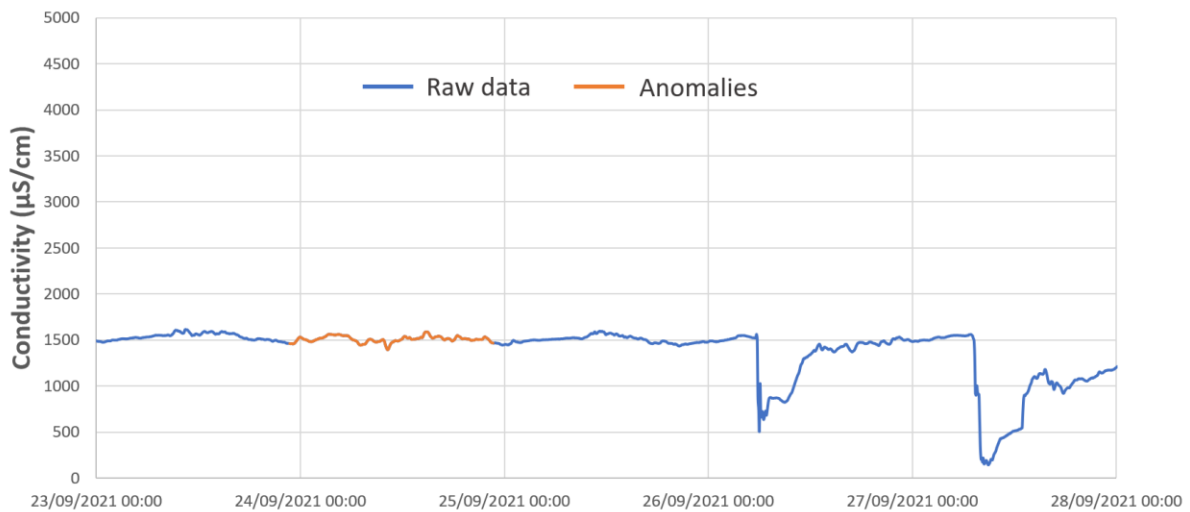
**Figure 11-10** highlights one of the anomalies detected by the Matrix Profile model. This anomaly is characterized by a significant increase in conductivity, rising from 500  $\mu\text{S}/\text{cm}$  to 2000  $\mu\text{S}/\text{cm}$  after a rainfall event. This behavior is considered abnormal by the model, which can also be the case hydraulically, provided that no exceptional event took place at that precise moment. In the absence of external information to confirm this point, it is nevertheless interesting that the model flags this situation as an anomaly, in line with our objective of detecting any deviation from the network's usual behavior.



**Figure 11-10: True anomaly identified by Matrix Profile**

However, we also note that the model invalidates sequences that are entirely normal, as illustrated in **Figure 11-11**. Analysis of this sequence reveals that it follows standard dry weather conductivity behavior, remaining relatively stable throughout the day. This false alarm problem is probably linked to the number of anomalies imposed, which most certainly exceeds the anomaly rate inherent in the database. These false alarms are observed several times,

while the rest correspond to real anomalies, generally sudden and temporary drops in conductivity.



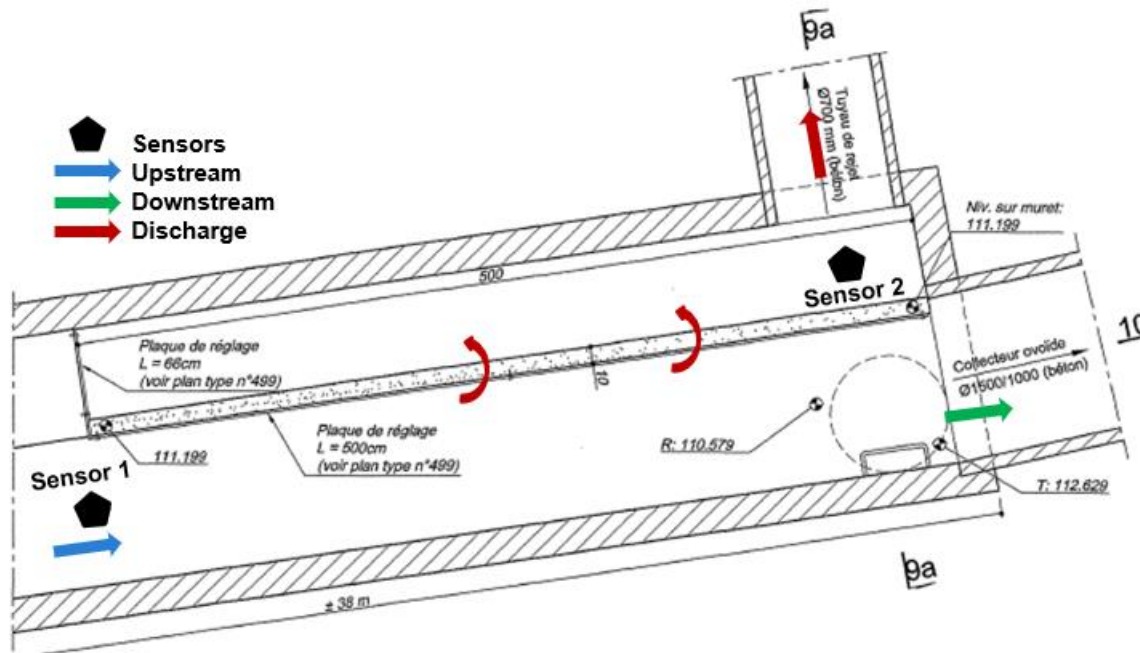
**Figure 11-11: False anomaly identified by Matrix Profile**

Furthermore, unless we undertake a thorough manual validation of the database, it is not possible to definitively determine the presence or absence of omitted anomalies. However, we have taken care to check that visible and successive drops in conductivity, which may initially look like faults, do indeed correspond to rainfall events, although this check is not exhaustive.

In conclusion, the application of MP model to the assessment of anomalies in wastewater network conductivity data offers promising prospects, while raising specific challenges. The qualitative nature of this assessment highlights the model's ability to detect unusual patterns. However, the absence of domain-specific validation for conductivity data limits the possibility of a quantitative assessment, but the model's performance in detecting real anomalies, such as sudden drops in conductivity, is encouraging. Although, the presence of false alarms underlines the need for thorough validation and suggests sensitivity to model parameters, particularly the anomaly rate.

### 11.4. Water level validation using Matrix Profile

The last adaptability tests have been carried out to detect anomalies in the processing of water level data. This data come from a storm overflow in the Waremme wastewater network, under the responsibility of the SPGE - Wallonia. The information collected includes water level measurements upstream and downstream of the spillway, obtained using two US sensors. The period covered by these data extends from May 7, 2021, to August 21, 2022. In the absence of manual validation by an expert, we opted to use the Matrix Profile model. Being unsupervised, this model dispenses with the need to pre-select sequences for training.



**Figure 11-12: Spillway configuration with the localization of the US sensors - © SPGE**

The first step in the analysis is to examine the acquisition frequencies of the various sensors, revealing irregular patterns. Subsequently, a pre-processing step was implemented to resample the data at a regular frequency of 1 minute, thus aligning the two sensors. For fine data, a linear interpolation approach was used.

Given the absence of the output target, the anomaly detection process involves exploring the hyperparameters of the Matrix Profile model via a grid-search. This search encompasses variations in the number of anomalies from 1 to 20, and in the window size from 2 hours to 72 hours, with a step of 2 hours. The concatenation of results involves evaluating, for each measurement, the number of models that identify it as abnormal. This assessment is then visualized using a heat-map, where color shades ranging from blue representing normality to red signifying an anomaly confirmed by several models, allow visual interpretation of the results.

**Figure 11-13** shows the evolution of results as a function of the number of anomalies detected using data from Sensor 1. Overall, a satisfactory stability is observed from a number of anomalies equal to 10. The additional anomalies identified appear to be more related to fault limits or have relatively low scores, indicating that they are not frequently selected by the various models. Analysis of the newly added defects reveals that they are no longer accurate. According to domain expertise and hydraulic analysis, these sequences are quite normal. The

model identifies them only because of the constraint imposed to select a specific number of defects.

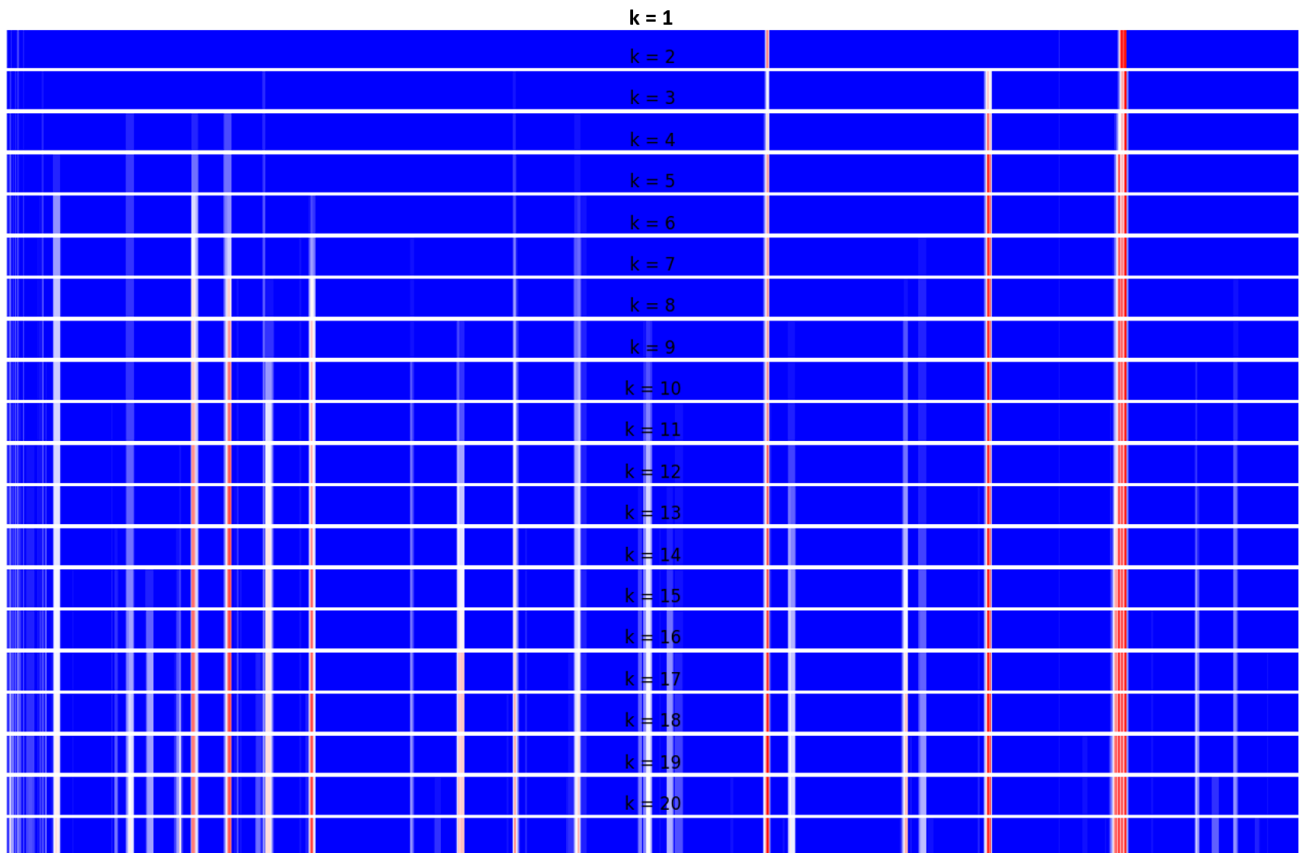


Figure 11-13: Prioritization of anomalies by concatenating results from different window sizes

Figure 11-14 shows the results for a number of anomalies set at 10. When these anomalies are examined in correlation with the logbook and exogenous data, it emerges that they are legitimately identified as anomalies.

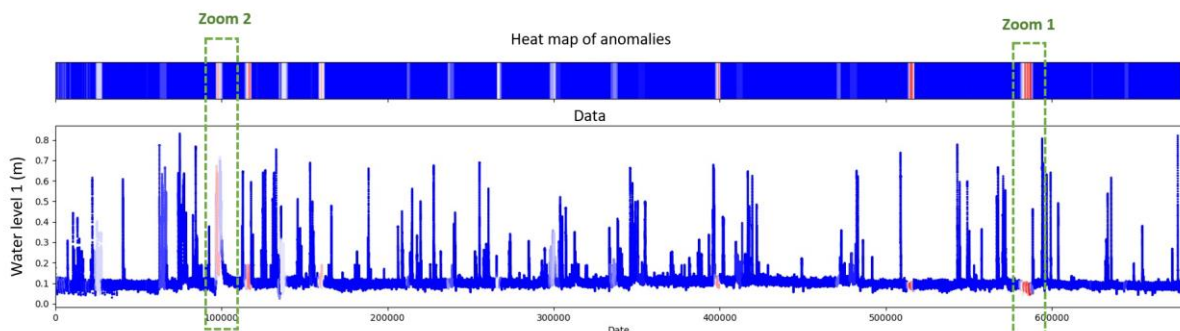
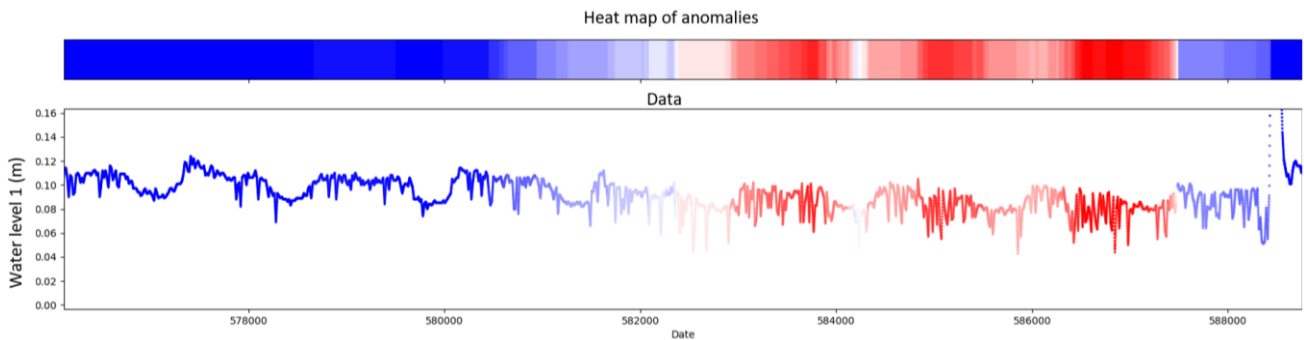


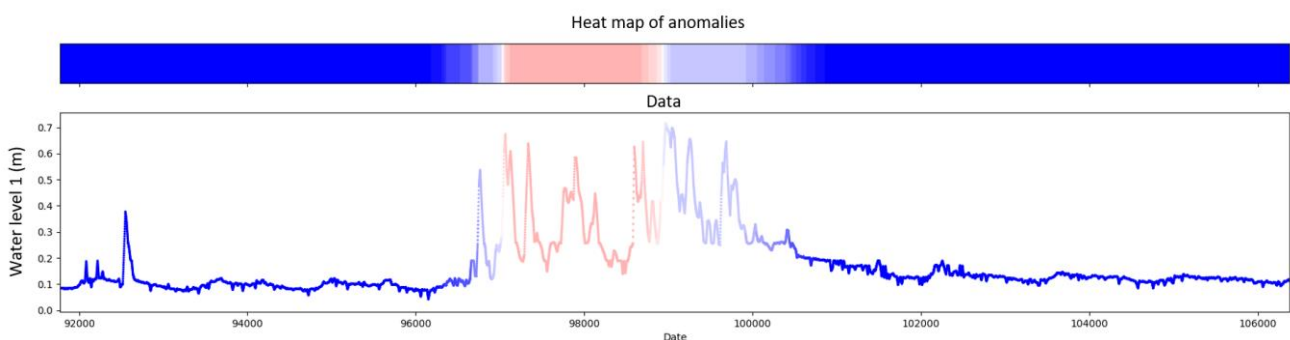
Figure 11-14: Concatenation of data validation of sensor 1 results using a multi-window size approach and  $k = 10$

Here's an in-depth look at two specific anomalies. On the one hand, **Figure 11-15** highlights the most flagrant anomaly according to various models with different window sizes. The signal is quite noisy, making the dry weather structure difficult to perceive, and a slight downward trend is observed. This anomaly is particularly relevant in this context, as it is subtle and corresponds to anomalies that are difficult to identify combining progressive drift and a noisy signal.



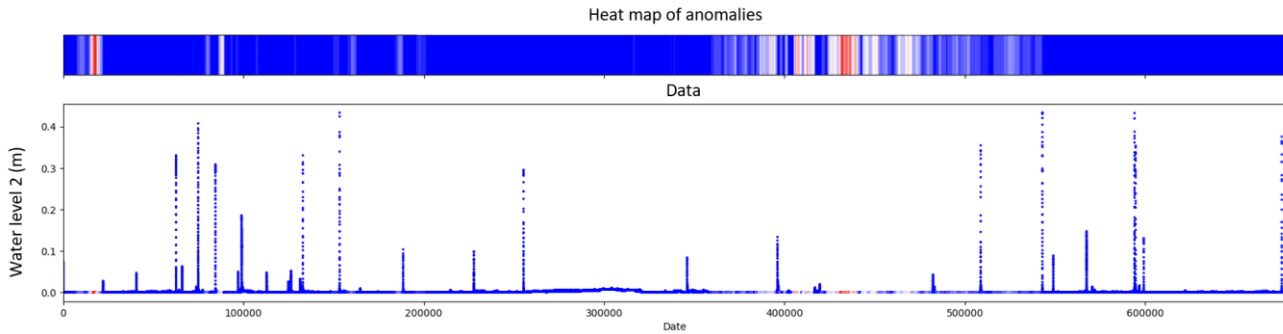
**Figure 11-15: Zoom 1: The anomaly with the highest score using a multi-window size approach**

**Figure 11-16** shows the main anomaly detected during a rainy period. At first glance, the structure does not appear aberrant, especially when compared to that during rainfall. However, it refers to the floods that hit Belgium from July 13 to 16, 2021. This example illustrates an anomaly linked to the phenomenon measured, representing an event never observed before. Still, it's an interesting event to identify.



**Figure 11-16: Zoom 2: The main anomaly during the rainy weather using the multi-window size approach**

On the other hand, **Figure 11-17** shows the result with  $k=10$  for data from sensor 2. It can be seen that all the anomalies identified are located in areas with no spills, where water levels are zero. However, these sequences are not unique, which makes it curious that the model flags them as anomalies in view of its similarity join principle.



**Figure 11-17: Concatenation of data validation of sensor 2 results using a multi-window size approach and  $k = 10$**

In fact, this situation is linked to the definition of the algorithm for calculating the distances between the various sequences using Pyscamp. In the case of flat subsequences, these regions have a mean-centered norm of 0, which means that their normalized Euclidean distances  $z$  are undefined. Totally flat regions do not correspond to any subsequence and will return NaN as the distance to the nearest neighbor. This constraint also applies to regions that are almost flat. Consequently, the application of Matrix Profile to this type of sequence is not suitable. Using this model on data with only zeros or constant values can be problematic. For example, for discharge data, it would make more sense to select only sequences during spills for analysis. However, in this particular case, a hydraulic and manual analysis of the downstream spill data indicates that the data are quite valid, which prevented us from putting this sequence pre-selection strategy into practice.

In conclusion, the adaptability tests conducted for anomaly detection in water level data have provided valuable insights. The Matrix Profile model demonstrated stability in identifying anomalies. The approach of averaging the grid search results overcomes the bias of window size tuning. However, an important anomaly number  $k$  can involve additional non relevant anomalies, highlighting the importance of domain expertise in interpreting results and choosing this parameter. Moreover, the model's sensitivity to flat regions and the challenges associated with undefined distances in such cases were discussed. Finally, the Matrix Profile model presents a valuable tool for anomaly detection in water level data, offering insights into both subtle and significant deviations. While challenges exist in handling specific data patterns, the model's adaptability and interpretability make it a promising approach for monitoring and detecting anomalies in water level datasets.



### 11.5. Synthesis of Chapter 11

The aim of this chapter was to consolidate our best results, obtained with the AE model, by focusing on the reliability of the performance metrics and the transposability of the model over time. In the first part, we evaluate the impact of different ground truths provided by different experts on the performance of the pre-trained autoencoder model. Despite variations in the classification results, the model remains interesting for mitigating human bias and workload. Model performance improves with higher inter-annotator agreement, but precautions are needed due to the risk of over-optimistic estimates with consensus methods. Performance stability is linked to low annotation variance, and the use of an expert validation pool enables confidence in model performance to be quantified at around 5%.

In addition, we examined the extrapolation of the pre-trained AE model to new data from the same site, noting a deterioration in results, particularly precision. Despite similarities between the input chronicles, the analysis reveals uncertainties, notably linked to the prior selection of 100% valid sequences. More conclusive tests would require a larger database to assess the extrapolation of the model to new chronicles.

We have also explored the adaptability of our models to other types of data, such as conductivity and water level. In the absence of references, we turn to unsupervised models, in particular the Matrix Profile model. This section highlights the promising prospects of MP in the assessment of conductivity anomalies, despite quantitative limitations due to the lack of domain-specific validation. MP's results on water level data indicate its stability in detecting anomalies, despite challenges related to sensitivity to flat regions and undefined distances in some cases. Hence, the MP model is considered a valuable tool for detecting anomalies in water level data, offering information on both subtle and significant deviations.

## ***Synthesis of Part III***

This section offers concrete elements in response to our problem of validating wastewater data using models from our benchmark (see [Section 5.2](#)). First, we focus on the challenges involved in obtaining a ground truth database and highlight the importance of a validation pool to assess the biases introduced by manual validation in artificial intelligence models. Pairwise F1 scores are calculated to assess agreement between experts, resulting in an overall mean F1 score of 0.81.

Secondly, we examine the evaluation of the different models using the various tests described in [Section 5.2](#). All three models performed well in specific contexts, but their strengths and weaknesses varied according to data characteristics and evaluation conditions (see [Table 54](#)). The autoencoder outperforms in anomaly detection compared to the two other models. But this does not prevent Matrix Profile from proving effective for data sets with a low anomalies rate, requiring less pre-processing and selection of input data. ResNet, using a regression approach, stands out for its accuracy in predicting anomaly rates, although adapting it to new sites remains a challenge, in addition to the difficulties associated with the practical interpretation of this result and the model's supervised approach, requiring rigorous validation.

In conclusion, the final chapter consolidates the best results obtained with the autoencoder model. The impact of various ground truths on the model's performance is assessed, highlighting the model's ability to mitigate human biases. Model extrapolation to new data is examined, revealing challenges and uncertainties. Model adaptability to other data types, such as conductivity and water level, is explored using Matrix Profile, showing promising prospects for anomaly detection in wastewater datasets.

**Table 54: Overview of model’s strengths and weaknesses for anomaly detection in wastewater data**

<p><b>Autoencoder (semi-supervised)</b></p>	<p><b>High performance</b> : optimal performance, with an F1 score of 0.93 and an MCC of 0.858 under certain conditions.  <b>Possible improvement</b>: optimizing performance with an increase in the database.  <b>False positive reduction</b>: the ensemble approach significantly reduced false positives, achieving a precision of 0.99.</p>	<p><b>Generalization Challenge</b>: The model encountered difficulties in adapting to dynamics beyond the reference site, leading to errors on valid data and requiring site-specific training.</p>
<p><b>Matrix Profile (unsupervised)</b></p>	<p><b>Effectiveness for Low Prevalence of Anomalies</b>: MP proved effective for datasets with a low anomalies rate, without hyperparameter calibration.  <b>Adaptability and Interpretability</b>: The model's ability to detect anomalies in new data types and using an unsupervised approach underscores its adaptability and interpretability.</p>	<p><b>Need for Calibration</b>: For data sets with higher anomaly rates, hyperparameter calibration becomes crucial, and MP may become inadequate beyond around 25% prevalence.  <b>Suboptimal performance in Multivariate Approach</b>: The multivariate approach with turbidity and conductivity showed suboptimal results, indicating model perturbation by the inclusion of conductivity.</p>
<p><b>ResNet (supervised)</b></p>	<p><b>Anomaly Rate Prediction Accuracy</b> : The ResNet model showed significant accuracy in predicting anomaly rates per sequence, with an F1 score of 77%.  <b>In-depth exploration of hyperparameters</b>: Extensive testing of hyperparameters, including time window size, helped optimize model performance.</p>	<p><b>Sensitivity to Data Preprocessing</b> : The model's sensitivity to data preparation, including time window size and classification threshold adjustment, was highlighted.  <b>Generalization challenge</b>: Generalization of the model to other sites showed significant variations in precision and recall</p>



# **Conclusion and perspectives**



# Conclusion and perspectives

---

## Retracing steps : A comprehensive overview

This thesis focused on the validation of wastewater network data and the detection of anomalies using artificial intelligence techniques. Although this discipline has been widely studied in other fields, to our knowledge, it has never been tackled in the context of data generated by a real sewer network. Thus, the first complexity inherent in this research lies in **the creation of the database!**

Indeed, due to regulatory and confidentiality constraints, obtaining an open-access database for this type of structure is arduous. In this context, we used turbidity data from the SMA wastewater network (**Chapter 4**). This data were collected at the same time as the thesis progressed. The choice of this database is based on operational considerations such as regular sensor monitoring with operational feedback and physical redundancy, as well as scientific considerations, as turbidity data presents significant challenges due to its dynamics.

However, to assess the performance of any model, it is crucial to compare it with **the ground truth!**

To this end, it was essential to define a manual data validation approach based on domain expertise, given the absence of formalized guidelines in the literature for the specific validation of this type of data. Nevertheless, this validation task proves tedious, leading to validation errors and remaining subjective, depending on the expert's interpretation, despite our efforts to objectify a large part of the process through a filtering step.

Given these constraints, **how reliable is the labeling of a given expert?**

To answer this question, we set up a validation pool made up of four experts with solid experience in the operation of wastewater systems and trained in such a way as not to compromise their subjectivity (**Chapter 7**). The assessment of annotator agreement relies on addressing the following questions:

- To what extent do the experts concur?
- Does the agreement among experts transcend random chance?
- Does disagreement among experts compromise the reliability of the reference?
- Are there any expert outliers or expert duplication?

By obtaining positive answers to these inquiries through an analysis comparing agreement levels among different experts, it becomes plausible to assert that the outcome from a single expert aligns closely with the ground truth. This premise was maintained throughout the remainder of the study. Additional tests were conducted to assess the model's sensitivity to the chosen reference for evaluation, further substantiating these findings. Nonetheless, it is crucial to acknowledge that this evaluation is contingent upon a fundamental underlying assumption: the reference originates, at least from a part, from the expert domain analysis, serving as a cornerstone for the study's conclusions.

### Benchmarking models

After collecting the data and its labeling, our focus shifted to the selection of models for evaluation. It should be noted that a wide range of anomaly detection models are listed in the literature. In this thesis, we have deliberately opted for an in-depth examination of three specific models. This selection was made with the aim of discerning and comparing the different validation approaches adopted for each of these models.

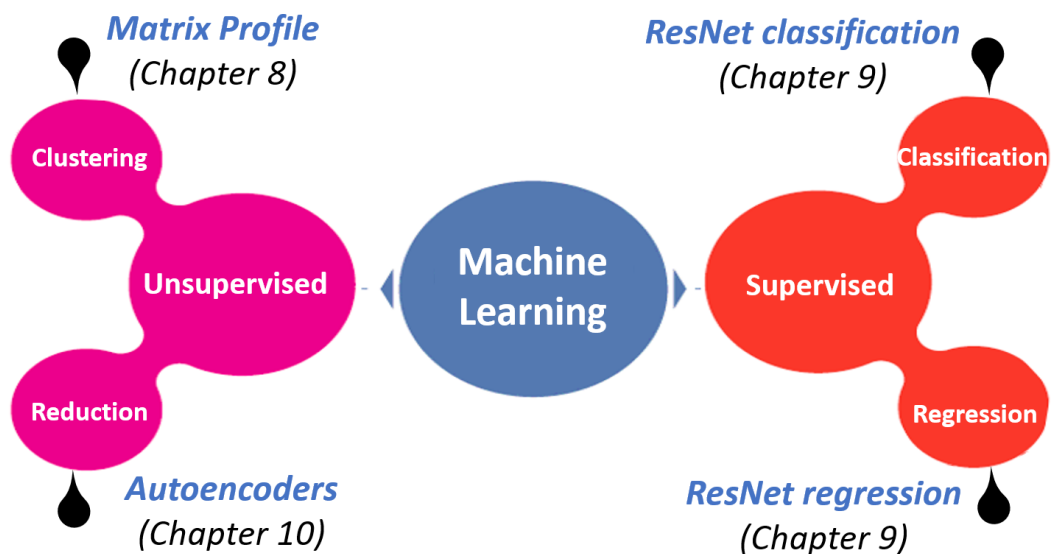


Figure 1: Benchmark of the evaluated models

Table 55 summarizes the optimal results obtained following a process of adjusting hyperparameters and implementing strategies to improve results, aimed at remedying the specific shortcomings of each model. The comparative study of anomaly detection models highlights some interesting results, while underlining the need to consider the specific context and issues associated with each approach.

Table 55: Best results of the evaluated models using 24-hours sequences

Model	Approach	Precision	Recall	F1 score	MCC
<b>Matrix Profile</b>	Reconstructed T with pre-validation	0.81	0.71	0.76	0.71
<b>ResNet</b>	Regression with a threshold at 15%	0.76	0.78	0.77	0.66
<b>Autoencoder</b>	Ensemble model invalidating a sequence at an invalid time step	0.94	0.98	0.96	0.91

Firstly, the autoencoder (AE) stands out for its promising performance, but its use requires a semi-supervised approach. This constraint imposes a pre-analysis of the sequences, to provide a sufficient sample of valid data. Exhaustivity (i.e. labelling each and every available data as either valid or invalid) is not necessary as invalid data are of no use for training purposes. Actually, this sample of valid data can be provided automatically in the case when redundant sensors are implemented. As the depth of the model increases, so does the learning capacity, requiring more input sequences. However, it should be noted that these results are bounded, and beyond a certain complexity (in our case, with three hidden layers) and a defined number of sequences, the model stops learning, or even risks overfitting. Constant vigilance with regard to this constraint is therefore necessary. As output, the autoencoder generates a reconstruction of the input sequence, but validation requires the establishment of a classification threshold. Crucially, the best results are obtained by invalidating a sequence as soon as an invalid time step is detected. However, this approach can be very penalizing in an operational context, underlining the importance of carefully considering these aspects when using the autoencoder in practical applications.

In contrast, the use of the ResNet model is based on a fully supervised approach. This approach proves to be particularly demanding during the learning phase, requiring rigorous manual validation over a representative period. The quality of the model's output depends closely on that of the input, which means that the model learns all the biases and variances inherent in the reference. Due to its supervised nature, the ResNet model is sensitive to imbalance between the two classes. In order to optimize learning, it is necessary to have an enhanced database, thus ensuring equivalent weighting of the two classes during the learning process. However, this assumption can be complex to achieve in the context of anomaly detection, where defects are generally in the minority. Despite these challenges, when all the learning conditions are met, the ResNet model can produce interesting results, particularly with the application of a regression approach. This strategy bypasses the problems associated with



unbalanced classes by providing an anomaly rate per sequence. However, to move on to formal classification, it is necessary to define a classification threshold beyond which a sequence is considered abnormal.

Furthermore, in a totally unsupervised approach, Matrix Profile offers the possibility of skipping the learning phase and selecting the most relevant sequences for this stage. However, this mode of operation has one major limitation: the absence of a pre-saved model that can be directly deployed for new data. Each time it is used, the model must revisit the chronicle (or possibly a representative period) to compare it with the new sequences. While this approach is effective, it loses accuracy as soon as the anomaly rate in the database becomes significant, or when repetitive defects appear. Despite these limitations, the Matrix Profile model remains interesting in terms of calculation time, especially in situations where no reference is available. What's more, its validity has been confirmed by tests on other types of data (conductivity and water level), reinforcing its usefulness in the field of wastewater network data validation.

### And what about the operational dimension?

Let's review the operational context of this study: we are dealing with aggressive and loaded wastewater networks, equipped with multiple sensors. Here, we are focusing in particular on turbidimeters, with a data acquisition frequency of 5 minutes. Globally, wastewater quality varies considerably and rapidly. In the absence of precipitation, we observe characteristic patterns on a daily and weekly scale. During rainy events, these dynamics become more variable, depending on the intensity and duration of precipitation.

On the other hand, the stakes and resources invested in this instrumentation are modest compared to other fields such as air quality or cybersecurity, resulting in a higher risk of failure and lower reliability of the data collected. Thus, we can observe a variety of faults, which can be either measurement errors or contaminated data not representative of the phenomenon of interest.

Conversely, we distinguish between local anomalies, which are brief and often trivial, and sequential anomalies, which are more prolonged and subtle. In this study, models are evaluated to validate data post-hoc, taking into account the various possible anomalies. On the one hand, we seek to perform validation on the scale of each measurement (time step by time step), and on the other to explore validation by sequence, which is more relevant for detecting sequential anomalies. This approach may also be of interest from an operational point of view, particularly when it comes to carrying out daily validation, useful for drawing up compliance and operating reports, for example.

The three models evaluated operate by processing fixed-length sequences, enabling them to take into account the context of each measurement. This approach recognizes that faults in wastewater networks rarely manifest themselves as isolated values, but rather as a series of abnormal values. It should be noted that the length of these sequences can vary in practice. Thus, by globally invalidating sequences of fixed length, the models inevitably generate false positives at a time step of 5 minutes.

To balance this constraint with operational objectives, the evaluation of the various models was based on daily sequences. However, it is important to point out that in some cases, the numerically optimal window size may differ. For all the implementations provided, the models are supported by a pre-validation step which identifies all trivial anomalies (missing data, out-of-range and blocking).

From an operational point of view, the process begins by dividing the data into 24-hours subsequences. The choice of the optimum model for validating turbidity data in a wastewater network depends on the issues at stake and the requirements:

1. If the site operates in a standard way, like our typical site:
  - a. If the aim is to use AI to pre-select faults to be submitted to the expert:
    - i. MP can be used by setting a threshold to identify the number of invalid sequences according to the operator's time constraints. This enables defects to be prioritized. The operator can adjust the number of sequences as required until he finds that the sequences identified are no longer outliers.
    - ii. However, if the site has a high rate of anomalies, the use of MP can be inaccurate and generate many false negatives.
  - b. If the aim is to detect the slightest variation in the input data:
    - i. AE can be used by applying a classification threshold based on MSE of the reconstruction established with the model trained on Cottage data. This threshold invalidates a sequence as soon as a time step is invalid.
    - ii. The expert can re-examine these sequences to validate those that do not appear to be outliers and/or use hardware redundancy, where applicable, to validate the sequences consolidated by the two probes.
  - c. If the objective is to have a an operational, yet accurate validation :
    - i. ResNet can be used with a predictive approach to the anomaly rate per sequence. Switching to binary validation involves applying a threshold at which a sequence is considered invalid, generally around 4.5 hours, based on the results of Cottage data.

- ii. ResNet validation can be supported by redundancy validation to reduce false positives.
2. If the site operates atypically, model recalibration may be required:
  - a. MP can be used to identify the most aberrant anomalies with subsequent prioritization and expert intervention.
  - b. MP can also be used to identify sequences representing patterns in order to feed another model, such as AE. Manual validation by an operator or exploitation of hardware redundancy can also be used to select normal sequences representative of dry and rainy weather operation. A transfer learning can be performed to learn the new normal data, but this requires the establishment of a classification threshold.
  - c. If exhaustive and reliable manual validation has been carried out over a representative period, a transfer learning using the ResNet model can be performed.

For the validation of other types of data, further proofs of concept are required, particularly for the AE and ResNet models, although in theory they may work if the functioning and patterns are similar to those of turbidity. The differences will lie mainly in the classification threshold to be applied. MP has already been tested on water level and conductivity, showing that the limit of this model is generally the imposed anomaly rate, which must be determined by experience or assessed by an operator as they go along.

### Acknowledging the Study's Limitations

At this point of this research work, it is imperative to recognize the limitations inherent in this study to ensure an informed interpretation of the results obtained. Firstly, the need to standardize the input dataset emerges as a crucial consideration for further comparison of anomaly detection models. Currently, the different sizes of the input dataset can introduce potential biases, making direct comparison of model performance complex. By homogenizing the data set, it will be possible to strengthen the validity of comparisons between models, offering more reliable insights into their relative effectiveness. However, each model requires a different selection of input data: Matrix Profile scans the entire input chronicle, while Autoencoders uses only valid sequences for training. Thus, it would be difficult to erase this bias if we want to benefit from the advantages of each model. So, standardization is more a question of the validation set.

Another limitation to consider is the size of the database used, underlining the need for significant expansion, accompanied by a separate validation set. The size of the database has a direct impact on the generalizability of the results obtained. A larger dataset would enable a

more exhaustive exploration of model performance under a variety of conditions, thereby reinforcing the robustness of the conclusions drawn from this study. The introduction of a separate validation set would also help to mitigate the risks of over-fitting the models, ensuring a more accurate assessment of their ability to generalize to new data.

Furthermore, the composition of the validation pool, characterized by an even number of experts, raises questions about consensual decision-making during the evaluation of annotator agreement. The use of an even number of experts can potentially lead to situations of indecision, as it accounts for 15% of the identified anomalies in our case study. It is noteworthy to highlight that having 4 experts remains more advantageous than having 3. These findings do not cast doubt on the reliability of the expert agreement evaluation process, especially since the results demonstrate that the database is highly exploitable in its current state by validating that the opinion of a single expert (which was used for evaluating various models) closely aligns with the unknown ground truth. Nevertheless, for enhanced precision, employing a higher and odd number of experts could be considered.

### Mapping future perspectives

At this stage, three distinct avenues are envisaged in the context of data validation, which map out the landscape of future research. Each of these areas is of particular importance and has its own specific considerations.

#### **Exploring recurrent patterns:**

A significant advance in anomaly detection can result from an evaluation of recurrent neural networks (RNN) and recurrent autoencoders (AE-RNN). This is of particular importance to ensure optimal adaptation of models to the intrinsic temporal nature of the data, thus promoting robust and adaptive performance in anomaly detection.

#### **Data augmentation models:**

In order to improve the robustness of anomaly detection techniques, a thorough exploration of data augmentation models is essential. The integration of these models offers the possibility of enriching limited data sets, thereby enhancing the performance of detection models. By including this dimension, the research aims to increase the generalizability of the models while guaranteeing their ability to handle scenarios where the available data is intrinsically restricted.

#### **Extension to other types of data, especially velocity:**

A major challenge is to extend the application of anomaly detection models to a variety of data types, including velocity, which is a particularly complex parameter in wastewater networks. Exploring this avenue offers the opportunity to broaden the scope of the models, subjecting

them to diversified challenges and thus fostering a deeper understanding of their adaptability. This approach will help strengthen the validity and relevance of detection models in real operational contexts.

Once data validation has been thoroughly explored under various approaches, a crucial question arises regarding the management of invalid data. This opens up a new perspective on the field of:

### **Data reconstruction:**

The challenge here lies in the ability to accurately and reliably restore data sequences identified as invalid or abnormal by validation models. Data reconstruction is positioned as an essential research field, involving the development of sophisticated methods to replace suspicious values with plausible data while preserving the overall integrity of the time series. This goes beyond the simple identification of anomalies, offering a proactive solution for optimal use of data in wastewater management systems. Future perspectives in this area involve the exploration of advanced reconstruction techniques, such as sequence generation models and deep learning approaches. The successful reconstruction of the data would be a significant step forward in improving the quality and usefulness of the information collected, thus consolidating the foundations for effective management of wastewater networks.

In addition, it is imperative to recognize that wastewater systems are constantly evolving entities, characterized by the frequent addition of new connections and new control structures. These structural changes can have a significant impact on normal hydraulic operation, generating a constantly changing dynamic. Thus, the last crucial research track to explore is:

### **Artificial intelligence adaptable to network evolutions:**

The prospect of adaptable artificial intelligence is an innovative and crucial area of research. The potential dynamics of a wastewater network, likely to evolve over time, require models capable of adjusting to new configurations. This concept underlines the need to explore methods for automatically adapting models to structural changes in the network. By anticipating and integrating this flexibility, the research aims to ensure the relevance and effectiveness of anomaly detection models in a constantly evolving operational context.



# Résumé étendu



# Résumé étendu

---

## 1. Introduction

### 1.1. Contexte et enjeux

La surveillance et le suivi du fonctionnement des réseaux d'assainissement revêtent une importance cruciale. D'un point de vue environnemental, la gestion adéquate des eaux usées est essentielle pour prévenir la pollution des milieux naturels engendrée par les rejets urbains. Cet objectif est appuyé par le cadre réglementaire où le suivi des réseaux d'assainissement est exigé pour assurer la conformité aux normes environnementales. Les organismes de régulation (Agences et/ou Police de l'Eau en France par exemple) imposent des critères stricts pour garantir que les rejets d'eaux usées soient quantifiés et maîtrisés. Sur le plan opérationnel, le suivi des réseaux d'assainissement permet d'assurer un fonctionnement efficace des installations et du réseau de collecte. Tandis qu'en parallèle, ce suivi offre une base de données essentielle pour les études techniques et pour la recherche. En somme, suivre de près le fonctionnement des réseaux d'assainissement est une démarche multidimensionnelle qui concilie des préoccupations environnementales, réglementaires, opérationnelles et scientifiques pour garantir une gestion durable [10].

Ainsi, l'utilisation de capteurs joue un rôle central dans cette collecte d'information avec des données portant sur de nombreux paramètres tels que le niveau d'eau, la vitesse d'écoulement et la qualité de l'effluent [11]. Cependant, l'environnement chargé et agressif des réseaux d'assainissement peut poser des problèmes de dysfonctionnement, tels que le colmatage, la corrosion, les défaillances électroniques et autres. Ces défauts se répercutent ainsi sur la qualité des données collectées [14]. Les données invalides peuvent entraîner des conséquences significatives, impactant la fiabilité des informations utilisées dans les études hydrauliques, la conformité réglementaire et la gestion opérationnelle. Il est donc primordial de s'assurer de la fiabilité des données avant toute exploitation.

Les moyens actuels de validation des données impliquent des processus de validation automatisés et/ou manuels. Les premiers sont certes rapides, mais restent généralement superficiels par rapport au panel des éventuelles anomalies, faisant appel principalement à des techniques de traitement de données (calcul de l'écart type, dépassement de seuils, analyse de l'étendue des mesures) [16]. Par ailleurs, la validation manuelle nécessite l'intervention d'un opérateur qualifié pour évaluer de manière globale la vraisemblance des résultats obtenus. Cette approche profite des compétences humaines de reconnaissance des motifs caractéristiques et de mémorisation des défauts passés. Cependant, elle hérite



également de ses carences qui sont la subjectivité de l'opérateur et l'erreur humaine. D'un point de vue opérationnel, cette approche s'avère lente et coûteuse : elle nécessite environ deux heures de temps homme pour la validation d'une chronique de turbidité d'un mois [17].

L'objectif de ce travail de recherche est d'explorer le domaine de l'intelligence artificielle (IA) pour améliorer le processus de validation des données dans les réseaux d'assainissement. Il s'agit d'une discipline d'exploration de données qui est largement étudiée et déployée dans différents domaines, tels que la détection d'intrusions, la détection de fraudes et autres. Elle consiste à identifier des éléments ou événements qui diffèrent significativement des données « normales » (habituelles). Alors que la validation des données par l'IA est courante en hydrologie (cf. [Appendix D](#)), l'application spécifique aux réseaux d'assainissement urbains est un domaine qui n'a pas encore été pleinement évalué. Cette thèse vise donc à explorer les outils d'IA pour affiner le processus de la validation des données dans le contexte des données issues d'un réseau d'assainissement.

### 1.2. Objectifs et contenu de la thèse

La motivation de ce travail de recherche découle du fait que l'installation de capteurs dans les réseaux d'assainissement est devenue une pratique en expansion pour assurer une gestion rigoureuse et une interprétation directe des phénomènes en cours. Cependant, en raison des volumes substantiels de données générées par ces capteurs, collectées à des intervalles de temps variables (allant de la minute à la journée) et acquises dans des environnements difficiles, les approches de validation classiques atteignent leurs limitations [18]. La contribution de cette thèse réside dans son accent sur la validation automatisée des données de mesure d'un réseau d'assainissement via des techniques d'IA. Il est important de souligner que cette validation est exclusivement réalisée à posteriori, une décision justifiée par les besoins prévalents, qui s'opèrent souvent en temps différé.

Pour mettre en œuvre et évaluer ces outils, nous avons utilisé des données de pollution du réseau d'assainissement de la Communauté d'Agglomération de Saint-Malo (SMA). Notre choix de ces données est motivé à la fois par une perspective opérationnelle, les données de SMA représentant une base de données accessible, et par une perspective scientifique, les données de pollution étant parmi les plus difficiles à valider en raison de leur dynamique rapide et fluctuante, en particulier pour les turbidimètres.

La thèse est structurée en trois parties. L'état de l'art examine les méthodologies existantes, les défis et les perspectives de validation des données par l'intelligence artificielle. La section Matériel et méthodes présente la base de données utilisée et une gamme de modèles d'intelligence artificielle à évaluer pour la détection des anomalies. Tandis que la section

Résultats et analyses présente les résultats des tests, l'évaluation des modèles, et une analyse comparative des différentes approches.

## 2. État de l'art

### 2.1. La validation des données en assainissement

La détection de défauts et la validation de données hydrométriques (pollution mais surtout débit) mesurées en continu<sup>14</sup> font l'objet de nombreux travaux de recherche depuis une trentaine d'années [240], [44]. Ils mettent en œuvre des techniques statistiques plus ou moins élaborées, y compris des méthodes de prédiction (filtres de Kalman, ARMA) ou des modèles à base physique permettant de créer et d'exploiter une redondance virtuelle [39], [63], [64].

La mise en œuvre de ces méthodes dans un cadre opérationnel de surveillance des réseaux de collecte reste encore limitée. En pratique, une validation automatique basée sur des caractéristiques locales du signal est utilisée en appliquant des critères assez simples, voire triviaux (valeurs hors gamme ou constantes). Cette (pré-)validation est souvent complétée par une analyse plus globale par un expert humain [16], [240], [241] permettant d'obtenir en temps différé un diagnostic plus poussé.

Si les processus de validation ont été bien formalisés pour la surveillance de la qualité de l'air [188], ils restent encore empiriques dans le domaine de l'hydrométrie urbaine.

#### 2.1.1. Pré-validation des données

L'étape de pré-validation est impérative pour identifier rapidement et automatiquement les anomalies triviales. Elle englobe une série de vérifications standards, notamment la recherche de valeurs manquantes [42], l'identification des mesures hors gamme [16], la détection des anomalies de saturation/blocage [43], et l'identification de gradients importants dans les données [16]. Ces vérifications doivent prendre en compte les capacités spécifiques du capteur en question et des conditions locales de fonctionnement. Dans le cas où un site est équipé avec deux capteurs redondants (par exemple, deux capteurs de turbidité), une analyse comparative de leurs mesures et de leurs comportements est effectuée pour identifier des éventuels divergences. Lorsque la différence dépasse une limite prédéterminée, les données associées sont considérées comme douteuses, nécessitant un examen manuel supplémentaire par un opérateur.

---

<sup>14</sup> A pas de temps courts et réguliers (en moyenne 5 minutes)

Le processus de pré-validation utilise des techniques d'analyse statistique de base et des algorithmes avancés pour identifier des irrégularités locales. Bien que ces tests soient souvent déjà mis en œuvre dans les outils de supervision, il est important de noter qu'ils représentent le strict minimum. La gamme des défauts possibles est bien plus large, incluant des éléments tels que le biais, la dérive, la dégradation de la précision, et d'autres facteurs. La complexité de ces défauts est substantielle, les rendant plus exigeants et difficiles à identifier par des moyens conventionnels. Détecter ces problèmes nécessite donc des approches plus sophistiquées et techniquement avancées [45]. En somme, le processus de pré-validation constitue une étape cruciale dans l'assurance de la qualité des données, préparant le terrain pour des considérations plus approfondies par un opérateur métier.

### **2.1.2. Validation experte des données**

La phase de validation, qui succède à la pré-validation, se distingue par son approche plus globale par rapport à l'évaluation locale des signaux individuels des capteurs. La pré-validation se concentre sur l'analyse de la cohérence des signaux de chaque capteur, tandis que la validation adopte une perspective étendue, impliquant souvent plusieurs capteurs, afin d'identifier des anomalies subtiles et complexes qui pourraient ne pas être évidentes pendant la pré-validation.

L'approche de validation manuelle implique un opérateur compétent, qualifié d'expert, chargé d'examiner les chroniques à la recherche de tendances, de ruptures et de motifs, garantissant ainsi une validation précise et rigoureuse des données. Cependant, la validation manuelle présente des limites, notamment le besoin d'expertise, les coûts en temps et ressources, la subjectivité, les erreurs humaines et la détection limitée des anomalies complexes, surtout dans le cas de grands ensembles de données.

Pour surmonter ces limitations, des approches automatisées, peuvent être adoptées. Des outils statistiques, tels que les méthodes basées sur la distribution des données et les modèles de régression, sont couramment utilisés [48]. Les méthodes basées sur la distribution gaussienne ajustent un modèle statistique aux données, identifiant les anomalies comme des instances présentant une faible probabilité selon le modèle. Les modèles de régression évaluent les résidus pour déterminer les scores d'anomalie. Or, il convient de noter que ces méthodes sont rarement adaptées à la structure spécifique des données issues des réseaux d'assainissement, qui sont non-stationnaires avec des saisonnalités multiples et des séries temporelles partiellement auto-corrélées [66], [69], [73]. Par ailleurs, la modélisation hydraulique, qu'elle soit numérique en 3D (à l'échelle de l'ouvrage) ou 1D (à l'échelle du réseau), offre une compréhension approfondie de la dynamique du système d'assainissement. Les données simulées générées par ces modèles peuvent être comparées aux données

réellement mesurées pour la validation. Cependant, la modélisation présente des limites, notamment les coûts, le besoin d'expertise technique et les erreurs potentielles liées à la structure du modèle et à la définition des conditions aux limites. La modélisation de la pollution, en particulier, reste un défi [74], [75], [76].

Ainsi, la validation des données dans le contexte des réseaux d'assainissement reste un défi complexe, nécessitant la recherche de nouvelles approches mieux adaptées au contexte opérationnel et qui permettent de vérifier la fiabilité des données tout en optimisant les ressources mis en œuvre.

### **2.2. La détection des anomalies avec l'intelligence artificielle**

L'utilisation de l'intelligence artificielle (IA) pour la détection d'anomalies en hydrologie urbaine a été motivée par des incidents majeurs tels que la contamination bactériologique des réseaux d'eau potable en Écosse et en Turquie [115]. Ainsi, la revue de littérature se concentre principalement sur la surveillance de la qualité de l'eau en rivière et dans les réseaux de distribution, avec un accent sur la détection de risques de contamination (cf. Appendix D). Les études liées aux eaux usées représentent une part modeste, se concentrant principalement sur les stations de traitement des eaux usées. Les approches prédominantes dans la détection d'anomalies en hydrologie urbaine peuvent être classées en 3 sous-catégories :

#### **2.2.1. La classification supervisée**

Les approches de classification supervisée reposent sur l'apprentissage d'une frontière discriminante à partir de données étiquetées pendant la phase d'entraînement, puis sur l'utilisation de ce modèle pour classer une instance de test comme normale ou anormale pendant la phase de test. Des modèles traditionnels d'apprentissage supervisé, tels que les machines à vecteurs de support [123], les forêts aléatoires [124] et le k-Nearest Neighbors [127], sont couramment utilisés pour cette tâche. Les modèles traditionnels sont bien documentés mais présentent des controverses quant à leur performance. Ils peuvent souffrir de surajustement et nécessitent une distribution connue des données pour une généralisation efficace [72]. Pour surmonter ces défis, la communauté scientifique se tourne vers des approches plus sophistiquées, comme les réseaux neuronaux profonds.

Les DNN, tels que le perceptron multicouche (MLP) et les réseaux neuronaux convolutifs (CNN), sont de plus en plus adoptés en raison de leur performance. Le MLP est un réseau neuronal entièrement connecté, adapté à la classification temporelle avec l'utilisation de fenêtres glissantes pour maintenir les dépendances temporelles [135]. Les CNN, bien que développés pour le traitement d'images, sont efficaces pour capturer les dépendances temporelles dans les séries chronologiques [134].

### 2.2.2. Les approches non supervisées de clustering

Lorsqu'il n'est pas possible d'accéder à des étiquettes dans les ensembles de données, ce qui est souvent le cas en raison des coûts associés à leur obtention, les approches non supervisées pour la détection d'anomalies sont fréquemment utilisées. Ces algorithmes opèrent sans étiquettes préalables, identifiant des motifs inhabituels dans les données brutes. Les méthodes de détection d'anomalies non supervisées utilisent des modèles tels que One-Class Support Vector Machine [144] et Isolation Forest [148] pour identifier des motifs anormaux.

Ces approches, bien qu'efficaces dans de nombreux cas, peuvent ne pas capturer adéquatement les dépendances temporelles dans les séries chronologiques, limitant leur performance pour détecter des anomalies locales ou influencées par des caractéristiques de haute dimensionnalité [17].

### 2.2.3. Les approches de prédiction

Pour aborder le problème de la détection d'anomalies dans les séries temporelles, une autre approche consiste à utiliser des modèles de prédiction. Au lieu de rechercher directement des anomalies, cette méthode prédit ce qui devrait être attendu dans un ensemble de données et compare ces prédictions avec les données réelles. Si les prédictions correspondent aux données réelles, les observations sont considérées comme valides. Cependant, si les données réelles divergent des prédictions, cela peut indiquer la présence d'anomalies ou d'événements anormaux.

Les réseaux neuronaux récurrents (RNN) sont un choix logique pour cette approche, car ils sont conçus pour traiter des données séquentielles, ce qui les rend adaptés à la modélisation des séries temporelles. Les RNN peuvent prendre en compte la dépendance temporelle des données, essentielle pour détecter des anomalies dans les séquences [156]. Par ailleurs, les autoencodeurs (AE) offrent une alternative intéressante aux RNN pour la détection d'anomalies dans les données de séries temporelles [164]. Ils sont conçus pour reconstruire efficacement des données normales, mais présentent des difficultés pour la reconstruction des données anormales. Bien que ces approches présentent des performances compétitives, leur mise en œuvre peut être complexe et exigeante en termes de ressources de calcul.

Finalement, en dehors du domaine de l'hydrologie urbaine, l'utilisation de modèles d'apprentissage automatique et d'apprentissage profond pour la détection d'anomalies dans les séries temporelles a évolué avec le temps. Certains modèles traditionnels, tels que l'algorithme de clustering K-Means, ont été abandonnés au profit de méthodes plus récentes [171]. D'autres modèles, tels que Matrix Profile (MP), ont été introduits spécifiquement pour

l'exploration de données temporelles. Ce modèle se concentre sur l'identification des parties des données ayant des caractéristiques nettement différentes des autres, les considérant comme potentiellement anormales [195].

Dans le domaine de l'apprentissage profond, des architectures comme les réseaux résiduels (ResNet) ont émergé comme des choix performants pour la détection d'anomalies [134]. Les ResNets se distinguent par leur architecture innovante incorporant des connexions de raccourci linéaires, permettant à l'information de traverser les couches plus efficacement [208]. D'autres approches, comme les réseaux génératifs adverses (GAN), sont également explorées pour leur rôle potentiel dans la détection d'anomalies, en générant des données synthétiques similaires aux séries temporelles réelles et en identifiant les incohérences [185].

Il est important de noter que le domaine de la détection d'anomalies dans les séries temporelles continue d'évoluer, avec de nouvelles méthodes et architectures qui émergent régulièrement. Des combinaisons de différents modèles et approches sont constamment explorées pour répondre aux besoins spécifiques de chaque application.

### 3. Matériel et méthodes

Ayant identifié le besoin crucial de validation des données dans le domaine des eaux usées urbaines et démystifié les spécificités de ces données ainsi que les modèles potentiels montrant des performances prometteuses, ce travail de recherche s'intéresse à l'application et la comparaison de certains modèles d'IA à des données issues d'un réseau d'assainissement à l'échelle réelle.

#### 3.1. Construction de la base de données

Contrairement à des domaines libre d'accès, il est difficile d'obtenir des ensembles de données provenant des services d'assainissement urbain en raison de préoccupations de confidentialité et de la législation en vigueur. Notre première étape consiste donc à construire une base de données fiable, robuste et complète sur laquelle baser nos futurs travaux de validation et d'évaluation des modèles d'IA. Cet objectif a été rendu possible grâce à la contribution de Saint-Malo Agglomération, qui nous a mis à disposition sa base de données, permettant ainsi l'accès à des données opérationnelles réelles.

##### 3.1.1. Réseau de capteurs et disponibilité des données

Le processus de collecte de données a été mené simultanément sur six intercepteurs. Les données sont systématiquement archivées au niveau de la supervision, facilitant la conservation des enregistrements et assurant la disponibilité à long terme des données

historiques. Des inspections régulières et des vérifications de maintenance sont mises en œuvre pour ajuster les performances du système de surveillance. En maintenant ces pratiques de gestion des données et de maintenance, l'intégrité des informations enregistrées est préservée dans le temps, faisant de celles-ci une ressource inestimable pour nos recherches et analyses continues.

Le champ des capteurs à Saint-Malo est très varié. Ceux d'intérêt dans ce travail sont particulièrement les capteurs de pollution ; à savoir les turbidimètres et les conductimètres.

L'objectif principal de la mesure de turbidité est d'évaluer la concentration de la pollution particulaire. SMA a mis en place une redondance de turbidité comme recommandé dans [37]. La redondance améliore la fiabilité des mesures, en particulier en cas de défaillance d'un capteur ou d'anomalies. Elle facilite en outre la validation des données en permettant de valider automatiquement les valeurs fournies par les deux capteurs lorsqu'elles sont cohérentes. La conductivité, quant à elle, est une propriété qui caractérise la capacité d'un matériau ou d'un liquide à conduire l'électricité. Elle est directement liée à la concentration de particules substances dissoutes présentes dans le liquide.

L'exploration approfondie de la base de données de turbidité provenant des intercepteurs de Saint Malo s'est déroulée sur une période étendue, de février 2021 à juillet 2023, avec une fréquence théorique de mesure toutes les 5 minutes. Cependant, des anomalies ont été constatées, telles que des doublons et des fréquences non constantes. Face à ces défis, des étapes de prétraitement ont été entreprises pour assurer la cohérence des données. Dans ce contexte, une élimination des doublons et un rééchantillonnage des données ont été réalisés afin d'homogénéiser la fréquence. De plus, les périodes de données manquantes ont été imputées par des zéros, une approche stratégique pour assurer la continuité des analyses tout en indiquant clairement les moments sans enregistrements (le zéro étant une valeur hors gamme pour la turbidité dans les réseaux d'assainissement).

### **3.1.2. Validation experte des données**

Le processus de validation experte des données de turbidité mis en œuvre dans le cadre de ce travail de recherche se déroule en trois phases: filtrage automatique, expertise manuelle, et agrégation des défauts. Un critère de cohérence est défini pour la validation automatique en prenant en compte la redondance des capteurs de turbidité. La validation experte examine les chroniques mensuelles, détectant des configurations telles que des valeurs nulles, une non-reproductibilité des profils quotidiens par temps sec, un bruit excessif, des pics inhabituels, et une baisse significative de la turbidité. Cette validation combine automatisation et expertise humaine, assurant des gains de temps substantiels grâce à la redondance

physique. L'expertise humaine reste cruciale pour améliorer le taux de disponibilité des données, en validant des mesures identifiées comme douteuses par la validation automatique.

La qualification des anomalies à la fin de ce processus attribue à chaque valeur de turbidité l'une des quatre étiquettes: validée par redondance (R), validée par expertise (V), invalide (I), ou manquante (M). En préparation pour l'utilisation des modèles d'intelligence artificielle, les labels des données sont convertis en une classification binaire. Les données valides et invalides sont regroupées en deux classes distinctes. La sélection du site "Cottage" comme site de référence dans l'étude en cours est motivée par son hydraulique standard et la représentativité des défauts, en faisant un choix pertinent pour l'analyse des modèles.

### **3.1.3. Mise en place d'un pôle de validation**

Bien que la validation experte soit importante, les risques intrinsèques liés à la subjectivité et à l'erreur humaine nous ont poussés à évaluer quantitativement ce biais afin d'estimer son impact sur l'évaluation des modèles d'intelligence artificielle. Ce problème n'est pas spécifique aux données issues des réseaux d'eaux usées. En effet, les annotateurs sont rarement en parfait accord lorsqu'ils expriment leur opinion. Or, l'existence d'une référence de base, souvent appelée "ground truth" est cruciale pour l'apprentissage efficace dans une approche supervisée et pour évaluer rigoureusement les performances des algorithmes. Cependant, obtenir cette référence peut être coûteux, voire impossible.

Pour se rapprocher de cette vérité de référence, une expérimentation a été mise en place, consistant en la constitution d'un pôle de validation impliquant plusieurs experts pour évaluer leur accord et leur variabilité. Cette approche vise à améliorer la transparence et la fiabilité de nos évaluations, en garantissant une compréhension solide de l'interaction entre la validation de base des experts et la performance des modèles d'intelligence artificielle.

Le pôle de validation s'appuie sur une équipe diversifiée de quatre experts. En raison de contraintes budgétaires et temporelles, la multi-validation a été réalisée sur quatre sites différents sur une période de six mois, couvrant l'ensemble de l'année et ses différentes saisons. Les mois sélectionnés ont été choisis avec soin, en évitant les premiers mois d'installation afin d'éliminer tout biais lié à l'optimisation de l'instrumentation. Tous les experts travaillent sur la base des résultats de la phase de filtrage, intervenant spécifiquement lors de la phase d'expertise du processus de validation pour apporter leur expertise à la validation des séquences douteuses.

Pour évaluer la variance de la validation entre les différents experts, diverses mesures sont utilisées pour quantifier l'accord ou le désaccord en considérant différents aspects :



- Coefficient de Kappa de Cohen évalue la concordance entre les annotateurs en tenant compte de la concordance fortuite.
- Score F1 par paire compare les experts entre eux. Les valeurs aberrantes sont identifiées sur la base de la différence moyenne du score F1 entre chaque annotateur et tous les autres, ce qui contribue à l'évaluation de la cohérence et de la fiabilité du groupe d'experts.
- Coefficient de Smyth fournit une approximation de la limite inférieure d'erreur dans les annotations par rapport à la vérité de base inconnue.
- Dendrogramme représente un schéma de regroupement hiérarchique organisant les experts dans une structure arborescente.

Ces mesures, à la fois globales et par paire, contribuent à une compréhension nuancée des résultats de la validation, garantissant une évaluation complète de la fiabilité des annotateurs au sein du groupe de validation. Il faut cependant souligner que cette expérimentation ne visait qu'à évaluer la fiabilité de l'expertise humaine, et que les données de référence labellisées utilisées pour l'apprentissage et l'évaluation des modèles d'IA n'ont été validées que par un expert pour le site type (assisté par la redondance).

### 3.2. Benchmark des modèles et des tests

Cette étude, axée sur la détection d'anomalies dans les données des réseaux d'eaux usées, implique une sélection stratégique de modèles d'apprentissage automatique (ML) et d'apprentissage profond (DL). Les modèles ML supervisés sont exclus en raison de leur sensibilité aux données d'entrée, tandis que l'attention est portée sur les réseaux neuronaux convolutifs (CNN), en particulier ResNet. Les modèles non supervisés sont considérés comme cruciaux, ce qui conduit à l'exploration du modèle Matrix Profile pour le ML traditionnel et des autoencodeurs pour le DL.

#### 3.2.1. Comment chaque modèle répond à notre objectif ?

- **Matrix Profile**

Introduit en 2016, Matrix Profile se distingue comme un algorithme de pointe pour l'analyse des séries temporelles. Son concept fondamental consiste à effectuer une recherche de similarité sur des données de séries temporelles. Plus précisément, l'algorithme permet d'extraire la sous-séquence la plus semblable à chaque sous-séquence (de longueur fixe  $w$ ) incluse dans une série temporelle [195]. En déplaçant la sous-séquence de référence le long de la série temporelle, on peut construire un profil des similitudes trouvées pour chaque sous-séquence dans la série. Les points les plus élevés du profil correspondent à des sous-séquences atypiques [201], dont même les sous-séquences les plus semblables restent

sensiblement différentes. En sélectionnant un nombre  $k$  prédéfini de ces maximums, on identifie les  $k$  sous-séquences les plus atypiques.

Matrix Profile se distingue par sa capacité à fonctionner sans nécessiter de processus d'apprentissage préalable. Le principal avantage de cet algorithme réside dans l'interprétabilité de ses résultats. L'application de MP à la détection d'anomalies implique souvent une analyse visuelle des résultats à l'aide de graphiques, tels que les profils matriciels, et de représentations visuelles telles que les cartes thermiques, afin de repérer les pics associés aux anomalies.

Selon [179], Matrix Profile est caractérisé comme un algorithme sans paramètre, même s'il y a deux paramètres à fixer à savoir la taille de fenêtre  $w$  et le nombre d'anomalies  $k$ . Selon [200], la taille de la fenêtre ( $w$ ) est considérée comme un choix de l'utilisateur plutôt que comme un hyperparamètre, reflétant la connaissance préalable du domaine et indiquant la durée typique d'un motif ou d'une anomalie. De même,  $k$  représente le nombre d'anomalies que l'utilisateur souhaite identifier. Cependant, dans notre cas d'utilisation spécifique, l'objectif est d'identifier toutes les anomalies dans la série temporelle sans avoir de connaissance préalable de la longueur de fenêtre appropriée ou du nombre d'anomalies. Par conséquent, deux hyperparamètres doivent être ajustés : la taille de la fenêtre ( $w$ ) et le nombre d'anomalies ( $k$ ).

Cependant, l'une des limites rencontrées lors de l'utilisation de Matrix Profile est la longueur fixe des anomalies déterminée par la taille de la fenêtre ( $w$ ). Dans les scénarios réels, les anomalies peuvent avoir des durées variables, allant de problèmes durables nécessitant une intervention sur place (par exemple, dysfonctionnement d'un capteur) à des incidents temporaires (par exemple, dépôts sur un capteur nettoyés naturellement par le flux). Les modèles d'ensemble ont été étudiés pour tenir compte de l'hétérogénéité des durées de discordance. Les méthodes d'ensemble visent à combiner plusieurs modèles en un méta-modèle pour améliorer les performances [193].

- **ResNet**

Le modèle ResNet, mis au point par [208], a joué un rôle central dans le projet ImageNet, en s'inspirant de la structure fondamentale des CNNs. Lorsqu'ils sont appliqués à la détection d'anomalies dans les séries temporelles, les modèles ResNet tirent parti de leur capacité à apprendre des représentations complexes et significatives des données temporelles. Pour mettre en œuvre ce processus, l'architecture du modèle ResNet est personnalisée afin d'accepter en entrée une séquence temporelle d'une durée prédéfinie et de produire en sortie l'étiquette de la séquence (valide ou invalide). L'architecture du modèle utilisée dans cette étude s'aligne sur la conception proposée par [133].

Le maintien d'une architecture fixe permet d'isoler et d'évaluer des modifications spécifiques, ce qui donne un aperçu de leurs contributions individuelles à la performance globale du système de détection des anomalies. Des tests de sensibilité sont effectués pour déterminer la taille optimale de la fenêtre d'entrée, en veillant à ce que la fenêtre sélectionnée capture efficacement les modèles et les caractéristiques pertinents des données.

En ce qui concerne la sortie du modèle, une transformation est mise en œuvre pour convertir les étiquettes par pas de temps en une étiquette par séquence. Si plus de 50 % d'une séquence est classé comme invalide, la séquence entière est étiquetée comme anormale. Cette simplification permet au modèle de se concentrer sur la classification de séquences entières plutôt que des pas de temps individuels.

Lors de l'interprétation des résultats du modèle ResNet, l'utilisation des cartes d'activation de classe (CAM) s'avère efficace pour comprendre les résultats de la détection d'anomalies dans une série temporelle. Les CAM permettent de visualiser les segments spécifiques de la série temporelle qui ont contribué de manière significative à la prédiction du modèle, offrant une interprétation granulaire pour comprendre les décisions du modèle.

Cependant l'une des limites du modèle apparaît lors du prétraitement des données. En effet, l'étiquetage initial est transformé en une classification binaire, utilisant un seuil de 50% pour déclarer une séquence valide ou invalide. Or, cette condition semble restrictive, considérant une séquence comme valide même si 49% de ses points ne le sont pas. Cela a conduit à l'exploration d'une approche de classification multi-classes, introduisant les catégories valide, intermédiaire et invalide. Néanmoins, la définition de seuils pour différencier les classes persiste encore. Cette réflexion a permis d'envisager de contourner le problème de la classification en adoptant une approche de prédiction directe du taux d'anomalie dans la séquence, à l'aide d'une approche de régression.

- **Autoencodeur**

Les autoencodeurs, initialement introduits par [213], sont des réseaux largement reconnus, conçus pour reproduire leurs entrées avec une distorsion minimale. Le concept fondamental des autoencodeurs réside dans leur capacité à encoder des informations de manière comprimée et à reconstruire avec précision les données originales à partir de cette représentation latente. Ce processus facilite non seulement l'exploration des structures de données sous-jacentes, mais permet également l'identification de caractéristiques significatives. L'adaptabilité des autoencodeurs à la détection des anomalies provient de leur capacité à capturer des structures de données normales tout en restant sensibles aux variations inhabituelles.

Diverses architectures d'autoencodeurs sont présentées dans la littérature, allant des simples autoencodeurs (AE) aux variantes hybrides telles que les autoencodeurs convolutifs (CNN-AE) ou récurrents (LSTM-AE). Dans notre projet, nous avons choisi d'utiliser des autoencodeurs profonds (Deep-AE). Notre approche explore ainsi différentes architectures de Deep-AE.

L'autoencodeur vise à reconstruire les données d'entrée afin de minimiser la perte d'information, en optimisant la reproduction fidèle des structures normales dans les séquences temporelles. En contrastant l'entrée originale avec sa sortie reconstruite, nous pouvons évaluer l'écart entre la représentation anticipée et celle générée par le modèle. Les anomalies, caractérisées par leur nature inhabituelle ou divergente, peuvent se manifester par des différences substantielles entre l'entrée et la sortie de l'autoencodeur.

Cette approche basée sur le calcul de la fonction de perte, ici l'erreur quadratique moyenne, explore la capacité de l'autoencodeur à apprendre des représentations concises de données normales tout en restant sensible aux variations inhabituelles. Toutefois, il peut s'avérer nécessaire d'ajuster le seuil de classification d'une séquence comme anormale en fonction des caractéristiques spécifiques de l'ensemble de données. Par conséquent, le seuil joue un rôle central dans la prise de décision, ce qui met en évidence l'une des limites de ce modèle, à savoir le défi que représente le calage de ce seuil pour passer des valeurs d'erreur aux classifications de séquences.

### 3.2.2. Phase d'entraînement et d'évaluation

Les données de turbidité du site de Cottage sont utilisées pour l'expérimentation, soumettant les modèles à un processus défini en 6 étapes. Les phases initiales consistent à évaluer la sensibilité aux données d'entrée, en décidant d'utiliser les données brutes mesurées ou la turbidité reconstruite à partir de la redondance. Le prétraitement porte sur la régularité temporelle, les données manquantes et la mise à l'échelle des séries temporelles. En outre, des tests de sensibilité explorent l'utilisation de l'ensemble des données versus la présélection des données.

Le réglage des hyperparamètres est effectué avec une exploration systématique de l'espace de configuration du modèle. Une fois le meilleur modèle identifié, la phase de diagnostic des résultats implique l'analyse et la comparaison de ces derniers avec les conclusions des experts. Cette phase guide des tests visant des potentielles améliorations des résultats. Des approches multivariées sont ensuite explorées, utilisant les données des différents capteurs sur site. Finalement, la généralisation à d'autres sites est évaluée en termes d'adaptabilité et de généralisation, ce qui permet d'obtenir des informations pour une application plus large.

L'évaluation complète conduit à une bonne compréhension des performances de chaque modèle dans divers contextes.

L'évaluation des modèles d'IA pour la détection des anomalies implique de comparer les résultats du modèle à la référence obtenue par la redondance complétée par la validation manuelle. Des mesures clés telles que la matrice de confusion et la courbe ROC sont essentielles pour évaluer les performances. La première fournit une vue détaillée des performances du modèle et permet de comprendre les erreurs de classification pour un seuil donné. En revanche, la courbe ROC évalue les performances du modèle à différents seuils de classification, permettant ainsi des potentiels recalages pour atteindre la meilleure performance du modèle.

### 4. Résultats et discussion

Une fois que la base de données a été établie et que le benchmark des modèles a été réalisé, les résultats des divers tests sont examinés dans cette section. L'accord entre les experts est d'abord évalué, en analysant la cohérence des annotations générées par des experts distincts. Ensuite, l'évaluation des modèles, à savoir Matrix Profile (MP), l'autoencodeur (AE), et ResNet dans la détection d'anomalies, est approfondie. Une comparaison détaillée des performances de chaque modèle est présentée, mettant en lumière leurs points forts, leurs limitations et les défis auxquels ils sont confrontés dans différents contextes. Enfin, la question de la généralisation est abordée, évaluant la capacité des modèles à maintenir des performances stables au-delà de leurs ensembles de données d'entraînement initiaux. L'objectif est de fournir une perspective complète et objective sur la pertinence et la fiabilité de ces modèles dans des scénarios d'application en contexte opérationnel.

#### 4.1. Évaluation de l'accord entre les experts (assistés par la redondance)

Au vu de la subjectivité intrinsèque au processus de validation experte, il est important d'évaluer les biais éventuels à prendre en compte lors de l'évaluation des modèles d'intelligence artificielle (cf. §3.1.3). Dans le cadre de notre base de données test, des matrices de confusion sont créées à partir de comparaisons par paires, montrant des différences dans les taux d'identification des données invalides entre les experts, mais avec une cohérence générale.

Le calcul du F1 Score par paire révèle des scores variant entre les experts, mais avec une moyenne globale de 0.81. En utilisant la différence moyenne dans le F1 Score, aucun expert ne dépasse le seuil critique de 0.22 (moyenne + écart-type des F1 scores), indiquant l'absence

d'experts « anormaux ». Dans l'ensemble, les résultats sont considérés comme satisfaisants, aucun expert n'étant identifié comme s'écartant significativement du consensus du groupe.

En plus, l'analyse du dendrogramme offre des aperçus sur les dynamiques relationnelles entre les experts, mettant en évidence à la fois les convergences et les différences au sein du groupe d'experts, sans interférence de la phase d'apprentissage ou/et de différenciation liée au niveau d'expertise.

Les résultats du coefficient de Kappa de Cohen entre différents experts montrent des scores élevés, indiquant un accord significatif qui est au-delà du hasard. Les résultats sont cohérents avec les analyses de regroupement hiérarchique et du score F1 pair à pair. Cependant, bien que le Kappa de Cohen soit un outil précieux pour mesurer l'accord inter-annotateurs, il présente des limitations inhérentes aux contextes de déséquilibre de classe et de petite taille d'échantillon.

Enfin, l'analyse du coefficient de Smyth offre une évaluation globale de l'accord entre différents experts, se distinguant des mesures axées sur les comparaisons bilatérales. L'estimation de la limite inférieure d'erreur, à 3,5%, se situe bien en dessous de la limite recommandée de 10%, renforçant la fiabilité et la cohérence des annotateurs. Ces faibles taux d'erreur indiquent une fiabilité appréciable et renforcent la crédibilité des évaluations obtenues dans le cadre de cette expérience.

En conclusion, toutes les analyses convergent vers une évaluation robuste et crédible de la performance des experts en détection d'anomalies. Celle-ci peut en partie être attribuée au rôle de la redondance dans le processus de validation. Malgré les défis liés à la diversité des interprétations, les résultats mettent en évidence une cohérence et une fiabilité considérables dans les évaluations, renforçant la confiance dans la qualité du processus d'annotation et fournissant des perspectives enrichissantes pour le développement de modèles d'intelligence artificielle dans le cadre de cette étude.

### **4.2. Évaluation du modèle Matrix Profile**

Matrix Profile est entraîné à l'aide des données de turbidité recueillies à Cottage entre février 2021 et septembre 2021. L'évaluation des performances de MP diffère de celle des modèles traditionnels. Plutôt que d'avoir des ensembles distincts de données d'apprentissage et de test, MP exploite généralement l'ensemble des données disponibles pour l'apprentissage. L'objectif est d'identifier les structures sous-jacentes dans les séries temporelles qui peuvent indiquer des événements anormaux. Par conséquent, le modèle est généralement évalué sur l'ensemble de la période disponible. L'absence de cette distinction dans le contexte de Matrix

Profile s'explique par la nature exploratoire du modèle : il s'agit de découvrir des modèles ou des comportements inhabituels plutôt que de prédire des résultats spécifiques.

Les tests suivants, telles que la sensibilité aux données d'entrée avec le prétraitement et la sélection des données d'entrée, ainsi que l'optimisation des hyperparamètres, sont des éléments cruciaux de la méthodologie. Le processus d'apprentissage et d'évaluation est centré sur l'utilisation de séquences glissantes (la taille de la séquence étant l'un des hyperparamètres du modèle), où un horodatage donné peut recevoir différentes étiquettes. Afin d'interpréter ces résultats, un post-traitement est effectué, permettant à tous les pas de temps constituant une séquence invalide d'être instantanément signalés comme tels. Il n'est donc plus nécessaire d'attribuer des étiquettes aux séquences à la référence et d'exploiter directement les résultats de la validation manuelle, bien que des tests supplémentaires soient envisagés. D'autres améliorations potentielles des résultats incluent l'utilisation de modèles d'ensemble et d'étapes de pré-validation. Cette section explore ensuite des aspects tels que la généralisation à d'autres sites et la détection d'anomalies multivariées.

### **4.2.1. Sensibilité aux données d'entrée**

Les premiers tests concernant les données d'entrée exposent les étapes de prétraitement indispensables pour lancer Matrix Profile. Communes à divers algorithmes de détection d'anomalies pour les séries temporelles, la synchronisation des données pour maintenir un pas de temps constant et la normalisation des données sont cruciales. MP intègre naturellement la standardisation des données, éliminant ainsi le besoin de prétraitement supplémentaire. Afin de garantir une chronique complète des données tout en préservant les phénomènes de perte de données, les valeurs manquantes ont été comblées par des zéros. Pour une identification précise des anomalies, une fréquence d'acquisition suffisamment fine est nécessaire. Il est crucial de ne pas lisser le bruit, car un bruit prononcé dans les mesures de turbidité dans les réseaux d'assainissement est considéré comme une anomalie, indiquant potentiellement un dysfonctionnement de la sonde.

Ensuite, l'objectif est d'analyser la sensibilité du modèle MP à diverses données d'entrée. Cela implique deux aspects : premièrement, évaluer la performance du modèle en utilisant les données brutes de turbidité provenant des deux turbidimètres, et deuxièmement, évaluer la performance du modèle en utilisant la turbidité reconstruite issue de la phase de filtrage et en exploitant la redondance lors de la validation manuelle. Avec des hyperparamètres fixes, les résultats révèlent d'importantes nuances dans la performance du modèle selon les différentes sources de données d'entrée. Tout d'abord, le modèle appliqué aux données brutes montre une plus faible performance par rapport à celui utilisant la turbidité reconstruite. Le réglage spécifique des hyperparamètres pour chaque jeu de données d'entrée montre des disparités

dans les meilleurs hyperparamètres mais confirme la surperformance en utilisant les données de turbidité reconstruites. En effet, la référence utilisée pour évaluer les résultats du modèle dans le cas des données brutes peut présenter un biais dans certaines situations. Cela se produit lorsque les données sont automatiquement invalidées lors de la phase de filtrage en raison du seuil de redondance établi, même si en réalité elles pourraient avoir une structure valide. L'utilisation des données reconstruites permet donc de s'affranchir de cette limite.

### 4.2.2. Sensibilité aux hyperparamètres

Dans le cadre de l'évaluation des performances du modèle Matrix Profile (MP), des tests de réglage des hyperparamètres ont été effectués. Cette phase vise à déterminer la combinaison optimale de la taille de la fenêtre et du taux d'anomalie qui maximisent la performance du modèle pour la détection d'anomalies dans les données de turbidité du réseau d'assainissement.

Dans ce cas test, le modèle MP est plus sensible à la taille de la fenêtre qu'au taux d'anomalie. Avec des fenêtres de petite taille (quelques heures), le calcul du profil matriciel introduit un bruit excessif et une plage de variabilité limitée, où toutes les sous-séquences semblent plus ou moins similaires. Cette configuration entrave l'identification des pics ou des chutes correspondant aux anomalies et aux motifs. Les hyperparamètres optimaux, utilisant la turbidité reconstruite, sont déterminés comme étant  $w=48$  heures et  $k=9,5\%$ .

De manière générale, le modèle MP est particulièrement sensible à ses hyperparamètres. Lorsque l'on utilise une fenêtre de 48 heures, l'identification de défauts plus courts devient difficile. Par conséquent, l'exploration d'une approche multifenêtres pour détecter des anomalies de durées variables devient une piste intéressante. Avec un taux d'anomalie fixe, l'algorithme est contraint de donner la priorité aux anomalies les plus importantes en amplitude, en s'écartant potentiellement de celles identifiées par l'expert.

### 4.2.3. Analyse et diagnostic des résultats

L'une des principales divergences entre le modèle Matrix Profile et la validation manuelle assistée par la redondance apparaît dans la délimitation des anomalies. Les biais inhérents à la validation manuelle, qui découlent de la subjectivité et de la nature complexe des données, rendent difficile la définition précise des anomalies, en particulier pour les défauts subtils et progressifs. En plus, la phase d'agrégation des défauts, absente du post-traitement du modèle MP, contribue aux faux négatifs, pendant les périodes d'anomalies fusionnées.

Par ailleurs, la taille fixe des séquences imposée par Matrix Profile pose des problèmes, car des séquences entières sont considérées comme invalides alors que potentiellement seule une partie est défectueuse, ce qui peut donner lieu à des faux positifs. Les anomalies de courte



durée peuvent échapper à la détection en raison de l'exigence d'une taille d'anomalie importante. Ainsi, trouver un équilibre entre le rappel et la précision reste un défi, d'autant plus que la modification du taux d'anomalie à identifier affecte les deux mesures.

Un cas intrigant concerne les faux positifs, lorsque le modèle identifie une période comme anormale, contrairement à la validation effectuée par l'expert. Selon sa définition, si Matrix Profile signale une sous-séquence comme anormale, cela implique une différence par rapport au reste de la chronique. Ainsi, la disparité entre la validation MP et la validation manuelle provient essentiellement du fait que l'expert valide une sous-séquence en s'appuyant sur des données exogènes supplémentaires, telles que la redondance. Cela souligne la nécessité d'une approche multivariable et de la prise en compte de données supplémentaires pour la validation.

#### 4.2.4. Pistes d'amélioration

Après avoir affiné les hyperparamètres du modèle et obtenu un score F1 de 0,678 en utilisant des données de turbidité reconstruites, une fenêtre de 48 heures et un taux d'anomalie de 9,5 %, le modèle présente des résultats prometteurs tout en étant confronté à des défis intrinsèques, notamment la taille fixe de la fenêtre et l'identification exclusive d'anomalies uniques. Cette section vise à explorer les possibilités d'amélioration des performances.

Tout d'abord, il convient d'examiner les avantages potentiels de la combinaison des résultats à l'aide des données brutes redondantes au lieu de s'appuyer sur la turbidité reconstruite. En pratique, les séquences anormales dans la chronique de turbidité reconstruite correspondent à des périodes de défaillances simultanées dans les deux turbidimètres. En résultat, la combinaison des résultats de validation des deux turbidités brutes, en identifiant uniquement les anomalies communes, donne des résultats inférieurs à ceux obtenus en utilisant directement la turbidité reconstituée. Bien que cette dernière approche nécessite une phase de filtrage, elle s'avère plus efficace.

Deuxièmement, un modèle d'ensemble est utilisé pour examiner des anomalies de durées variables en utilisant plusieurs fenêtres (12 heures, 24 heures et 48 heures). La recherche par grille a permis d'identifier les deux dernières tailles de fenêtre, tandis que la première est choisie pour pouvoir identifier des courtes anomalies. Le vote majoritaire du modèle d'ensemble surpasse les trois sous-modèles mais n'atteint pas le meilleur modèle individuel. Le vote minoritaire, tout en atteignant un rappel de 0,85, conduit à un nombre accru de fausses alertes. Ainsi, chacune des approches d'ensemble utilisées peut avoir une utilité spécifique, mais la performance globale reste inférieure à celle d'un modèle unique avec une fenêtre et un taux d'anomalie optimaux.

Enfin, une étape de pré-validation a été intégrée afin d'améliorer la robustesse du modèle en tenant compte des biais liés à la répétition de défauts courants. Des règles simples sont utilisées pour identifier et invalider les anomalies triviales telles que les données manquantes, les valeurs hors plage et le blocage ou la saturation. Bien que nécessaire d'un point de vue opérationnel, cette étape, tout en améliorant marginalement le score F1 de 0,678 à 0,679, n'améliore pas de manière significative les résultats globaux en raison de la rareté de ces défauts dans l'ensemble de données.

### **4.2.5. Généralisation du modèle**

Afin d'élargir l'applicabilité du modèle MP, il convient d'évaluer la sensibilité de l'algorithme à ses hyperparamètres sur plusieurs points de mesure. Cette évaluation porte sur trois sites distincts caractérisés par des taux d'anomalie réels variables.

Fondamentalement, le modèle MP se révèle efficace pour détecter les anomalies dans les ensembles de données présentant une faible prévalence d'anomalies (moins de 5%), avec des résultats satisfaisants et des scores F1 supérieurs à 80%, ce qui correspond aux observations de Keogh [179]. Dans ces cas, il n'est pas nécessaire de calibrer les hyperparamètres.

Cependant, dans les ensembles de données présentant des taux d'anomalie plus élevés (entre 5% et 25%), les performances du modèle dépendent de la taille de la fenêtre, ce qui nécessite un calage basé sur la connaissance du domaine concernant la saisonnalité intrinsèque des données ou les résultats de la validation par des experts. Dans le contexte des données sur les eaux usées, connues pour leurs dynamiques typiques sur 24 heures, un ordre de grandeur similaire a été identifié à la fois sur Cottage et sur Goutte, entraînant des scores F1 compris entre 65% et 70%.

Néanmoins, lorsque le taux d'anomalie dépasse environ 25%, la méthode MP s'avère inadéquate et ne permet pas d'identifier efficacement les anomalies. Cette limitation découle d'un conflit fondamental avec le principe du modèle, qui repose sur l'unicité des défauts.

### **4.2.6. Approche multivariable**

En ce qui concerne l'intégration d'une entrée multivariable, l'utilisation d'une approche bivariée pour la détection des anomalies à partir des données brutes des deux turbidimètres donne un score F1 de 0,68. Ce score s'aligne sur les performances obtenues grâce à l'approche monovariée utilisant la turbidité reconstruite. Par conséquent, l'approche bivariée souligne l'importance de la redondance, permettant l'entrée directe des deux chroniques brutes dans le modèle sans avoir besoin de l'étape de filtrage pour la définition de la turbidité reconstruite, tout en maintenant des niveaux de performance cohérents.

L'intégration d'une approche multivariée par l'ajout de la conductivité entraîne toutefois une dégradation des résultats. Cela suggère que l'inclusion des données de conductivité perturbe le modèle et n'apporte pas de valeur ajoutée.

Un dernier test consiste à explorer la combinaison de l'approche multivariée avec l'approche d'ensemble, dans l'intention de tirer parti de leurs avantages respectifs. Alors que l'approche multivariée améliore la détection des anomalies en atténuant les faux positifs, l'approche d'ensemble, qui utilise le vote minoritaire, minimise les faux négatifs, ce qui permet d'améliorer l'identification des anomalies. Trouver un compromis entre ces deux approches s'avère difficile, en particulier parce que le modèle d'ensemble repose sur un vote minoritaire appliqué au modèle multivariable, qui, à son tour, est basé sur un vote unanime. Cette nature compétitive rend difficile l'atteinte d'un optimum tout en conservant les avantages de chaque méthode. Par conséquent, cette approche combinée ne donne pas un résultat final amélioré par rapport à une approche monovariée utilisant la turbidité reconstruite ou une approche bivariée avec les deux turbidimètres.

### 4.3. Évaluation du modèle ResNet

Le modèle ResNet fait l'objet d'une évaluation de ses performances, en tenant compte de divers facteurs susceptibles d'influer sur son efficacité. L'architecture de ResNet demeure un point de référence constant tout au long de nos évaluations [133], en utilisant un ensemble de données dérivées des données de turbidité de Cottage couvrant la période de février 2021 à juillet 2022.

Dans un premier temps, une analyse de sensibilité du modèle aux données d'entrée a été réalisée. Cette analyse porte sur différents aspects, notamment les techniques de prétraitement, les stratégies d'augmentation des données et les caractéristiques inhérentes aux données d'entrée. Ensuite, des tests de sensibilité pour évaluer l'impact des hyperparamètres sur les performances du ResNet ont eu lieu, en tenant compte de facteurs tels que la taille de la fenêtre d'entrée et la transformation des probabilités en étiquettes de séquence. Afin d'augmenter les capacités prédictives du modèle, diverses approches sont étudiées, y compris des stratégies de pré-validation et de classification multi-classes. En outre, les avantages potentiels de la prédiction des taux d'anomalie par séquence sont étudiés, ce qui contribue à une exploration approfondie des capacités de ResNet.

En élargissant la portée de notre évaluation, les capacités de généralisation du modèle ResNet sont analysées via des évaluations directes, de l'apprentissage par transfert et/ou les calages spécifiques aux sites, fournissent des informations sur l'adaptabilité du modèle à divers

contextes. Enfin, la détection d'anomalies s'étend à l'échelle multivariable, en sondant la capacité de ResNet à identifier des anomalies à l'aide de variables multiples.

### 4.3.1. Sensibilité aux données d'entrée

En tant que modèle supervisé, le modèle ResNet prend en entrée les données de mesure séquentielles et la classification correspondante. Le prétraitement des données d'entrée respecte ainsi principalement les pratiques standards et s'inspire des tests effectués avec Matrix Profile. Pour garantir la mise à l'échelle des données, un processus de standardisation est appliqué aux données d'entrée. Le pas de glissement des séquences est fixé à la moitié de la taille de la fenêtre, ce qui permet de trouver un équilibre entre l'amélioration des données et l'évitement du surajustement.

Étant donné le processus d'apprentissage, qui implique l'injection d'échantillons des deux classes, et l'objectif principal de détection des anomalies (en se concentrant sur les données invalides), le déséquilibre entre les classes peut poser des problèmes. Par conséquent, l'utilisation des approches d'augmentation des données pour équilibrer les deux classes devient pertinente. Les approches standards telles que le suréchantillonnage, la génération d'anomalies synthétiques et l'apprentissage sensible aux coûts présentent des avantages limités. Une autre approche consiste à exploiter les données disponibles sur d'autres sites, ce qui élimine la nécessité de créer des données artificielles. Les résultats indiquent que cette méthode n'améliore pas significativement les performances par rapport à l'utilisation exclusive des données du site-type, à l'exception de la stabilisation des performances.

Finalement, des tests visant à évaluer la réactivité du modèle à diverses données d'entrée, en particulier les données brutes et la turbidité reconstruite, ont été menés. Peu de distinctions sont observées entre l'utilisation des différents entrants. L'utilisation des séquences dont la validité ou l'invalidité est de 100 %, est effectuée pour neutraliser les biais, montrant ainsi une meilleure performance.

### 4.3.2. Sensibilité aux hyperparamètres

Par défaut, il a été décidé de ne pas modifier l'architecture du modèle ResNet. Par conséquent, les ajustements des hyperparamètres se concentrent sur des éléments autres que l'architecture elle-même, influençant les résultats globaux.

La première considération concerne l'utilisation d'une séquence de 24 heures, ce qui laisse sans réponse la question de savoir s'il s'agit de la taille de fenêtre optimale. L'analyse des résultats révèle que les performances maximales sont atteintes avec une fenêtre de 36 heures, ce qui donne un score F1 de 0,65. Toutefois, au-delà de ce seuil, l'écart-type des résultats devient remarquablement important, ce qui indique une instabilité significative. En l'absence

d'une tendance distincte ou d'une taille de fenêtre exceptionnelle, une taille de fenêtre de 24 heures est considérée pour les tests ultérieurs.

Le deuxième point d'intérêt concerne le seuil de classification. Bien que l'objectif de sortie soit une étiquette par séquence, le modèle génère, pour chaque séquence, une probabilité d'appartenance à chaque classe. La classe finale est déterminée par une probabilité supérieure à 0,5. La pertinence de ce seuil est examinée à l'aide de la courbe ROC et/ou de la courbe PR. Étant donné que l'objectif est de maximiser le score F1, l'analyse rétroactive de la courbe PR permet d'identifier le seuil qui maximise le score F1 sur l'ensemble de la base de données d'apprentissage. L'application d'un seuil de 0,43 à la sortie du modèle ResNet s'avère efficace pour améliorer les performances métriques.

Ainsi, l'ajustement du seuil de classification permet une amélioration significative des résultats, notamment pour les séquences contenant exclusivement des données valides ou invalides, sans nécessiter une précision exagérée dans son établissement. Toutefois, cette approche implique un processus d'apprentissage en deux étapes : tout d'abord, l'apprentissage du modèle en utilisant le F1 score comme métrique sur l'ensemble de validation, puis l'évaluation du modèle sur l'ensemble de données complet. La courbe PR est analysée afin d'identifier le score qui maximise l'aire sous la courbe et, par la suite, ce seuil est imposé comme seuil de classification pour la sortie du modèle, représentant la probabilité d'appartenir à une classe spécifique.

### **4.3.3. Analyse et diagnostic des résultats**

La précision la plus élevée, observée pour la turbidité reconstruite, implique que le modèle est plus susceptible de minimiser les prédictions faussement positives lorsqu'il est appliqué aux données reconstruites qu'aux données brutes. Cette tendance peut être attribuée à la définition du seuil de cohérence par redondance, qui invalide automatiquement la turbidité la plus élevée. Il convient de noter que la présence d'une turbidité élevée n'indique pas nécessairement des données aberrantes, car elle peut partager la même structure qu'une turbidité plus faible, un aspect qui serait évalué par un expert. Ce scénario introduit des erreurs dans la base de données de référence pour les données brutes, ce qui entraîne des faux négatifs.

En plus, les erreurs faussement positives proviennent également du seuil déterminant si une séquence est considérée comme invalide. Ce problème peut concerner la référence : si l'expert n'a invalidé que 140 pas de temps (sur 288), on attribue une étiquette valide à la séquence alors qu'elle représente une anomalie importante. Le modèle invalidant cette séquence aboutirait à un faux positif. Ce problème peut également concerner le résultat de la

validation par le modèle ResNet. En effet, la sortie du réseau comprend des probabilités d'appartenance à chaque classe respective. La classe prédominante est celle dont la probabilité dépasse 0,5 et qui est attribuée à la séquence d'entrée. Cependant, dans certaines situations limites, des probabilités proches de 50-50 peuvent obliger le modèle à prendre une décision basée sur une légère différence en faveur d'une classe par rapport à l'autre.

Par conséquent, l'analyse des résultats nécessite un examen attentif de la base de données de référence. La nature supervisée de l'apprentissage du modèle implique l'assimilation de certains biais présents dans la référence, reproduisant potentiellement ces biais dans les évaluations ultérieures. L'ensemble de ces résultats a favorisé l'utilisation exclusive de séquences 100% valides ou invalides et l'adaptation du seuil de classification.

Par ailleurs, le rappel présente une valeur plus faible, ce qui constitue un problème pour l'identification des défauts, avec une proportion notable d'anomalies omises. Cela est dû en partie à la difficulté d'identifier des anomalies triviales associées à des séquences nulles. Si l'on est exposé à de telles séquences pendant l'apprentissage, la multiplication des valeurs nulles par les poids du réseau neuronal se traduit par une sortie nulle pour chaque neurone. Cela peut entraîner une perte importante d'informations, car les poids associés à ces séquences ne contribuent pas de manière significative à la mise à jour des paramètres du réseau au cours de l'apprentissage. D'où l'intérêt de mettre en œuvre une étape de pré-validation pour les anomalies triviales.

#### **4.3.4. Pistes d'amélioration**

Tout d'abord, une phase de pré-validation du modèle est introduite, visant à aligner la base de données sur ce qui est présenté à l'expert. Cette étape invalide automatiquement les anomalies triviales telles que les données manquantes, les valeurs hors gamme, le blocage ou la saturation. Simultanément, elle valide automatiquement les séquences répondant au critère de redondance. La mise en œuvre de cette pré-validation a permis de réduire considérablement le nombre de faux positifs et d'améliorer les performances du modèle ResNet, qui a obtenu un score F1 de 64 % et un MCC de 56 %.

Deuxièmement, la performance du modèle est évaluée dans le contexte de la classification multiclasse. Celle-ci comprend des séquences nettement classées comme valides ou invalides, ainsi que des séquences intermédiaires marquées par l'incertitude, qui peuvent nécessiter une expertise plus poussée. Au final, il apparaît que cette approche n'améliore pas les résultats, introduisant un biais lié à la définition de la classe intermédiaire. En outre, les résultats ne permettent pas de satisfaire notre intention initiale d'avoir deux classes distinctes certaines, c'est-à-dire valide et non valide.

Une troisième approche consiste à modifier le modèle ResNet pour prédire le taux d'anomalie de chaque séquence d'entrée plutôt que de fournir une probabilité d'anomalie. Cette approche vise, de prime abord, à éliminer la dépendance à l'égard des seuils de classification et des biais introduits par la définition des classes lors de l'apprentissage du modèle. On constate que le résultat dépasse le meilleur score obtenu avec une approche de classification binaire. Par ailleurs, l'utilisation du modèle de régression comme précurseur de la classification permet également d'améliorer significativement les performances.

### **4.3.5. Généralisation du modèle**

L'évaluation du modèle ResNet pour la prédiction sur différents sites de Saint Malo Agglomération révèle des performances nuancées. Le réentraînement du modèle avec des données provenant de divers sites démontre une amélioration globale, bien qu'accompagnée de problèmes spécifiques à certains sites. En effet, il existe des variations de précision entre les sites, malgré une amélioration globale du rappel. Le transfert d'apprentissage par le biais d'un réapprentissage total est également prometteur. Par ailleurs, l'adaptation du modèle au site de Roosevelt, caractérisé par un comportement hydraulique distinct, suggère que la réinitialisation du processus d'apprentissage et le réglage partiel sont des stratégies efficaces pour capturer la structure de données spécifique au site.

### **4.3.6. Approche multivariable**

L'objectif de ce test final est d'évaluer une approche multivariée, où le modèle est alimenté avec diverses mesures collectivement. Malgré une certaine variabilité entre les différents tests, une tendance constante se dégage, indiquant que la performance optimale est généralement atteinte avec un seuil de validation manuelle (cible) de 0,66 et un seuil de modèle de 0,42, en moyenne. Notamment, le seuil du modèle est inférieur au seuil cible, ce qui suggère une tolérance potentielle dans le jugement du modèle par rapport à l'expert. En comparant les scores F1 de différentes configurations, il est observé que l'inclusion de la conductivité a contribué à améliorer la détection des anomalies.

## **4.4. Évaluation du modèle Autoencodeur**

Le fonctionnement de l'autoencodeur est basé sur l'entrée de séquences et la création du modèle qui reproduit la même séquence que la sortie. Les séquences utilisées pour l'apprentissage sont dérivées des données de turbidité de Cottage couvrant de février 2021 à août 2022. Des tests supplémentaires se concentrent sur les variations dans le traitement de ces séquences, y compris les modifications de taille et de sélection (en utilisant l'ensemble complet des séquences ou une présélection). Ensuite, la sortie du modèle est comparée à l'entrée par des calculs d'erreur quadratique moyenne (EQM) pour chaque séquence. Pour la

validation des séquences, un classificateur est appliqué pour établir un seuil, déterminant si une séquence est jugée valide ou non valide.

L'architecture du modèle reste flexible à ce stade et fera l'objet de tests spécifiques pour assurer sa stabilité. Plusieurs exécutions ont lieu et les performances du modèle sont évaluées sur l'ensemble de données de Cottage. Les évaluations finales consistent à évaluer l'efficacité du modèle à l'aide de données provenant de différents sites et à explorer une approche multivariée.

### 4.4.1. Sensibilité aux données d'entrée

Ces premiers tests visent à effectuer des tests de sensibilité à l'entrée sur le modèle autoencodeur (AE). Les résultats de cette approche consistent à utiliser exclusivement des séquences 100% valides pendant la phase d'apprentissage (approche semi-supervisée) et à intégrer la mise à l'échelle par la standardisation.

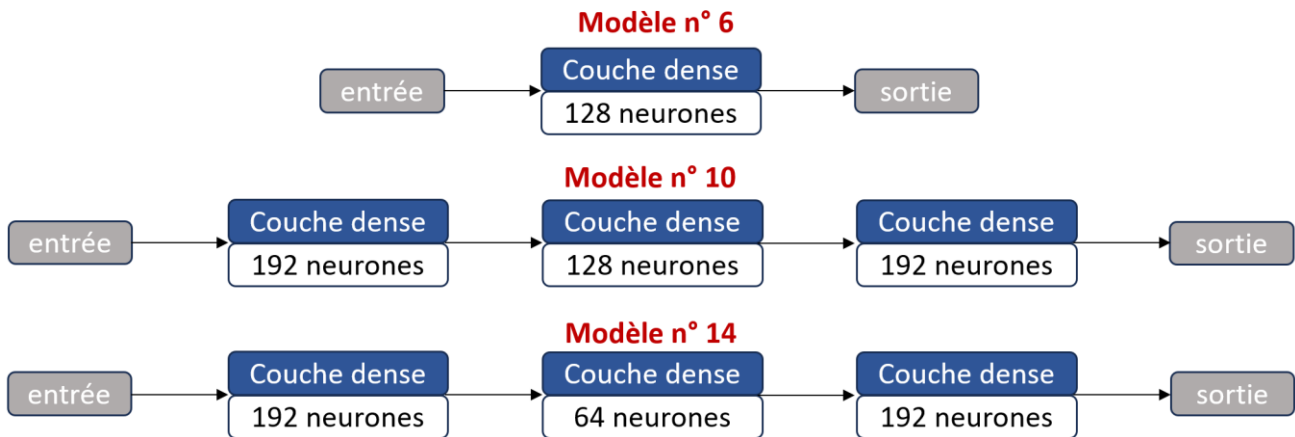
L'objectif du test suivant est d'évaluer comment le modèle répond aux différentes entrées, en particulier les données brutes et la turbidité reconstruite. On observe que les performances ont tendance à se dégrader lors de l'utilisation de données reconstruites. Cette dégradation peut être attribuée au nombre limité d'échantillons invalides, ainsi même de légères erreurs peuvent avoir un impact significatif sur les mesures de performance. Par conséquent, étant donné la taille restreinte de notre ensemble de données, toutes les séquences de T1 et T2 seront utilisées comme entrée pour générer une base de données augmentée.

### 4.4.2. Sensibilité aux hyperparamètres

La sélection stratégique du nombre de neurones et de couches sert d'aspect fondamental dans la conception d'un réseau neuronal, représentant des hyperparamètres cruciaux qui définissent sa capacité et sa complexité. Par conséquent, le réglage de l'architecture du modèle d'autoencodeur profond (AE) devient l'hyperparamètre principal à optimiser.

À la suite de divers tests comparatifs entre les modèles et d'une prise en compte des risques de surajustement associés, il s'avère inutile de dépasser trois couches cachées en raison du nombre limité d'exemples disponibles. Les modèles optimaux, à savoir les modèles 6, 10 et 14 sont illustrés dans [Figure 2](#). En effet, les modèles 10 et 14, tous deux présentent des performances équivalentes avec une structure à trois couches où les premières et troisièmes couches sont chacune composées de 192 neurones. Les deux tailles de code (64 et 128 neurones) démontrent des performances comparables. Étant donné que le rendement d'un modèle est lié à sa complexité, ce qui a une incidence sur le temps d'entraînement, le modèle 14 est considéré comme la deuxième meilleure architecture, après le modèle 6.





**Figure 2: Résultat des meilleurs architectures d'autoencodeur pour la détection des anomalies**

En ce qui concerne la taille des fenêtres, les performances s'améliorent à mesure que la taille de la séquence d'entrée augmente. En effet, des séquences plus grandes sont essentielles pour que l'autoencodeur discerne des caractéristiques distinctives, facilitant ainsi une meilleure reconstruction de séquence. Les résultats deviennent particulièrement intéressants à partir d'une séquence de 6 heures, bien que légèrement inférieurs au maximum atteint avec des séquences de 24 heures.

De plus, une évaluation de la sensibilité du modèle au pas de glissement des séquences appliqué aux données d'entrée révèle une dégradation des résultats avec un pas croissant, malgré l'augmentation efficace du volume de données d'entraînement. En effet, dans certaines situations, cette approche peut introduire une redondance, limitant la diversité des informations pertinentes pour l'apprentissage. Cette duplication compromet la capacité de l'autoencodeur à extraire des caractéristiques significatives, expliquant la dégradation progressive des performances observée.

#### 4.4.3. Analyse et diagnostic des résultats

Compte tenu des performances notables, il est naturel de s'interroger sur l'occurrence potentielle du surajustement, une condition qui pourrait entraver la généralisation du modèle à de nouvelles données. Fait remarquable, au-delà de l'utilisation de trois couches, les performances du modèle présentent une dégradation et une instabilité, avec une augmentation significative de l'écart-type entre les analyses. Ce constat permet de conclure que ces architectures spécifiques ne sont pas bien adaptées à notre contexte applicatif. Plusieurs facteurs contribuent à ces résultats, y compris la complexité excessive du modèle par rapport aux données disponibles, ce qui peut mener à un ajustement excessif ou à une sensibilité accrue aux variations aléatoires des données d'entraînement. Ainsi, les tests de

surapprentissage sur les meilleurs modèles permettent de valider leur utilité et la fiabilité de leurs résultats.

En comparant les modèles 6 et 10, qui partagent le même espace latent, les séquences reconstruites par ces modèles sont examinées. Aucun des deux modèles n'introduit systématiquement de bruit dans les reconstructions. De plus, certaines séquences présentent un accord complet de reconstruction entre les deux modèles. Une analyse des séquences invalides révèle que les deux modèles commettent des erreurs distinctes, chacune étant erronée de différentes manières. Cela souligne que les deux modèles produisent des sorties non identiques pour une séquence d'entrée donnée. Malgré leurs performances similaires, il est évident qu'ils mettent l'accent sur des caractéristiques différentes, traduisant la diversité dans leurs approches de la reconstruction de séquence.

En comparant ensuite les deux meilleures architectures (modèles 6 et 10), en examinant les projections des codes générés par les deux autoencodeurs, des indications sur la similarité des représentations latentes (codes) produites par ces modèles peuvent être obtenues. Les observations des représentations graphiques révèlent des différences dans les visualisations t-SNE des deux codes, indiquant que les deux modèles capturent des structures distinctes au sein des données d'entrée. Par conséquent, il peut être intéressant de tirer parti des avantages des deux modèles via un modèle ensembliste.

L'analyse des résultats montre quelques cas d'erreurs de validation, où le modèle valide des séquences que l'expert juge totalement invalides. Parmi ces cas figurent deux cas de données nulles, où les données étaient initialement manquantes et remplacées par des zéros. Malgré la simplicité de cette erreur, l'autoencodeur (AE), ainsi que d'autres modèles d'apprentissage automatique testés, ne parvient pas à détecter de telles anomalies. L'incorporation d'une étape de pré-validation pourrait efficacement isoler et traiter ces séquences. Un autre scénario implique une séquence avec une variabilité minimale, que le modèle reconstruit avec succès, conduisant à sa validation. Cependant, en analysant rétrospectivement cette séquence, il devient évident que l'expert aurait pu la valider. Plus précisément, il s'agit d'une séquence antérieure de deux jours de défauts que l'expert a invalidée pour assurer une délimitation large des défauts.

#### 4.4.4. Pistes d'amélioration

Après avoir établi les étapes de prétraitement et déterminer les architectures optimales avec une taille de fenêtre d'entrée de 24 heures sans chevauchement, Il est maintenant temps d'explorer les améliorations possibles des résultats par des ajustements aux règles de classification. De plus, il est possible d'étudier des stratégies d'amélioration en combinant les forces des meilleurs modèles et en intégrant des processus de pré-validation.

Une fois que le modèle AE a subi un apprentissage sur des séquences jugées valides à 100% conformément à son principe d'apprentissage, la phase de classification des séquences repose sur l'établissement d'un seuil pour l'erreur carrée moyenne (EQM) de la reconstruction. Ce seuil détermine à quel niveau d'erreur une séquence est considérée comme invalide. Pour cela, deux approches ont été explorées : la règle des 3 sigma et l'approche basée sur la courbe PR. La règle des 3 sigma est traditionnellement utilisée dans les contextes statistiques classiques pour les données suivant une distribution normale connue. Cependant, dans notre contexte non supervisé, où les caractéristiques des données peuvent être plus complexes et moins bien définies, la mise en œuvre de la règle des 3 sigma peut donc poser des défis importants. Inversement, la construction de la courbe PR nécessite une référence, compromettant ainsi la nature non supervisée du modèle AE. Pour remédier à ce problème, un test de sensibilité du modèle au seuil de classification en post-traitement a été effectué. À mesure que le seuil d'invalidation d'une séquence augmente, les mesures de performance du modèle diminuent, ce qui indique une précision décroissante dans la classification des séquences comme non valides. La performance optimale du modèle est obtenue avec le seuil le plus bas, où chaque séquence est invalidée dès qu'un pas de temps est considéré comme anormal.

Par ailleurs, la combinaison des résultats des deux meilleurs modèles, notamment grâce à un consensus basé sur la règle des 3 sigma, est prometteuse. Cette approche maintient une nature non supervisée, éliminant le besoin d'une référence de comparaison tout en atteignant une performance remarquable avec un nombre minimal de faux négatifs (empêchant l'omission de fautes) et un taux de fausses alarmes de 6%.

Enfin, la mise en œuvre d'une étape de pré-validation similaire à celle utilisée pour ResNet - invalidation des anomalies triviales et validation des séquences redondantes - entraîne une réduction significative du nombre de faux positifs. Ces derniers respectent le critère de redondance, permettant l'exploitation de la redondance matérielle qui reste transparente au modèle AE en considérant une approche monovariée.

#### 4.4.5. Généralisation du modèle

Dans une tentative d'élargir l'application de la détection d'anomalies à divers sites, l'évaluation des meilleurs modèles directement sur les données de turbidité de différents sites révèle des F1 scores élevés mais des résultats MCC relativement faibles. En effet, les modèles présentent de nombreuses erreurs sur des données valides, un phénomène attribué à la dynamique distincte d'autres sites. Le fonctionnement normal de ces sites diffère de celui de notre site de référence, "Cottage", ce qui rend difficile la capacité du modèle à reconnaître leurs modèles normaux uniques. Par conséquent, le modèle tend à invalider les séquences qui manquent de similitudes avec celles observées au "Cottage."

Un réapprentissage du modèle à l'aide de séquences valides provenant de divers sites s'avère bénéfique, ce qui se traduit par une amélioration des résultats globaux avec un MCC supérieur à 0,65. Dans ce scénario, le modèle adopte une approche plus générique, mettant l'accent sur l'identification des caractéristiques des anomalies d'une manière globale et commune à travers divers sites plutôt que d'adapter spécifiquement son fonctionnement à un site particulier.

En fin de compte, l'apprentissage de modèles spécifiques pour chaque site augmente encore les performances, donnant des scores F1 proches de 0,95 et un MCC dépassant 0,8. Cela suggère que l'architecture du modèle est généralisable; cependant, atteindre des scores comparables à ceux du "Cottage" nécessite un apprentissage spécifique au site.

#### **4.4.6. Approche multivariable**

L'objectif de cette dernière section est d'évaluer une approche multivariée en utilisant d'une part un modèle à couches denses et d'autre part, un modèle à couches convolutives de manière à garder le maximum de lien entre les différentes variables. L'analyse des résultats du premier modèle montre que ce dernier accorde une plus grande importance à l'une des turbidités brutes. La meilleure performance, bien que subtile, est observée en combinaison avec la turbidité reconstruite, tandis que l'ajout de conductivité ne donne pas d'améliorations significatives.

En revanche, le modèle avec des couches convolutives attribue des poids relativement égaux aux deux variables. Les meilleurs résultats sont obtenus lorsque la conductivité est incluse, et l'ajout de turbidité reconstruite a plutôt tendance à perturber le modèle.

En résumé, T1 et T2 apparaissent comme les deux principales variables porteuses du plus grand nombre d'informations, tandis que l'impact de la turbidité et de la conductivité reconstruites reste relativement limité. De plus, la comparaison de ces résultats avec l'approche monovariée révèle une détérioration des performances, conduisant à la conclusion que cette approche n'est pas très prometteuse dans notre cas.

#### 4.5. Comparaison des différents modèles

Au vu des résultats d'évaluation des différents modèles, il s'avère que les trois modèles ont bien performé dans des contextes spécifiques, mais leurs forces et faiblesses varient en fonction des caractéristiques des données et des conditions d'évaluation.

En ce qui concerne l'autoencodeur (AE), il se distingue par ses performances supérieures, mais son utilisation requiert une approche semi-supervisée, impliquant uniquement des séquences valides. La complexité du modèle, notamment le nombre de couches cachées, nécessite de disposer d'un ensemble de données représentatif, tout en soulignant la limite au-delà de laquelle le modèle risque de ne plus apprendre, voire de présenter un surajustement.

D'autre part, l'utilisation du modèle ResNet repose sur une approche complètement supervisée, exigeant une validation métier rigoureuse et une base de données équilibrée. La sensibilité au déséquilibre entre les classes constitue un défi, mais une approche de régression peut offrir des résultats intéressants, malgré la nécessité de fixer un seuil de classification pour passer de la régression à la classification.

Enfin, l'approche complètement non-supervisée de Matrix Profile présente l'avantage de se dispenser de la phase d'apprentissage, mais elle nécessite de balayer la chronique à chaque utilisation, ce qui peut entraîner une perte de précision en présence d'un taux élevé d'anomalies ou de défauts répétitifs. Néanmoins, cette méthode offre une alternative intéressante en termes de temps de calcul, particulièrement utile en l'absence de référence et a démontré son efficacité dans des contextes tels que la validation des données de conductivité et de hauteur d'eau en réseau d'assainissement.

En conclusion, le choix du modèle dépendra du compromis entre les performances attendues, les contraintes opérationnelles et la disponibilité de données étiquetées, chacun des modèles offrant des avantages spécifiques adaptés à des contextes particuliers.

Table 56: Synthèse des modèles évalués

	Matrix Profile	ResNet	Autoencodeur
<b>Approche</b>	Non-supervisée	Supervisée	Semi-supervisée
<b>Prérequis</b>	Diminution de la précision en présence d'un taux élevé d'anomalies ou de défauts répétitifs.	Un apprentissage exigeant nécessitant une validation métier rigoureuse sur une période représentative.	Une préanalyse métier des séquences est requise, mais l'exhaustivité n'est pas obligatoire.
<b>Limites</b>	Pas de modèle pré-enregistré, nécessité de balayer la chronique à chaque utilisation.	<ul style="list-style-type: none"> <li>Sensible au déséquilibre entre les classes, nécessitant une base de données équilibrée</li> <li>Seuil de classification nécessaire pour passer de la régression à la classification.</li> </ul>	<ul style="list-style-type: none"> <li>Au-delà d'une certaine complexité et d'un certain nombre de séquences, risque de surapprentissage.</li> <li>La définition d'un seuil de classification est nécessaire, et la performance dépend du seuil choisi.</li> </ul>
<b>Performance</b>	F1 score = 0.759 MCC = 0.715	F1 score = 0.770 MCC = 0.658	F1 score = 0.960 MCC = 0.908

#### 4.6. Généralisation et ouverture

L'objectif final de ce travail de recherche est d'établir la robustesse du modèle dans des situations réelles, en prenant en compte la variabilité des données et en explorant sa capacité à généraliser à de nouveaux ensembles de données.

D'une part, l'hypothèse de robustesse de la base de données utilisée pour l'évaluation des différents modèles est vérifiée en comparant les performances du modèle AE en utilisant différentes vérités de terrain fournies par les divers experts. Les résultats montrent que la performance du modèle est stable à travers différents accords d'annotateurs, avec une variabilité similaire à celle des experts humains. Cependant, lors de l'extrapolation du modèle à une nouvelle chronique, une détérioration des performances est observée, ce qui soulève des questions quant à la capacité du modèle à se généraliser à de nouveaux ensembles de données.

D'autre part, l'étude explore l'application des modèles MP pour la détection d'anomalies dans les données de conductivité et de niveau d'eau. Pour les données de conductivité, le modèle est prometteur dans la détection des anomalies, mais les fausses alarmes soulignent la nécessité d'une validation approfondie et d'une sensibilité aux paramètres du modèle. Dans le cas des données sur le niveau de l'eau, le modèle MP fait preuve de stabilité dans l'identification des anomalies, l'importance de l'expertise dans le domaine étant soulignée pour l'interprétation des résultats et le choix des paramètres. Le modèle est considéré comme un outil prometteur pour surveiller et détecter les anomalies dans les ensembles de données issues des réseaux d'assainissement.

### 5. Conclusion

Pour conclure, la validation des données joue un rôle essentiel dans la gestion des réseaux d'eaux usées, étant donné les implications de l'exploitation de ces données. Les approches actuelles, souvent dépourvues d'objectivité et coûteuses, suscitent des interrogations quant à leur efficacité. L'objectif de cette thèse était d'évaluer la capacité des progrès en intelligence artificielle à assurer une validation robuste des données.

Différents modèles, à savoir Matrix Profile, ResNet et l'Auto-encodeur, ont été soumis à des tests approfondis pour répondre à cette question. Les résultats, particulièrement prometteurs pour ce dernier, ont démontré une capacité notable à détecter les séquences anormales dans les séries temporelles, avec un F1 score de 0.96. Les deux autres modèles peuvent être exploités dans des configurations spécifiques. En ce qui concerne Matrix Profile, les tests ont révélé que ce modèle excelle dans une approche totalement non supervisée. Cette caractéristique en fait un choix optimal pour des sites présentant des faibles taux de défaillances. Pour ResNet, les résultats suggèrent que ce modèle peut être utile dans des situations où le site d'étude présente des problèmes plus substantiels, avec un nombre significatif d'anomalies. Cependant, il est crucial de mettre en œuvre une phase de validation manuelle préalable, nécessaire pour l'entraînement du modèle.

Il convient de noter que ces conclusions doivent être nuancées en raison des différences de taille des bases de données et des taux d'anomalies utilisés pour le test. Malgré cela, cette thèse représente une première dans l'évaluation d'outils d'IA à l'échelle des réseaux d'assainissement, fournissant des tendances prometteuses pour l'exploitation de ces techniques. Les perspectives incluent des tests plus approfondis sur des bases de données de validation pour consolider les résultats. Un nouvel axe de développement peut concerner la reconstruction des séquences identifiées comme invalides par les modèles.

# Bibliography

- [1] Direction Départementale des Territoires et de la Mer d'Ille-et-Vilaine, *Plan de Prévention des Risques naturels prévisibles de Submersion Marine (PPRSM)*. 2017.
- [2] Direction Eau, Assainissement, et Développement Durable, *Règlement d'assainissement collectif Saint Malo Agglomération*.
- [3] J. W. Davies, D. Butler, C. J. Digman, and C. Makropoulos, *Urban Drainage*, 4th ed. CRC Press, 2018. doi: [10.1201/9781351174305](https://doi.org/10.1201/9781351174305).
- [4] Association Scientifique et Technique pour l'Eau et l'Environnement (ASTEE), "Mémento Technique - Conception et dimensionnement des systèmes de gestion des eaux pluviales et de collecte des eaux usées," Dec. 2017.
- [5] C. Parent-Raoult and J.-C. Boisson, "Impacts des rejets urbains de temps de pluie (RUTP) sur les milieux aquatiques : État des connaissances," *Rev. Sci. Eau*, vol. 20, no. 2, pp. 229–239, Jun. 2007, doi: [10.7202/015881ar](https://doi.org/10.7202/015881ar).
- [6] A. O. Sojobi and T. Zayed, "Impact of sewer overflow on public health: A comprehensive scientometric analysis and systematic review," *Environ. Res.*, vol. 203, p. 111609, Jan. 2022, doi: [10.1016/j.envres.2021.111609](https://doi.org/10.1016/j.envres.2021.111609).
- [7] Légifrance, *Arrêté du 31 juillet 2020 modifiant l'arrêté du 21 juillet 2015 modifié relatif aux systèmes d'assainissement collectif et aux installations d'assainissement non collectif, à l'exception des installations d'assainissement non collectif recevant une charge brute de pollution organique inférieure ou égale à 1,2 kg/j de DBO5*. Accessed: Oct. 16, 2023. Available: <https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000042413404>
- [8] Directorate-General for Environment, "Proposal for a revised Urban Wastewater Treatment Directive." Accessed: Oct. 16, 2023. Available: [https://environment.ec.europa.eu/publications/proposal-revised-urban-wastewater-treatment-directive\\_en](https://environment.ec.europa.eu/publications/proposal-revised-urban-wastewater-treatment-directive_en)
- [9] H.-N. Lefebvre, B. Prévost, and L. Dotta, "Guide pratique: Mise en œuvre de l'autosurveillance des systèmes d'assainissement des collectivités et des industries - Équipements et contrôles," Agence de l'eau Loire Bretagne, Nov. 2015.
- [10] S. Isel, "Développement de méthodologies et d'outils numériques pour l'évaluation du débit en réseau hydraulique à surface libre," PhD Thesis, Université de Strasbourg, 2014. Available: <https://theses.hal.science/tel-01142996>
- [11] M. Lepot, "Mesurage en continu des flux polluants en MES et DCO en réseau d'assainissement," PhD Thesis, INSA de Lyon. Available: <https://theses.hal.science/tel-00782324>
- [12] Groupe de recherche Rhône-Alpes sur les infrastructures et l'eau, "Autosurveillance des réseaux d'assainissement - Retour d'expérience: points caractéristiques, modélisation, supervision, métrologie," GRAIE, Apr. 2014, p. 84.
- [13] D. Colin, C.-A. Herault and N. Venandet, "Guide Pratique: Mise en place de l'autosurveillance des réseaux d'assainissement," Agence de l'eau Rhin Meuse, Feb. 2016.
- [14] S. Methnani, "Diagnostic, reconstruction et identification des défauts capteurs et actionneurs : application aux station d'épurations des eaux usées," PhD Thesis, Université de Toulon (France) , École nationale d'ingénieurs de Sfax (Tunisie), 2012. Available: <https://theses.hal.science/tel-00843868>
- [15] N. Vernin, H. Gilet, M. Oget, and M. Vialle, "Évolution des méthodes de validation et de valorisation des données d'autosurveillance du réseau d'assainissement du département du Val de Marne," presented at the Colloque Hydrométrie de la SHF, Lyon, 2017.
- [16] M. Mourad and J.-L. Bertrand-Krajewski, "A method for automatic validation of long time series of data in urban hydrology," *Water Sci. Technol.*, vol. 45, no. 4–5, pp. 263–270, Feb. 2002, doi: [10.2166/wst.2002.0601](https://doi.org/10.2166/wst.2002.0601).



- [17] I. Zidaoui *et al.*, "Utilisation de l'intelligence artificielle pour la validation des mesures en continu de la pollution des eaux usées," *Tech. Sci. Méthodes*, vol. 11, pp. 39–51, Nov. 2022, [doi: 10.36904/tsm/202211039](https://doi.org/10.36904/tsm/202211039).
- [18] E. M. Dogo, A. F. Salami, N. I. Nwulu, and C. O. Aigbavboa, "Blockchain and Internet of Things-Based Technologies for Intelligent Water Management System," in *Artificial Intelligence in IoT*, F. Al-Turjman, Ed., in Transactions on Computational Science and Computational Intelligence. , Cham: Springer International Publishing, 2019, pp. 129–150. [doi: 10.1007/978-3-030-04110-6\\_7](https://doi.org/10.1007/978-3-030-04110-6_7).
- [19] Commission nationale de l'informatique et des libertés (CNIL), "Intelligence artificielle, de quoi parle-t-on?" Accessed: Oct. 16, 2023. Available: <https://www.cnil.fr/fr/intelligence-artificielle/intelligence-artificielle-de-quoi-parle-t-on>
- [20] E. M. Dogo, N. I. Nwulu, B. Twala, and C. Aigbavboa, "A survey of machine learning methods applied to anomaly detection on drinking-water quality data," *Urban Water J.*, vol. 16, no. 3, pp. 235–248, Jun. 2019, [doi: 10.1080/1573062X.2019.1637002](https://doi.org/10.1080/1573062X.2019.1637002).
- [21] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly Detection for a Water Treatment System Using Unsupervised Machine Learning," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, New Orleans, LA, Nov. 2017, pp. 1058–1065. [doi: 10.1109/ICDMW.2017.149](https://doi.org/10.1109/ICDMW.2017.149).
- [22] F. Hamioud, "Validation de données débitométriques issues de réseaux d'assainissement," PhD Thesis, Automatique et traitement du signal, l'Institut National Polytechnique de Lorraine, Nancy, 2007. Available: <https://theses.hal.science/tel-00168826>
- [23] J. L. Bertrand-Krajewski, D. Laplace, C. Joannis, and G. Chebbo, "Quelles mesures pour quels objectifs ? Métrologie en réseaux d'assainissement" *Techniques Sciences et Méthodes (TSM)*, vol. 2, Feb. 2001.
- [24] GRAIE - Groupe de travail autosurveillance, "Fiche n°11: Acquisition et transmission des mesures en réseaux d'assainissement." 2012.
- [25] S. S. Young, *Computerized Data Acquisition and Analysis for the Life Sciences: A Hands-on Guide*. Cambridge: Cambridge University Press, 2001. [doi: 10.1017/CBO9780511609558](https://doi.org/10.1017/CBO9780511609558).
- [26] P. Breil, C. Joannis, G. Raimbault, F. Brissaud, and M. Desbordes, "Drainage des eaux claires parasites par les réseaux sanitaire. De l'observation à l'élaboration d'un modèle prototype," *Houille Blanche*, vol. 79, no. 1, pp. 45–58, Feb. 1993, [doi: 10.1051/lhb/1993005](https://doi.org/10.1051/lhb/1993005).
- [27] P. Meylan, A.-C. Favre, and A. Musy, *Hydrologie fréquentielle: une science prédictive*. PPUR presses polytechniques et universitaires romandes, 2008. [ISBN: 978-2-88074-797-8](https://doi.org/978-2-88074-797-8)
- [28] A. Carreño, I. Inza, and J. Lozano, "Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework," *Artif. Intell. Rev.*, vol. 53, pp. 3575 - 3594 Jun. 2020, [doi: 10.1007/s10462-019-09771-y](https://doi.org/10.1007/s10462-019-09771-y).
- [29] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *Assoc. Comput. Mach.*, vol. 54, no. 3, pp. 1 - 32, Feb. 2020, [doi: 10.1145/3444690](https://doi.org/10.1145/3444690).
- [30] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, "Revisiting Time Series Outlier Detection: Definitions and Benchmarks," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021, p. 14. Available: <https://openreview.net/forum?id=r8lvOsnHchr>
- [31] M. Braei and S. Wagner, "Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art," *ArXiv200400433 Cs Stat*, Apr. 2020. Available: <http://arxiv.org/abs/2004.00433>
- [32] K. Golmohammadi and O. R. Zaiane, "Time series contextual anomaly detection for detecting market manipulation in stock market," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Paris, France, Oct. 2015, pp. 1–10. [doi: 10.1109/DSAA.2015.7344856](https://doi.org/10.1109/DSAA.2015.7344856).
- [33] M. A. Hayes and M. A. Capretz, "Contextual anomaly detection framework for big sensor data," *J. Big Data*, vol. 2, no. 1, p. 2, Feb. 2015, [doi: 10.1186/s40537-014-0011-y](https://doi.org/10.1186/s40537-014-0011-y).

- [34] GRAIE - Groupe de travail autosurveillance, "Fiche n°14 : Quel suivi de la qualité et pourquoi ?" Jan. 2018.
- [35] A. Maréchal, "Relations entre caractéristiques de la pollution particulaire et paramètres optiques dans les eaux résiduaires urbaines," PhD Thesis, Vandœuvre-lès-Nancy, INPL, 2000. Available: <https://www.theses.fr/2000INPL052N>
- [36] AFNOR, *NF EN ISO 7027 - 1*. 2016, p. 18.
- [37] P.-A. Versini, C. Joannis, and G. Chebbo, *Guide technique sur le mesurage de la turbidité dans les réseaux d'assainissement*. in Guides et protocoles. Vincennes: ONEMA, 2015.
- [38] Q. Yang and X. Wu, "10 Challenging Problems in Data Mining Research," *Int. J. Inf. Technol. Decis. Mak. IJITDM*, vol. 05, pp. 597–604, Dec. 2006, doi: [10.1142/S0219622006002258](https://doi.org/10.1142/S0219622006002258).
- [39] J. Alferes, P. Poirier, and P. A. Vanrolleghem, "Efficient data quality evaluation in automated water quality measurement stations," in *Managing Resources of a Limited Planet*, Leipzig, Germany, Jul. 2012, p. 9.
- [40] A. J. Fox, "Outliers in Time Series," *J. R. Stat. Soc. Ser. B Methodol.*, vol. 34, no. 3, pp. 350–363, Jul. 1972, doi: [10.1111/j.2517-6161.1972.tb00912.x](https://doi.org/10.1111/j.2517-6161.1972.tb00912.x).
- [41] M. Van Bijnen and H. Korving, "Application and results of automatic validation of sewer monitoring data," presented at the 11th International Conference on Urban Drainage, Edinburgh, Scotland, UK, 2008, p. 9.
- [42] H. Kang, "The prevention and handling of the missing data," *Korean J. Anesthesiol.*, vol. 64, no. 5, pp. 402–406, May 2013, doi: [10.4097/kjae.2013.64.5.402](https://doi.org/10.4097/kjae.2013.64.5.402).
- [43] Q. K. Dang and Y. S. Suh, "Sensor Saturation Compensated Smoothing Algorithm for Inertial Sensor Based Motion Tracking," *Sensors*, vol. 14, no. 5, Art. no. 5, May 2014, doi: [10.3390/s140508167](https://doi.org/10.3390/s140508167).
- [44] J. Alferes and P. A. Vanrolleghem, "Automated data quality assessment: Dealing with faulty on-line water quality sensors," in *International Environmental Modeling and Software Society (iEMSs)*, San Diego, CA, USA, Jun. 2014, p. 8.
- [45] C. K. Yoo, K. Villez, S. W. H. Van Hulle, and P. A. Vanrolleghem, "Enhanced process monitoring for wastewater treatment systems," *Environmetrics*, vol. 19, no. 6, pp. 602–617, Dec. 2007, doi: [10.1002/env.900](https://doi.org/10.1002/env.900).
- [46] J.-Y. Trépos, *La sociologie de l'expertise*. Paris: Presses Universitaires de France - PUF, 1996.
- [47] F. Berrada, S. Bennis, and L. Gagnon, "Validation des données hydrométriques par des techniques univariées de filtrage," *Can. J. Civ. Eng.*, vol. 23, no. 4, pp. 872–892, Aug. 1996, doi: [10.1139/196-895](https://doi.org/10.1139/196-895).
- [48] J.-L. Bertrand-Krajewski and C. Joannis, "Validation et critique des résultats de mesure en hydrologie urbaine," *Houille Blanche*, pp. 60–67, Jul. 2009, doi: [10.1051/lhb/2009028](https://doi.org/10.1051/lhb/2009028).
- [49] F. J. Anscombe and I. Guttman, "Rejection of Outliers," *Technometrics*, vol. 2, no. 2, pp. 123–147, 1960, doi: [10.2307/1266540](https://doi.org/10.2307/1266540).
- [50] V. Kumar, A. Banerjee, and V. Chandola, "Anomaly detection for symbolic sequences and time series data," 2009.
- [51] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: [10.1145/1541880.1541882](https://doi.org/10.1145/1541880.1541882).
- [52] G. A. Young, *Mathematical Statistics: An Introduction to Likelihood Based Inference* Richard J. Rossi John Wiley & Sons, 2018, doi: [10.1111/insr.12315](https://doi.org/10.1111/insr.12315).
- [53] E. Grafarend, "Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models," Jan. 2006.
- [54] V. Barnett, "The Ordering of Multivariate Data," *J. R. Stat. Soc. Ser. Gen.*, vol. 139, no. 3, pp. 318–355, 1976, doi: [10.2307/2344839](https://doi.org/10.2307/2344839).
- [55] V. Barnett and T. Lewis, *Outliers in Statistical Data*. Chichester ; New York: John Wiley & Sons Ltd, 1978.
- [56] R. J. Beckman and R. D. Cook, "Outlier ... .. s," *Technometrics*, vol. 25, no. 2, pp. 119–149, May 1983, doi: [10.1080/00401706.1983.10487840](https://doi.org/10.1080/00401706.1983.10487840).

- [57]B. V. Kini and C. C. Sekhar, "Large margin mixture of AR models for time series classification," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 361–371, Jan. 2013, [doi: 10.1016/j.asoc.2012.08.027](https://doi.org/10.1016/j.asoc.2012.08.027).
- [58]A. M. Bianco, M. García Ben, E. J. Martínez, and V. J. Yohai, "Outlier Detection in Regression Models with ARIMA Errors using Robust Estimates," *J. Forecast.*, vol. 20, no. 8, pp. 565–579, 2001, [doi: 10.1002/for.768](https://doi.org/10.1002/for.768).
- [59]D. Chen, X. Shao, B. Hu, and Q. Su, "Simultaneous wavelength selection and outlier detection in multivariate regression of near-infrared spectra," *Anal. Sci. Int. J. Jpn. Soc. Anal. Chem.*, vol. 21, no. 2, pp. 161–166, Feb. 2005, [doi: 10.2116/analsci.21.161](https://doi.org/10.2116/analsci.21.161).
- [60]G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th Edition. Hoboken, New Jersey: Wiley–Blackwell, 2015.
- [61]D. Andrés, "Introduction to ARIMA models - ML Pills." Accessed: Oct. 24, 2023. Available: <https://mlpills.dev/time-series/introduction-to-arima-models/>
- [62]V.-T.-V. Nguyen and J.-L. Bisson, "Validation en temps réel des données des apports naturels journaliers pour la gestion des réservoirs," *Can. J. Civ. Eng.*, vol. 25, no. 6, pp. 1096–1102, Dec. 1998, [doi: 10.1139/l98-036](https://doi.org/10.1139/l98-036).
- [63]E. Piatyszek, P. Voignier, and D. Grailot, "Fault detection on a sewer network by a combination of a Kalman filter and a binary sequential probability ratio test," *J. Hydrol.*, vol. 230, no. 3–4, pp. 258–268, May 2000, [doi: 10.1016/S0022-1694\(00\)00213-4](https://doi.org/10.1016/S0022-1694(00)00213-4).
- [64]S. Bennis and N. Kang, "Multivariate Technique for Validating Historical Hydrometric Data with Redundant Measurements," *Hydrol. Res.*, vol. 31, no. 2, pp. 107–126, Apr. 2000, [doi: 10.2166/nh.2000.0008](https://doi.org/10.2166/nh.2000.0008).
- [65]H. Xu *et al.*, "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, 2018, pp. 187–196. [doi: 10.1145/3178876.3185996](https://doi.org/10.1145/3178876.3185996).
- [66]R. Conejo, E. Guzmán, and J.-L. Pérez-de-la-Cruz, "Knowledge-Based Validation for Hydrological Information Systems," *Appl. Artif. Intell.*, vol. 21, no. 8, pp. 803–830, Sep. 2007, [doi: 10.1080/08839510701526582](https://doi.org/10.1080/08839510701526582).
- [67]A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007, [doi: 10.1016/j.comnet.2007.02.001](https://doi.org/10.1016/j.comnet.2007.02.001).
- [68]Y. Chen, R. Mahajan, B. Sridharan, and Z.-L. Zhang, "A provider-side view of web search response time," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, in SIGCOMM '13. New York, NY, USA: Association for Computing Machinery, Aug. 2013, pp. 243–254. [doi: 10.1145/2486001.2486035](https://doi.org/10.1145/2486001.2486035).
- [69]A. Saberi, "Automatic outlier detection in automated water quality measurement stations," *Maîtrise en génie électrique*, Université Laval, Québec, Canada, 2015.
- [70]R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking," in *Proceedings of the 6th International Conference*, in Co-NEXT '10. New York, NY, USA: Association for Computing Machinery, Nov. 2010, pp. 1–12. [doi: 10.1145/1921168.1921179](https://doi.org/10.1145/1921168.1921179).
- [71]S. Shanbhag and T. Wolf, "Accurate anomaly detection through parallelism," *IEEE Netw.*, vol. 23, no. 1, pp. 22–28, Jan. 2009, [doi: 10.1109/MNET.2009.4804320](https://doi.org/10.1109/MNET.2009.4804320).
- [72]D. Liu *et al.*, "Opprentice: Towards Practical and Automatic Anomaly Detection Through Machine Learning," in *Proceedings of the 2015 Internet Measurement Conference*, in IMC '15. New York, NY, USA: Association for Computing Machinery, Oct. 2015, pp. 211–224. [doi: 10.1145/2815675.2815679](https://doi.org/10.1145/2815675.2815679).
- [73]H. Motulsky, *Intuitive Biostatistics*, 1st edition. New York: Oxford University Press, 1995.
- [74]Y. Oukid, V. Libaud, and C. Daux, "Apports et enjeux de la modélisation hydraulique 3D pour la conception et la réhabilitation des ouvrages hydrauliques," *Houille Blanche*, vol. 106, no. 3, pp. 55–65, Jun. 2020, [doi: 10.1051/lhb/2020029](https://doi.org/10.1051/lhb/2020029).
- [75]J. Vazquez, "Hydrologie et hydraulique urbaine en réseau d'assainissement." 2016.
- [76]Y. Belghaddar, C. Delenne, N. Chahinian, A. Seriai, and A. Begdouri, "Parametrization of a wastewater hydraulic model under incomplete data constraint," presented at the 14th

- International Conference on Hydroinformatics, Jul. 2022, p. 012053. [doi: 10.1088/1755-1315/1136/1/012053](https://doi.org/10.1088/1755-1315/1136/1/012053).
- [77] N. Maslej *et al.*, “Artificial Intelligence Index Report 2023.,” *AI Index Steer. Comm. Inst. Hum.-Centered AI Stanf. Univ. Stanf. CA*, vol. abs/2310.03715, p. 386, Apr. 2023, [doi: 10.48550/ARXIV.2310.03715](https://doi.org/10.48550/ARXIV.2310.03715).
- [78] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, [doi: 10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433).
- [79] N. Shadbolt, “‘From So Simple a Beginning’: Species of Artificial Intelligence,” *Daedalus*, vol. 151, no. 2, pp. 28–42, 2022.
- [80] J. Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation*, First Edition. San Francisco: W H Freeman & Co, 1976.
- [81] D. Crevier, *Ai: The Tumultuous History Of The Search For Artificial Intelligence*, First Edition. New York, NY: Basic Books, 1993.
- [82] Turing Post, “The Story of AI Winters and What it Teaches Us Today (History of LLMs. Bonus),” Turing Post. Accessed: Oct. 27, 2023. [Online]. Available: <https://www.turingpost.com/p/aiwinters>
- [83] M. Haenlein and A. Kaplan, “A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence,” *Calif. Manage. Rev.*, vol. 61, p. 000812561986492, Jul. 2019, [doi: 10.1177/0008125619864925](https://doi.org/10.1177/0008125619864925).
- [84] D. Silver *et al.*, “Mastering the game of Go without human knowledge,” *Nature*, vol. 550, no. 7676, Art. no. 7676, Oct. 2017, [doi: 10.1038/nature24270](https://doi.org/10.1038/nature24270).
- [85] High-level Expert Group on Artificial Intelligence set up by the European Commission, *Ethics guidelines for trustworthy AI*. LU: Publications Office of the European Union, 2019. Accessed: Oct. 27, 2023. Available: <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>
- [86] S. Bubeck *et al.*, “Sparks of Artificial General Intelligence: Early experiments with GPT-4.” arXiv, Apr. 13, 2023. [doi: 10.48550/arXiv.2303.12712](https://doi.org/10.48550/arXiv.2303.12712).
- [87] A. Kaplan and M. Haenlein, “Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence,” *Bus. Horiz.*, vol. 62, Nov. 2018, [doi: 10.1016/j.bushor.2018.08.004](https://doi.org/10.1016/j.bushor.2018.08.004).
- [88] D. Chaves, “Unsupervised Time Series Outlier Detection,” Aalborg University, Denmark, Rapport Master, Jun. 2021.
- [89] J. C. Giarratano and G. D. Riley, *Expert Systems: Principles and Programming, Fourth Edition*, 4th edition. Boston, Mass: Course Technology, 2004.
- [90] O. Campesato, *Artificial Intelligence, Machine Learning, and Deep Learning*. Mercury Learning and Information, 2020.
- [91] A. Munoz, “Machine Learning and Optimization,” Courant Institute of Mathematical Sciences, 2014.
- [92] R. Kohavi and F. Provost, “Glossary of Terms,” *Mach. Learn.*, vol. 2, pp. 271–274, Jan. 1998, [doi: 10.1023/A:1017181826899](https://doi.org/10.1023/A:1017181826899).
- [93] P. Ongsulee, “Artificial intelligence, machine learning and deep learning,” in *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, Bangkok: IEEE, Nov. 2017, pp. 1–6. [doi: 10.1109/ICTKE.2017.8259629](https://doi.org/10.1109/ICTKE.2017.8259629).
- [94] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, pp. 436–44, May 2015, [doi: 10.1038/nature14539](https://doi.org/10.1038/nature14539).
- [95] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain.,” *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958, [doi: 10.1037/h0042519](https://doi.org/10.1037/h0042519).
- [96] M. A. Nielsen, “Neural Networks and Deep Learning,” 2015, Accessed: Oct. 29, 2023. [Online]. Available: <http://neuralnetworksanddeeplearning.com>
- [97] “Explained: Neural networks,” MIT News | Massachusetts Institute of Technology. Accessed: Oct. 30, 2023. [Online]. Available: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
- [98] E. MALDONADO, S. ARIAS, and J.-L. PAROUTY, “Formation Introduction au Deep Learning,” 2020

- [99] J. Patterson and A. Gibson, *Deep learning: a practitioner's approach*, First edition. Beijing: O'Reilly, 2017.
- [100] J. Wei, "Forget the Learning Rate, Decay Loss." arXiv, Apr. 26, 2019. doi: [10.48550/arXiv.1905.00094](https://doi.org/10.48550/arXiv.1905.00094).
- [101] S. Mezzah and A. Tari, "Practical hyperparameters tuning of convolutional neural networks for EEG emotional features classification," *Intell. Syst. Appl.*, vol. 18, p. 200212, May 2023, doi: [10.1016/j.iswa.2023.200212](https://doi.org/10.1016/j.iswa.2023.200212).
- [102] "Generalization: Peril of Overfitting | Machine Learning," Google for Developers. Accessed: Oct. 30, 2023. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/generalization/peril-of-overfitting>
- [103] N. Baba, "A new approach for finding the global minimum of error function of neural networks," *Neural Netw.*, vol. 2, no. 5, pp. 367–373, Jan. 1989, doi: [10.1016/0893-6080\(89\)90021-X](https://doi.org/10.1016/0893-6080(89)90021-X).
- [104] "Intro to optimization in deep learning: Gradient Descent," Paperspace Blog. Accessed: Oct. 30, 2023. [Online]. Available: <https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/>
- [105] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey." arXiv, Jan. 23, 2019. doi: [10.48550/arXiv.1901.03407](https://doi.org/10.48550/arXiv.1901.03407).
- [106] C. Aggarwal, *Outlier Analysis*. 2013. doi: [10.1007/978-1-4614-6396-2](https://doi.org/10.1007/978-1-4614-6396-2).
- [107] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004, doi: [10.1007/s10462-004-4304-y](https://doi.org/10.1007/s10462-004-4304-y).
- [108] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A Survey on Malware Detection Using Data Mining Techniques," *ACM Comput. Surv.*, vol. 50, no. 3, p. 41:1-41:40, Jun. 2017, doi: [10.1145/3073559](https://doi.org/10.1145/3073559).
- [109] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Comput. Surv.*, vol. 1, no. 1, pp. 1–38, Jan. 2020, doi: [10.1145/3439950](https://doi.org/10.1145/3439950).
- [110] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier Detection for Temporal Data: A Survey," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 2250–2267, Jan. 2014, doi: [10.1109/TKDE.2013.184](https://doi.org/10.1109/TKDE.2013.184).
- [111] Samaneh Sorounejad, Z. Zojaji, R. E. Atani, and A. H. Monadjemi, "A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective." arXiv, Nov. 19, 2016. doi: [10.48550/arXiv.1611.06439](https://doi.org/10.48550/arXiv.1611.06439).
- [112] F. Giannoni, M. Mancini, and F. Marinelli, "Anomaly detection models for IoT time series data," *Eng. Comput. Sci.*, p. 10, Nov. 2018.
- [113] J. Slay and M. Miller, *Lessons Learned from the Maroochy Water Breach*, vol. 253. 2007, p. 82. doi: [10.1007/978-0-387-75462-8\\_6](https://doi.org/10.1007/978-0-387-75462-8_6).
- [114] P. L. Meinhardt, "Water and bioterrorism: preparing for the potential threat to U.S. water supplies and public health," *Annu. Rev. Public Health*, vol. 26, pp. 213–237, 2005, doi: [10.1146/annurev.publhealth.24.100901.140910](https://doi.org/10.1146/annurev.publhealth.24.100901.140910).
- [115] M. Kivanc, M. Yilmaz, and F. Demir, "The Occurrence of Aeromonas in Drinking Water, Tap Water and the Porsuk River," *Braz. J. Microbiol.*, vol. 42, no. 1, pp. 126–131, 2011, doi: [10.1590/S1517-83822011000100016](https://doi.org/10.1590/S1517-83822011000100016).
- [116] S. Russo, A. Disch, F. Blumensaat, and K. Villez, "Anomaly Detection using Deep Autoencoders for in-situ Wastewater Systems Monitoring Data," in *Proceedings of the 10th IWA Symposium on Systems Analysis and Integrated Assessment*, Copenhagen, Denmark, Sep. 2019, p. 7.
- [117] R. A. Cody, B. A. Tolson, and J. Orchard, "Detecting Leaks in Water Distribution Pipes Using a Deep Autoencoder and Hydroacoustic Spectrograms," *J. Comput. Civ. Eng.*, vol. 34, no. 2, p. 04020001, Mar. 2020, doi: [10.1061/\(ASCE\)CP.1943-5487.0000881](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000881).
- [118] S. Seshan, D. Vries, M. V. Duren, A. V. D. Helm, and J. Poinapen, "AI-based validation of wastewater treatment plant sensor data using an open data exchange architecture," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1136, no. 1, p. 012055, Jan. 2023, doi: [10.1088/1755-1315/1136/1/012055](https://doi.org/10.1088/1755-1315/1136/1/012055).

- [119] U. Shafi, R. Mumtaz, H. Anwar, A. M. Qamar, and H. Khurshid, "Surface Water Pollution Detection using Internet of Things," in *2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT)*, Islamabad: IEEE, Oct. 2018, pp. 92–96. [doi: 10.1109/HONET.2018.8551341](https://doi.org/10.1109/HONET.2018.8551341).
- [120] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd edition. New York Chichester Weinheim Brisbane Singapore Toronto: Wiley-Interscience, 2000.
- [121] "Toward Supervised Anomaly Detection | Journal of Artificial Intelligence Research." Accessed: Nov. 02, 2023. [Online]. Available: <https://www.jair.org/index.php/jair/article/view/10802>
- [122] P. Gogoi, B. Borah, and D. K. Bhattacharyya, "Anomaly Detection Analysis of Intrusion Data Using Supervised & Unsupervised Approach.," *JCIT*, vol. 5, pp. 95–110, Feb. 2010, [doi: 10.4156/jcit.vol5.issue1.11](https://doi.org/10.4156/jcit.vol5.issue1.11).
- [123] C. Hsu, C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin," Nov. 2003.
- [124] D. Jalal and T. Ezzedine, "Decision Tree and Support Vector Machine for Anomaly Detection in Water Distribution Networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, Limassol, Cyprus: IEEE, Jun. 2020, pp. 1320–1323. [doi: 10.1109/IWCMC48107.2020.9148431](https://doi.org/10.1109/IWCMC48107.2020.9148431).
- [125] M. Belgiu and L. Drăguț, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016, [doi: 10.1016/j.isprsjprs.2016.01.011](https://doi.org/10.1016/j.isprsjprs.2016.01.011).
- [126] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, [doi: 10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).
- [127] D. Coomans and D. L. Massart, "Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearest neighbour classification by using alternative voting rules," *Anal. Chim. Acta*, vol. 136, pp. 15–27, Jan. 1982, [doi: 10.1016/S0003-2670\(01\)95359-0](https://doi.org/10.1016/S0003-2670(01)95359-0).
- [128] Z. Zhu, Y. Xie, X. Yang, and W. Hu, "A fast anomaly network traffic detection method based on the constrained k-nearest neighbor," in *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2023, pp. 318–323. [doi: 10.1109/Confluence56041.2023.10048869](https://doi.org/10.1109/Confluence56041.2023.10048869).
- [129] V. Vercruyssen, W. Meert, G. Verbruggen, K. Maes, R. Baumer, and J. Davis, "Semi-Supervised Anomaly Detection with an Application to Water Analytics," in *2018 IEEE International Conference on Data Mining (ICDM)*, Singapore: IEEE, Nov. 2018, pp. 527–536. [doi: 10.1109/ICDM.2018.00068](https://doi.org/10.1109/ICDM.2018.00068).
- [130] Y. Qin and Y. Lou, "Hydrological Time Series Anomaly Pattern Detection based on Isolation Forest," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Mar. 2019, pp. 1706–1710. [doi: 10.1109/ITNEC.2019.8729405](https://doi.org/10.1109/ITNEC.2019.8729405).
- [131] F. Muharemi, D. Logofătu, C. Andersson, and F. Leon, "Approaches to Building a Detection Model for Water Quality: A Case Study," in *Modern Approaches for Intelligent Information and Database Systems*, Springer., vol. 769, A. Sieminski, A. Koziarkiewicz, M. Nunez, and Q. T. Ha, Eds., in *Studies in Computational Intelligence*, vol. 769. , Cham: Springer International Publishing, 2018, pp. 173–183. [doi: 10.1007/978-3-319-76081-0\\_15](https://doi.org/10.1007/978-3-319-76081-0_15).
- [132] F. Muharemi, D. Logofătu, and F. Leon, "Machine learning approaches for anomaly detection of water quality on a real-world data set," *J. Inf. Telecommun.*, vol. 3, no. 3, pp. 294–307, Jul. 2019, [doi: 10.1080/24751839.2019.1565653](https://doi.org/10.1080/24751839.2019.1565653).
- [133] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline." arXiv, Dec. 14, 2016. [doi: 10.48550/arXiv.1611.06455](https://doi.org/10.48550/arXiv.1611.06455).
- [134] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Min. Knowl. Discov.*, May 2019, [doi: https://doi.org/10.1007/s10618-019-00619-1](https://doi.org/10.1007/s10618-019-00619-1).

- [135] L. Perelman, J. Arad, M. Housh, and A. Ostfeld, "Event Detection in Water Distribution Systems from Multivariate Water Quality Time Series," *Environ. Sci. Technol.*, vol. 46, no. 15, pp. 8212–8219, Aug. 2012, [doi: 10.1021/es3014024](https://doi.org/10.1021/es3014024).
- [136] G. A. C. Cordoba, L. Tuhovčák, and M. Tauš, "Using Artificial Neural Network Models to Assess Water Quality in Water Distribution Networks," *Procedia Eng.*, vol. 70, pp. 399–408, Jan. 2014, [doi: 10.1016/j.proeng.2014.02.045](https://doi.org/10.1016/j.proeng.2014.02.045).
- [137] E. Haselsteiner and G. Pfurtscheller, "Using time-dependent neural networks for EEG classification," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 457–463, Dec. 2000, [doi: 10.1109/86.895948](https://doi.org/10.1109/86.895948).
- [138] C. Szegedy *et al.*, "Going Deeper with Convolutions." arXiv, Sep. 16, 2014. [doi: 10.48550/arXiv.1409.4842](https://doi.org/10.48550/arXiv.1409.4842).
- [139] H. Wu and X. Gu, "Max-Pooling Dropout for Regularization of Convolutional Neural Networks." arXiv, Dec. 04, 2015. [doi: 10.48550/arXiv.1512.01400](https://doi.org/10.48550/arXiv.1512.01400).
- [140] X. Chen, F. Feng, J. Wu, and W. Liu, "Anomaly detection for drinking water quality via deep biLSTM ensemble," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Kyoto Japan: ACM, Jul. 2018, pp. 3–4. [doi: 10.1145/3205651.3208203](https://doi.org/10.1145/3205651.3208203).
- [141] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019, [doi: 10.1109/ACCESS.2018.2886457](https://doi.org/10.1109/ACCESS.2018.2886457).
- [142] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: a comprehensive evaluation," *Proc. VLDB Endow.*, vol. 15, no. 9, pp. 1779–1797, May 2022, [doi: 10.14778/3538598.3538602](https://doi.org/10.14778/3538598.3538602).
- [143] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001, [doi: 10.1162/089976601750264965](https://doi.org/10.1162/089976601750264965).
- [144] R. Zhang, S. Zhang, S. Muthuraman, and J. Jiang, "One Class Support Vector Machine for Anomaly Detection in the Communication Network Performance Data," in *5th WSEAS Int.*, Tenerife, Spain, Dec. 2007, p. 7.
- [145] S. Fang, W. Sun, and L. Huang, "Anomaly Detection for Water Supply Data using Machine Learning Technique," *J. Phys. Conf. Ser.*, vol. 1345, no. 022054, p. 7, Nov. 2019, [doi: 10.1088/1742-6596/1345/2/022054](https://doi.org/10.1088/1742-6596/1345/2/022054).
- [146] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-Based Anomaly Detection," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, p. 3:1-3:39, Mar. 2012, [doi: 10.1145/2133360.2133363](https://doi.org/10.1145/2133360.2133363).
- [147] M. Christensen, "Time Series Outlier Detection," Master Project, Aalborg University, Denmark, Jun. 2021.
- [148] Y. Weng and L. Liu, "A Sequence Anomaly Detection Approach Based on Isolation Forest Algorithm for Time-Series," in *High-Performance Computing Applications in Numerical Simulation and Edge Computing*, C. Hu, W. Yang, C. Jiang, and D. Dai, Eds., in Communications in Computer and Information Science. Singapore: Springer, 2019, pp. 198–207. [doi: 10.1007/978-981-32-9987-0\\_17](https://doi.org/10.1007/978-981-32-9987-0_17).
- [149] Y. Yan, "Anomaly Detection for Water Quality Data," Master of Science - Computing and Software, McMaster University, Hamilton, Ontario, 2019.
- [150] I. Steinwart, D. Hush, and C. Scovel, "A Classification Framework for Anomaly Detection," *J. Mach. Learn. Res.*, vol. 6, no. 8, pp. 211–232, 2005.
- [151] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," p. 12.
- [152] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," in *Proceedings of the Third SIAM International Conference on Data Mining*, San Fransisco, CA, USA: Society for Industrial and Applied Mathematics, May 2003, pp. 25–36. [doi: 10.1137/1.9781611972733.3](https://doi.org/10.1137/1.9781611972733.3).
- [153] S. Oehmcke, O. Zielinski, and O. Kramer, "Event Detection in Marine Time Series Data," in *KI 2015: Advances in Artificial Intelligence*, vol. 9324, S. Hölldobler, R. Peñaloza,

- and S. Rudolph, Eds., in *Lecture Notes in Computer Science*, vol. 9324. , Cham: Springer International Publishing, 2015, pp. 279–286. [doi: 10.1007/978-3-319-24489-1\\_24](https://doi.org/10.1007/978-3-319-24489-1_24).
- [154] N. Mokuwa, C. wa Maina, and H. Kiragu, “Anomaly Detection for Raw Water Quality – A Comparative Analysis of the Local Outlier Factor Algorithm and the Random Forest Algorithms,” *Int. J. Comput. Appl.*, vol. 174, no. 26, pp. 47–54, Mar. 2021, [doi: 10.5120/ijca2021921196](https://doi.org/10.5120/ijca2021921196).
- [155] D. T. Ramotsoela, G. P. Hancke, and A. M. Abu-Mahfouz, “Attack detection in water distribution systems using machine learning,” *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 13, p. 22, Dec. 2019, [doi: 10.1186/s13673-019-0175-8](https://doi.org/10.1186/s13673-019-0175-8).
- [156] A. Tealab, “Time series forecasting using artificial neural networks methodologies: A systematic review,” *Future Comput. Inform. J.*, vol. 3, no. 2, pp. 334–340, Dec. 2018, [doi: 10.1016/j.fcij.2018.10.003](https://doi.org/10.1016/j.fcij.2018.10.003).
- [157] R. Pascanu, T. Mikolov, and Y. Bengio, “Understanding the exploding gradient problem,” *ArXiv*, Nov. 2012,
- [158] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training Recurrent Neural Networks.” *arXiv*, Feb. 15, 2013. [doi: 10.48550/arXiv.1211.5063](https://doi.org/10.48550/arXiv.1211.5063).
- [159] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning Precise Timing with LSTM Recurrent Networks”.
- [160] V. Fehst, H. C. La, T.-D. Nghiem, B. E. Mayer, P. Englert, and K.-H. Fiebig, “Automatic vs. manual feature engineering for anomaly detection of drinking-water quality,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, Kyoto Japan: ACM, Jul. 2018, pp. 5–6. [doi: 10.1145/3205651.3208204](https://doi.org/10.1145/3205651.3208204).
- [161] M. Leyli-Abadi, L. Labiod, and M. Nadif, “Denoising Autoencoder as an Effective Dimensionality Reduction and Clustering of Text Data,” in *Advances in Knowledge Discovery and Data Mining*, J. Kim, K. Shim, L. Cao, J.-G. Lee, X. Lin, and Y.-S. Moon, Eds., in *Lecture Notes in Computer Science*. Cham: Springer International Publishing, 2017, pp. 801–813. [doi: 10.1007/978-3-319-57529-2\\_62](https://doi.org/10.1007/978-3-319-57529-2_62).
- [162] M. Sakurada and T. Yairi, “Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction,” in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, Gold Coast Australia QLD Australia: ACM, Dec. 2014, pp. 4–11. [doi: 10.1145/2689746.2689747](https://doi.org/10.1145/2689746.2689747).
- [163] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, “Learning Discriminative Reconstructions for Unsupervised Outlier Removal,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1511–1519. [doi: 10.1109/ICCV.2015.177](https://doi.org/10.1109/ICCV.2015.177).
- [164] M. R. Gauthama Raman, W. Dong, and A. Mathur, “Deep autoencoders as anomaly detectors: Method and case study in a distributed water treatment plant,” *Comput. Secur.*, vol. 99, p. 102055, Dec. 2020, [doi: 10.1016/j.cose.2020.102055](https://doi.org/10.1016/j.cose.2020.102055).
- [165] I. T. Nicholaus, J. R. Park, K. Jung, J. S. Lee, and D.-K. Kang, “Anomaly Detection of Water Level Using Deep Autoencoder,” *Sensors*, vol. 21, no. 19, Art. no. 19, Jan. 2021, [doi: 10.3390/s21196679](https://doi.org/10.3390/s21196679).
- [166] Z. Chunkai and Y. Chen, *Time Series Anomaly Detection with Variational Autoencoders*. 2019.
- [167] E. Aronsson, “Unsupervised Anomaly Detection in Multivariate Time Series Using Variational Autoencoders,” *Master Thesis, Math. Sci.*, 2023, Available: <http://lup.lub.lu.se/student-papers/record/9135876>
- [168] S. E. Chandy, A. Rasekh, Z. A. Barker, B. Campbell, and M. E. Shafiee, “Detection of Cyber-Attacks to Water Systems through Machine-Learning-Based Anomaly Detection in SCADA Data,” in *World Environmental and Water Resources Congress 2017*, Sacramento, California: American Society of Civil Engineers, May 2017, pp. 611–616. [doi: 10.1061/9780784480625.057](https://doi.org/10.1061/9780784480625.057).
- [169] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” 1967.
- [170] “k-means clustering,” *Wikipedia*. Accessed: Nov. 01, 2023. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=K-means\\_clustering&oldid=1179749606](https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=1179749606)



- [171] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: implications for previous and future research," *Knowl. Inf. Syst.*, vol. 8, no. 2, pp. 154–177, Aug. 2005, [doi: 10.1007/s10115-004-0172-7](https://doi.org/10.1007/s10115-004-0172-7).
- [172] R. Fujimaki, S. Hirose, and T. Nakata, "Theoretical Analysis of Subsequence Time-Series Clustering from a Frequency-Analysis Viewpoint," in *Proceedings of the 2008 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, Apr. 2008, pp. 506–517. [doi: 10.1137/1.9781611972788.46](https://doi.org/10.1137/1.9781611972788.46).
- [173] T. Ide, *Why Does Subsequence Time-Series Clustering Produce Sine Waves?*, vol. 4213. 2006, p. 222. [doi: 10.1007/11871637\\_23](https://doi.org/10.1007/11871637_23).
- [174] M. Ohsaki, M. Nakase, and S. Katagiri, "Analysis of Subsequence Time-Series Clustering Based on Moving Average," in *2009 Ninth IEEE International Conference on Data Mining*, Dec. 2009, pp. 902–907. [doi: 10.1109/ICDM.2009.147](https://doi.org/10.1109/ICDM.2009.147).
- [175] J. R. Chen, "Making subsequence time series clustering meaningful," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Nov. 2005, p. 8, [doi: 10.1109/ICDM.2005.91](https://doi.org/10.1109/ICDM.2005.91).
- [176] T. Rakthanmanon, E. J. Keogh, S. Lonardi, and S. Evans, "Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data," in *2011 IEEE 11th International Conference on Data Mining*, Dec. 2011, pp. 547–556. [doi: 10.1109/ICDM.2011.146](https://doi.org/10.1109/ICDM.2011.146).
- [177] S. Zolhavarieh, S. Aghabozorgi, and Y. W. Teh, "A Review of Subsequence Time Series Clustering," *Sci. World J.*, vol. 2014, pp. 1–19, Jul. 2014, [doi: 10.1155/2014/312521](https://doi.org/10.1155/2014/312521).
- [178] H. Borges, R. Akbarinia, and F. Masegla, "Anomaly Detection in Time Series," vol. LNCS. TLDKS-12930, 2021, p. 46. [doi: 10.1007/978-3-662-64553-6\\_3](https://doi.org/10.1007/978-3-662-64553-6_3).
- [179] C.-C. M. Yeh *et al.*, "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Barcelona, Spain: IEEE, Dec. 2016, pp. 1317–1322. [doi: 10.1109/ICDM.2016.0179](https://doi.org/10.1109/ICDM.2016.0179).
- [180] I. Fernandez, A. Villegas, E. Gutierrez, and O. Plata, "Accelerating time series motif discovery in the Intel Xeon Phi KNL processor," *J. Supercomput.*, vol. 75, pp. 1–23, Nov. 2019, [doi: 10.1007/s11227-019-02923-5](https://doi.org/10.1007/s11227-019-02923-5).
- [181] A. Beattie, "Detecting temporal anomalies in time series data utilizing the matrix profile," Mechatronic System Design Master's Thesis, LUT University, 2022. Available: <https://lutpub.lut.fi/handle/10024/164072>
- [182] R. Wankhedkar and S. K. Jain, "Motif Discovery and Anomaly Detection in an ECG Using Matrix Profile," in *Progress in Advanced Computing and Intelligent Engineering*, C. R. Panigrahi, B. Pati, P. Mohapatra, R. Buyya, and K.-C. Li, Eds., in *Advances in Intelligent Systems and Computing*. Singapore: Springer, 2021, pp. 88–95. [doi: 10.1007/978-981-15-6584-7\\_9](https://doi.org/10.1007/978-981-15-6584-7_9).
- [183] A. Beattie *et al.*, "A Robust and Explainable Data-Driven Anomaly Detection Approach For Power Electronics," in *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Oct. 2022, pp. 296–301. [doi: 10.1109/SmartGridComm52983.2022.9961002](https://doi.org/10.1109/SmartGridComm52983.2022.9961002).
- [184] P. Mahajan, "Understanding ResNet Architecture," Analytics Vidhya. Accessed: Nov. 06, 2023. [Online]. Available: <https://medium.com/analytics-vidhya/understanding-resnet-architecture-869915cc2a98>
- [185] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks," *2020 IEEE Int. Conf. Big Data*, pp. 33–43, Nov. 2020.
- [186] E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," *Data Min. Knowl. Discov.*, vol. 7, no. 4, pp. 349–371, Oct. 2003, [doi: 10.1023/A:1024988512476](https://doi.org/10.1023/A:1024988512476).
- [187] M. Weber and M. Denk, "Imputation of Cross-Country Time Series: Techniques and Evaluation," Jan. 2010.
- [188] T. Macé, "Guide méthodologique de validation des données de mesures automatiques," 2015.

- [189] S. K. Warfield, K. H. Zou, and W. M. Wells, "Validation of image segmentation by estimating rater bias and variance," *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.*, vol. 366, no. 1874, pp. 2361–2375, Jul. 2008, [doi: 10.1098/rsta.2008.0040](https://doi.org/10.1098/rsta.2008.0040).
- [190] T. A. Lampert, A. Stumpf, and P. Gañçarski, "An Empirical Study into Annotator Agreement, Ground Truth Estimation, and Algorithm Evaluation," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2557–2572, Jun. 2016, [doi: 10.1109/TIP.2016.2544703](https://doi.org/10.1109/TIP.2016.2544703).
- [191] R. Artstein, "Inter-annotator Agreement," N. Ide and J. Pustejovsky, Eds., Dordrecht: Springer Netherlands, 2017, pp. 297–313. [doi: 10.1007/978-94-024-0881-2\\_11](https://doi.org/10.1007/978-94-024-0881-2_11).
- [192] R. Wu and E. J. Keogh, "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress," 2020, [doi: 10.48550/ARXIV.2009.13807](https://doi.org/10.48550/ARXIV.2009.13807).
- [193] S. Raschka and V. Mirjalili, *Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*, 2nd edition. Birmingham Mumbai: Packt Publishing, 2017.
- [194] J. Brownlee, "Hyperparameter Optimization With Random Search and Grid Search," *MachineLearningMastery.com*. Accessed: Nov. 15, 2023. [Online]. Available: <https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/>
- [195] C.-C. M. Yeh, "Towards a Near Universal Time Series Data Mining Tool: Introducing the Matrix Profile," PhD Thesis in Computer Science, University of California Riverside, 2018.
- [196] R. J. Bayardo, Y. Ma, and R. Srikant, "Scaling up all pairs similarity search," in *Proceedings of the 16th international conference on World Wide Web*, in WWW '07. New York, NY, USA: Association for Computing Machinery, May 2007, pp. 131–140. [doi: 10.1145/1242572.1242591](https://doi.org/10.1145/1242572.1242591).
- [197] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover, "Exact Discovery of Time Series Motifs," *Proc. SIAM Int. Conf. Data Min. SIAM Int. Conf. Data Min.*, vol. 2009, pp. 473–484, 2009, [doi: 10.1137/1.9781611972795.41](https://doi.org/10.1137/1.9781611972795.41).
- [198] "The Matrix Profile — stumpy 1.9.2 documentation." Accessed: Nov. 03, 2021. [Online]. Available: [https://stumpy.readthedocs.io/en/latest/Tutorial\\_The\\_Matrix\\_Profile.html](https://stumpy.readthedocs.io/en/latest/Tutorial_The_Matrix_Profile.html)
- [199] C. E. Yoon, O. O'Reilly, K. J. Bergen, and G. C. Beroza, "Earthquake detection through computationally efficient similarity search," *Sci. Adv.*, vol. 1, no. 11, p. e1501057, 2015, [doi: 10.1126/sciadv.1501057](https://doi.org/10.1126/sciadv.1501057).
- [200] Y. Zhu *et al.*, "Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Barcelona, Spain: IEEE, Dec. 2016, pp. 739–748. [doi: 10.1109/ICDM.2016.0085](https://doi.org/10.1109/ICDM.2016.0085).
- [201] V. Chandola, D. Cheboli, and V. Kumar, "Detecting Anomalies in a Time Series Database," Report, Feb. 2009.
- [202] "The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance." Accessed: Nov. 15, 2023. [Online]. Available: <https://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html>
- [203] Y. Zhu, C.-C. M. Yeh, Z. Zimmerman, K. Kamgar, and E. Keogh, "Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds," in *2018 IEEE International Conference on Data Mining (ICDM)*, Nov. 2018, pp. 837–846. [doi: 10.1109/ICDM.2018.00099](https://doi.org/10.1109/ICDM.2018.00099).
- [204] Z. Zimmerman *et al.*, "Matrix Profile XIV: Scaling Time Series Motif Discovery with GPUs to Break a Quintillion Pairwise Comparisons a Day and Beyond," in *Proceedings of the ACM Symposium on Cloud Computing*, in SoCC '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 74–86. [doi: 10.1145/3357223.3362721](https://doi.org/10.1145/3357223.3362721).
- [205] C.-C. M. Yeh, N. Kavantzias, and E. Keogh, "Matrix Profile VI: Meaningful Multidimensional Motif Discovery," in *2017 IEEE International Conference on Data Mining (ICDM)*, Nov. 2017, pp. 565–574. [doi: 10.1109/ICDM.2017.66](https://doi.org/10.1109/ICDM.2017.66).

- [206] A. H. V. Benschoten, A. Ouyang, F. Bischoff, and T. W. Marrs, “MPA: a novel cross-language API for time series analysis,” *J. Open Source Softw.*, vol. 5, no. 49, p. 2179, May 2020, [doi: 10.21105/joss.02179](https://doi.org/10.21105/joss.02179).
- [207] R. Nisbet, J. Elder, and G. Miner, “Model Evaluation and Enhancement,” 2009, pp. 285–312. [doi: 10.1016/B978-0-12-374765-5.00013-9](https://doi.org/10.1016/B978-0-12-374765-5.00013-9).
- [208] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 10, 2015. [doi: 10.48550/arXiv.1512.03385](https://doi.org/10.48550/arXiv.1512.03385).
- [209] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. [doi: 10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [210] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge.” arXiv, Jan. 29, 2015. [doi: 10.48550/arXiv.1409.0575](https://doi.org/10.48550/arXiv.1409.0575).
- [211] “ILSVRC2015 Results.” Accessed: Nov. 15, 2023. [Online]. Available: <https://image-net.org/challenges/LSVRC/2015/results>
- [212] J. Huber, “Batch normalization in 3 levels of understanding,” Medium. Accessed: Nov. 15, 2023. [Online]. Available: <https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338>
- [213] D. E. Rumelhart and J. L. McClelland, “Learning Internal Representations by Error Propagation,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, MIT Press, 1987, pp. 318–362.
- [214] E. Oja, “Simplified neuron model as a principal component analyzer,” *J. Math. Biol.*, vol. 15, no. 3, pp. 267–273, Nov. 1982, [doi: 10.1007/BF00275687](https://doi.org/10.1007/BF00275687).
- [215] P. Baldi, “Autoencoders, Unsupervised Learning, and Deep Architectures,” in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, JMLR Workshop and Conference Proceedings, Jun. 2012, pp. 37–49.
- [216] “Dimensional Reduction using Autoencoders,” OpenGenus IQ: Computing Expertise & Legacy.
- [217] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, [doi: 10.1126/science.1127647](https://doi.org/10.1126/science.1127647).
- [218] A. Buades, B. Coll, and J.-M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Model. Simul. SIAM Interdiscip. J.*, vol. 4, no. 2, pp. 490–530, 2005, [doi: 10.1137/040616024](https://doi.org/10.1137/040616024).
- [219] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised Learning with Deep Generative Models,” *Adv. Neural Inf. Process. Syst.*, vol. 4, p. 9, Jun. 2014.
- [220] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, 1st edition. New York: Basic Books, 2015.
- [221] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [222] D. Zhang, Y. Sun, B. Eriksson, and L. Balzano, “Deep Unsupervised Clustering Using Mixture of Autoencoders,” Dec. 2017.
- [223] Walber, *English: Precision and recall*. 2014. Accessed: Nov. 14, 2023. [Online]. Available: <https://commons.wikimedia.org/wiki/File:Precisionrecall.svg>
- [224] P. Christen, D. J. Hand, and N. Kirielle, “A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives,” *ACM Comput. Surv.*, vol. 56, no. 3, p. 73:1-73:24, Oct. 2023, [doi: 10.1145/3606367](https://doi.org/10.1145/3606367).
- [225] D. Hand and P. Christen, “A note on using the F-measure for evaluating record linkage algorithms,” *Stat. Comput.*, vol. 28, no. 3, pp. 539–547, May 2018, [doi: 10.1007/s11222-017-9746-6](https://doi.org/10.1007/s11222-017-9746-6).
- [226] J. P. Guilford, “The Minimal Phi Coefficient and the Maximal Phi,” *Educ. Psychol. Meas.*, vol. 25, no. 1, pp. 3–8, Mar. 1965, [doi: 10.1177/001316446502500101](https://doi.org/10.1177/001316446502500101).
- [227] D. Chicco and G. Jurman, “The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification,” *BioData Min.*, vol. 16, p. 4, Feb. 2023, [doi: 10.1186/s13040-023-00322-4](https://doi.org/10.1186/s13040-023-00322-4).

- [228] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020, [doi: 10.1186/s12864-019-6413-7](https://doi.org/10.1186/s12864-019-6413-7).
- [229] A. Dubey and S. Tarar, "Evaluation of approximate rank-order clustering using matthews correlation coefficient," *Int. J. Eng. Adv. Technol.*, vol. 8, pp. 106–113, Jan. 2018.
- [230] M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha, "Beyond kappa: A review of interrater agreement measures," *Can. J. Stat.*, vol. 27, no. 1, pp. 3–23, Mar. 1999, [doi: 10.2307/3315487](https://doi.org/10.2307/3315487).
- [231] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, pp. 37–46, 1960, [doi: 10.1177/001316446002000104](https://doi.org/10.1177/001316446002000104).
- [232] B. Sinha, P. Yimprayoon, and M. Tiensuwan, "Cohen's Kappa Statistic: A Critical Appraisal and Some Modifications," *Calcutta Stat. Assoc. Bull.*, vol. 58, Sep. 2006, [doi: 10.1177/0008068320060301](https://doi.org/10.1177/0008068320060301).
- [233] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, Mar. 1977.
- [234] P. Smyth, "Bounds on the mean classification error rate of multiple experts," *Pattern Recognit. Lett.*, vol. 17, no. 12, pp. 1253–1257, Oct. 1996, [doi: 10.1016/0167-8655\(96\)00105-5](https://doi.org/10.1016/0167-8655(96)00105-5).
- [235] S. Lo and V. Basile, "Hierarchical Clustering of Label-based Annotator Representations for Mining Perspectives," in *Main Session*, Kraków, Poland, Sep. 2023.
- [236] D. Kahneman, O. Sibony, and C. R. Sunstein, *Noise: A Flaw in Human Judgment*. New York, NY Boston London: Little, Brown Spark, 2021.
- [237] Y. Zhu *et al.*, "The Swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code," *Data Min. Knowl. Discov.*, vol. 34, no. 4, pp. 949–979, Jul. 2020, [doi: 10.1007/s10618-019-00668-6](https://doi.org/10.1007/s10618-019-00668-6).
- [238] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st edition. Oxford: Oxford University Press, USA, 1996.
- [239] V. Lakshmanan, *Machine Learning Design Patterns: Solutions to Common Challenges in Data Preparation, Model Building, and Mlops*. Sebastopol, CA: O'Reilly Media, 2020.
- [240] N. Branisavljević, D. Prodanović, and D. Pavlović, "Automatic, semi-automatic and manual validation of urban drainage data," *Water Sci. Technol.*, vol. 62, no. 5, pp. 1013–1021, Sep. 2010, [doi: 10.2166/wst.2010.350](https://doi.org/10.2166/wst.2010.350).
- [241] J.-D. Therrien, N. Nicolaï, and P. A. Vanrolleghem, "A critical review of the data pipeline: how wastewater system operation flows from data to intelligence," *Water Sci. Technol.*, vol. 82, no. 12, pp. 2613–2634, Dec. 2020, [doi: 10.2166/wst.2020.393](https://doi.org/10.2166/wst.2020.393).
- [242] F. van V. Leijnen Stefan, "The Neural Network Zoo," The Asimov Institute. Accessed: Nov. 07, 2023. [Online]. Available: <https://www.asimovinstitute.org/neural-network-zoo/>
- [243] E. Herberg, "Lecture Notes: Neural Network Architectures." arXiv, Apr. 18, 2023. [doi: 10.48550/arXiv.2304.05133](https://doi.org/10.48550/arXiv.2304.05133).
- [244] "Generative Adversarial Networks (GANs): A Complete Guide," clickworker.com. Accessed: Nov. 07, 2023. [Online]. Available: <https://www.clickworker.com/ai-glossary/generative-adversarial-networks/>
- [245] L. Bottou, "On-line Learning and Stochastic Approximations," in *On-Line Learning in Neural Networks*, 1st ed., D. Saad, Ed., Cambridge University Press, 1999, pp. 9–42. [doi: 10.1017/CBO9780511569920.003](https://doi.org/10.1017/CBO9780511569920.003).
- [246] Z. X. Tian, J. P. Jiang, L. Guo, and P. Wang, "Anomaly detection of Municipal Wastewater Treatment Plant operation using Support Vector Machine," in *International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)*, Mar. 2012, pp. 518–521. [doi: 10.1049/cp.2012.1030](https://doi.org/10.1049/cp.2012.1030).
- [247] S. Murray, M. Ghazali, and E. A. McBean, "Real-Time Water Quality Monitoring: Assessment of Multisensor Data Using Bayesian Belief Networks," *J. Water Resour. Plan. Manag.*, vol. 138, no. 1, p. 8, Jan. 2012, [doi: 10.1061/\(ASCE\)WR.1943-5452.0000163](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000163).
- [248] Y. Yuan and K. Jia, "A water quality assessment method based on sparse autoencoder," Sep. 2015, pp. 1–4. [doi: 10.1109/ICSPCC.2015.7338853](https://doi.org/10.1109/ICSPCC.2015.7338853).

- [249] Z. Y. Wu, M. El-Maghraby, and S. Pathak, "Applications of Deep Learning for Smart Water Networks," *Procedia Eng.*, vol. 119, pp. 479–485, Jan. 2015, [doi: 10.1016/j.proeng.2015.08.870](https://doi.org/10.1016/j.proeng.2015.08.870).
- [250] A. H. Ba-Alawi, P. Vilela, J. Loy-Benitez, S. Heo, and C. Yoo, "Intelligent sensor validation for sustainable influent quality monitoring in wastewater treatment plants using stacked denoising autoencoders," *J. Water Process Eng.*, vol. 43, p. 102206, Oct. 2021, [doi: 10.1016/j.jwpe.2021.102206](https://doi.org/10.1016/j.jwpe.2021.102206).
- [251] *Chapter 3 Time series decomposition | Forecasting: Principles and Practice (3rd ed)*. Accessed: Nov. 11, 2023. [Online]. Available: <https://otexts.com/fpp3/decomposition.html>
- [252] J. Perktold *et al.*, "statsmodels/statsmodels: Release 0.14.0." Zenodo, May 05, 2023. [doi: 10.5281/ZENODO.593847](https://doi.org/10.5281/ZENODO.593847).
- [253] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *PVLDB*, vol. 1, pp. 1542–1552, Aug. 2008.
- [254] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *ACM SIGMOD Rec.*, vol. 23, no. 2, pp. 419–429, May 1994, [doi: 10.1145/191843.191925](https://doi.org/10.1145/191843.191925).
- [255] A. Mueen, S. Nath, and J. Liu, "Fast approximate correlation for massive time-series data," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, in SIGMOD '10. New York, NY, USA: Association for Computing Machinery, Jun. 2010, pp. 171–182. [doi: 10.1145/1807167.1807188](https://doi.org/10.1145/1807167.1807188).
- [256] Z. Zimmerman, "SCAMP: SCALable Matrix Profile." Nov. 12, 2023. Accessed: Nov. 15, 2023. [Online]. Available: <https://github.com/zpzim/SCAMP>
- [257] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization." arXiv, Dec. 13, 2015. [doi: 10.48550/arXiv.1512.04150](https://doi.org/10.48550/arXiv.1512.04150).
- [258] "Class Activation Maps – Johannes S. Fischer." Accessed: Jan. 17, 2024. [Online]. Available: <https://johfischer.com/2022/01/27/class-activation-maps/>
- [259] "Difference between PCA VS t-SNE," GeeksforGeeks. Accessed: Jan. 17, 2024. [Online]. Available: <https://www.geeksforgeeks.org/difference-between-pca-vs-t-sne/>

# Appendices

Appendix A. Activation functions .....	373
Appendix B. Neural Networks Architectures .....	378
Appendix C. Stochastic Gradient Descent.....	381
Appendix D. Survey on anomaly detection using AI in the hydrological field. ....	382
Appendix E. Commonly used loss functions .....	386
Appendix F. Time series decomposition .....	388
Appendix G. Matrix Profile algorithms .....	389
Appendix H. Class Activation Maps .....	395
Appendix I. Visualizing Data using t-SNE.....	398
Appendix J. Pairwise F1 score results between annotators .....	400
Appendix K. ANOVA test.....	402
Appendix L. ResNet results using the multivariable approach.....	403



# Appendix A. Activation functions

The activation function plays an important role in the network's decision-making process. Essentially, it determines whether a neuron should be activated or not based on the inputs it receives and the respective weights associated with those inputs. The choice of activation determines the network's capability to capture intricate patterns in data and to make informed predictions. Below is a description of the main activation functions with their respective use cases.

- **Binary step function (Heaviside)**

The binary step function relies on a specific threshold value to determine whether or not a neuron should be activated. This activation function compares the input it receives to the established threshold. If the input exceeds this threshold, the neuron becomes active; otherwise, it remains inactive, effectively preventing its output from advancing to the subsequent hidden layer. The binary step function was initially employed in early versions of the perceptron but was swiftly abandoned due to its inherent limitations such as:

- *Lack of Multi-Valued Outputs:* The binary step function is inherently limited in that it can only produce binary outputs, making it unsuitable for tasks requiring multi-class classification. It cannot distinguish between multiple output classes or provide a probability distribution over several possible outcomes.
- *Zero Gradient:* Another significant drawback of the binary step function is that its gradient is constantly zero. This property presents a considerable challenge during the backpropagation process, as it hinders the efficient adjustment of connection weights through gradient-based optimization algorithms. Without gradient information, it becomes impossible to fine-tune the network's parameters effectively.

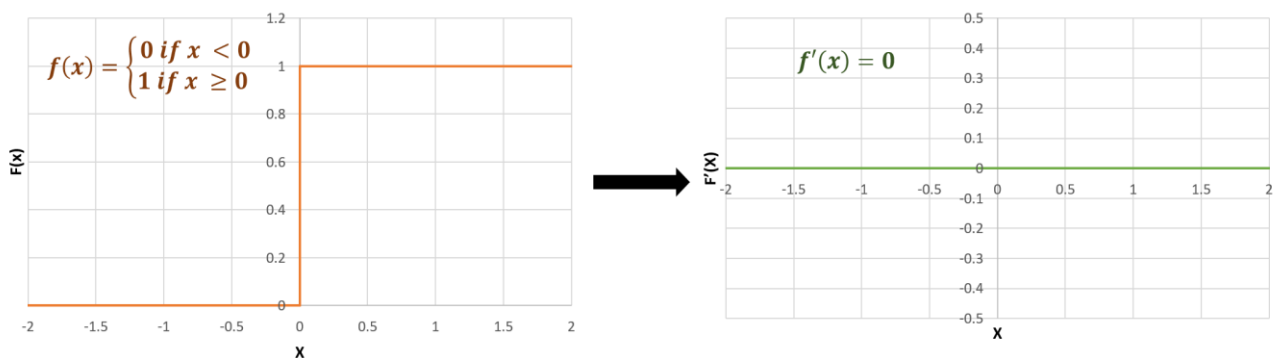


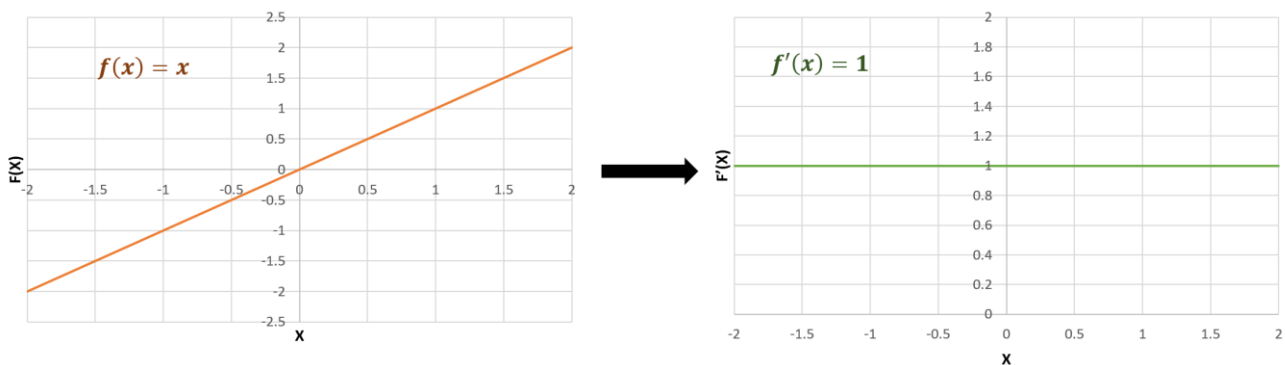
Figure A-1: Binary step function



- **Linear / Identity activation function**

The linear activation function, also referred to as the "no activation", maintains a direct proportionality between the activation and the input it receives. In essence, this function does not introduce any alterations to the weighted sum of the input; it merely outputs the same value it is provided with. Nevertheless, the linear activation function exhibits two significant limitations: Firstly, it renders the application of backpropagation unfeasible, as the derivative of the function remains constant and is entirely independent of the input variable,  $x$ .

Secondly, the usage of a linear activation function leads to the collapse of all network layers into a singular entity. Regardless of the number of layers integrated into the neural network, the final layer effectively transforms into a linear function of the initial layer. Consequently, a neural network employing a linear activation function becomes, in essence, a single-layer network.



**Figure A-2: Identity activation function**

Non-linear activation functions address the shortcomings of linear activation functions in the following ways:

1. They enable the utilization of backpropagation, as the derivative function now exhibits a dependence on the input. This enables the network to retroactively assess the contribution of individual input neuron weights towards improving predictions.
2. They facilitate the incorporation of multiple layers of neurons, as the output now represents a non-linear composition of inputs processed across multiple layers. This means that any output can be expressed as a functional computation within the neural network.

- **Sigmoid function**

The sigmoid function accepts real values as input and produces output values within the range of 0 to 1. The larger the input, the closer the output approaches 1, while the smaller the input, the closer the output tends to 0, as illustrated below. It is commonly applied in models where the prediction of probabilities is essential, aligning with its output range. The function's

differentiability and smooth gradient prevent sudden jumps in output values, thanks to its characteristic S-shaped curve. However, the sigmoid function has its drawbacks. Notably, its gradient is most significant within the range of -3 to 3, with diminishing gradients beyond this interval, leading to the Vanishing Gradient Problem in deep networks when values are too high or too low. Additionally, the sigmoid function lacks a centered output, potentially impacting the efficiency of weight updates. Furthermore, its reliance on exponential operations can slow down computations. Despite these limitations, the sigmoid function remains a valuable tool in various neural network architectures.

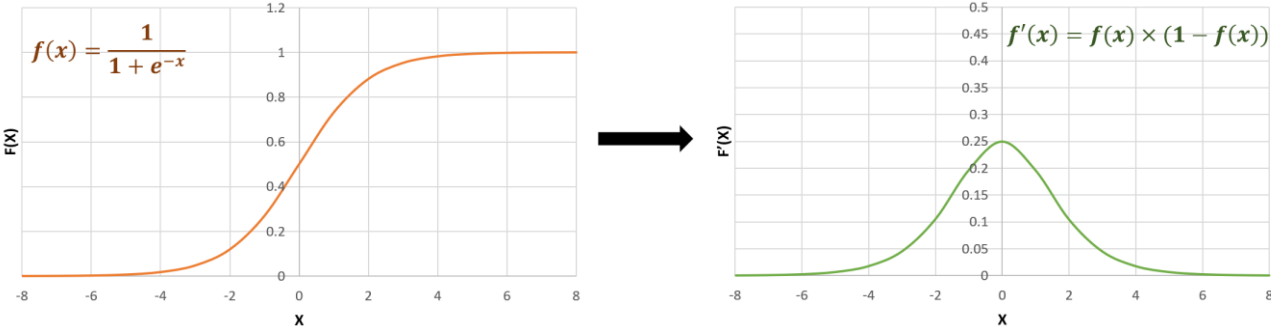


Figure A-3: Sigmoid function

- **Hyperbolic tangent (Tanh) activation function**

The hyperbolic tangent function (tanh) closely resembles the sigmoid activation function, characterized by its distinctive S-shaped curve, but with an output range spanning from -1 to 1. In the tanh function, as the input becomes larger, the output tends toward 1, while smaller inputs result in an output closer to -1. This activation function offers several advantages, including zero-centered output values, allowing for clear distinctions between strongly negative, neutral, and strongly positive outputs. It is commonly employed in the hidden layers of neural networks since its output range from -1 to 1 ensures that the mean of the hidden layer approximates 0, simplifying data centering and facilitating learning in subsequent layers. However, like the sigmoid function, the tanh function faces the challenge of vanishing gradients, although its gradient is notably steeper than that of the sigmoid function.

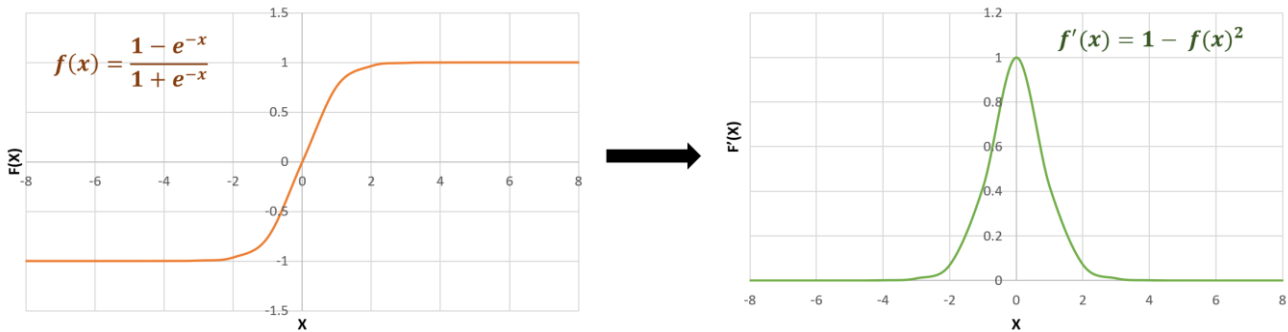


Figure A-4: Tanh activation function

- **ReLU (Rectified Linear Unit) activation function**

The Rectified Linear Unit (ReLU) function has gained widespread popularity in deep learning, surpassing other activation functions. Although it gives an impression of a linear function, ReLU has a derivative function and allows for backpropagation while simultaneously making it computationally efficient. In contrast to the sigmoid and tanh functions, ReLU boasts several advantages. Notably, when the input is positive, it avoids the gradient saturation problem. Furthermore, it excels in computational efficiency, as it maintains a linear relationship. This swift computation is attributed to the absence of exponent calculations required by sigmoid and tanh functions, which can slow down processing. Nevertheless, ReLU has its drawbacks, including the "Dead ReLU" problem, where it becomes entirely inactive for negative inputs, rendering gradients zero during backpropagation, akin to issues faced by sigmoid and tanh functions. Additionally, ReLU yields outputs of either 0 or positive values, signifying that it is not centered at zero, unlike zero-centric functions.

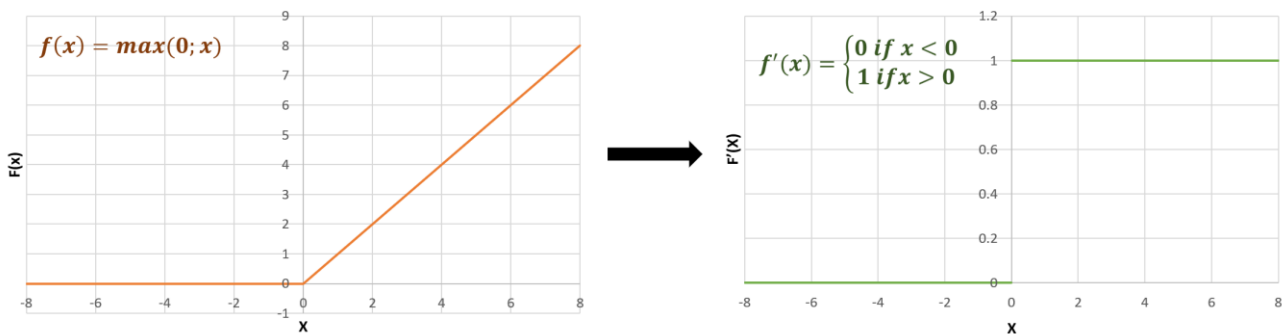


Figure A-5: ReLU activation function

- **Leaky ReLU activation function**

Leaky ReLU represents an enhanced iteration of the ReLU function, primarily aimed at mitigating the "Dying ReLU" issue by introducing a slight positive slope in the negative input range. Leaky ReLU inherits the advantages of ReLU, such as computational efficiency and avoidance of gradient saturation, with the added benefit of supporting backpropagation for negative input values. This modification results in a non-zero gradient on the left side of the

function's graph, eliminating the problem of dormant neurons in that region. Nevertheless, Leaky ReLU is not without its limitations, notably that predictions for negative input values may lack consistency. Additionally, the small gradient associated with negative values can prolong the process of learning model parameters.

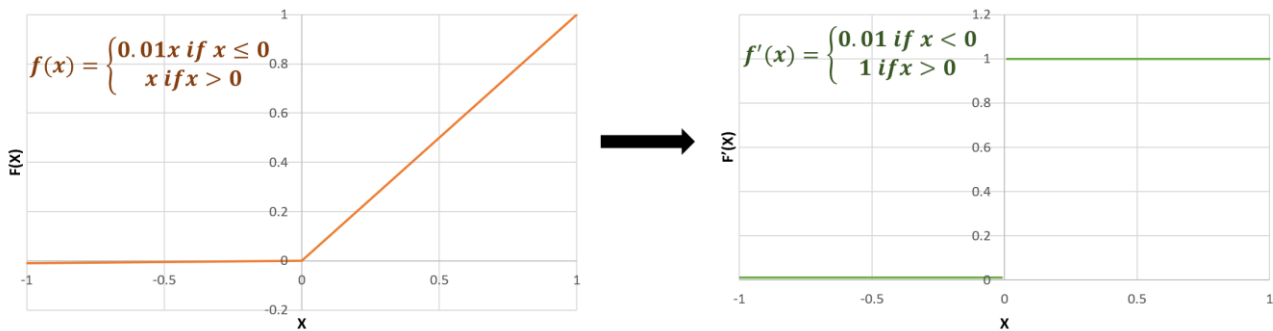


Figure A-6: Leaky ReLU activation function

- Other activation functions<sup>15</sup>

Name	Function $f(x)$	Derivative $f'(x)$	Plot	Characteristics
Parametric ReLU	$f(x) = \max(ax, x)$	$f'(x) = \begin{cases} a & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$		The parametric ReLU function is used when the leaky ReLU function still fails at solving the problem of dead neurons
Exponential Linear Unit	$f(x) = \begin{cases} a \times (e^x - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	$f'(x) = \begin{cases} ae^x & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$		ELU avoids dead ReLU problem by introducing log curve for negative values of input. It helps the network nudge weights and biases in the right direction.
Swish	$f(x) = \frac{x}{1 + e^{-x}}$	$f'(x) = \frac{1 + e^{-x} + xe^{-x}}{(1 + e^{-x})^2}$		Swish activation function can only be implemented when your neural network is $\geq 40$ layers.
Softmax	$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$	$\frac{\partial f(x_i)}{\partial x_j} = f(x_i)(\delta_{ij} - f(x_j))$		Softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.
Maxout	$f(x_i) = \max(x_i)$	$\frac{\partial f(x_i)}{\partial x_j} = \begin{cases} 1 & \text{if } j = \text{argmax } x_i \\ 0 & \text{if } j \neq \text{argmax } x_i \end{cases}$		A Maxout layer is simply a layer where the activation function is the max of the inputs.

<sup>15</sup> This table is not exhaustive.

## ***Appendix B. Neural Networks Architectures***

Deep Learning is a field of constant research. Every day, many new architectures of Neural Networks are proposed and updated. The Figure B-2 illustrates the scope of these potential architectures. For further exploration of these architectures, we invite interested readers to consult [242], which provides a detailed explanation of each model. However, in this study, we will limit ourselves to presenting the main classes of neural networks with brief descriptions [243].

- **Feedforward Neural Network (FFNN)**

FFNNs follow a straightforward data flow, transmitting information from the input to the output layer. Neural networks are commonly structured with layers, which can be categorized as input, hidden, or output layers, working in parallel. A single layer does not contain internal connections, and typically, two consecutive layers are fully interconnected, with each neuron from one layer linked to every neuron in the adjacent layer. The training of FFNNs usually employs the back-propagation method, where the network is provided with paired datasets of "input" and "desired output.". The error being propagated backward is typically a measure of the discrepancy between the input and the output, such as Mean Squared Error (MSE) or the linear difference. Theoretically, with a sufficient number of hidden neurons, the network can model the relationship between input and output.

- **Recurrent Neural Network (RNN)**

RNNs introduce a temporal dimension to Feed Forward Neural Networks (FFNNs), endowing them with a sense of continuity. Unlike FFNNs, RNNs are not stateless; they establish connections across time, forming inter-pass connections. Neurons in RNNs receive input not only from the preceding layer but also from their own state during the previous time step. This characteristic emphasizes the significance of the input sequence order. RNNs face a substantial challenge known as the vanishing (or exploding) gradient problem, wherein the information rapidly diminishes over time, much like deep FFNNs losing information in their layers. Despite the intuition that this might not be a critical concern because it involves weights rather than neuron states, it's essential to realize that the weights across time serve as the storage for past information. If the weight values become too small or exceedingly large, the previous states become less informative.

- **Convolutional Neural Network (CNN)**

CNNs represent a distinct category of neural networks with unique characteristics. They commonly accept input data in the form of a matrix or a three-dimensional tensor, preserving its spatial structure throughout the network's layers. CNNs operate by extracting information from small local regions, like squares or cubes (kernels), within the input images, and subsequently learn features from these patches. The input data is then passed through convolutional layers, which differ from conventional fully connected layers. In convolutional layers, not all nodes are connected to every other node. Each node primarily focuses on nearby cells, typically within a limited proximity. As CNNs progress in depth, the size of these convolutional layers tends to decrease, often by divisible factors of the input, such as going from 20 to 10 and then to 5. Additionally, CNNs frequently incorporate pooling layers, a technique for simplifying and abstracting details. One common pooling method is max pooling, which selects, for example, the pixel with the highest intensity in a 2 x 2-pixel region.

- **Generative Neural Network (GAN)**

GANs are used to generate data based on the patterns discovered from the input data. They consist of two neural networks, the generator and the discriminator, which compete. The generator creates synthetic data, while the discriminator tries to distinguish them from actual data. As the drive progresses, the generator improves to increasingly deceive the discriminator, while the latter refines its ability to discriminate between synthetic and real data. This creates a dynamic balance where the generator generates increasingly realistic data. GANs have many applications, including image generation, text to image translation, and data synthesis. They are widely used in the field of artificial intelligence to create artificial data useful for various tasks.

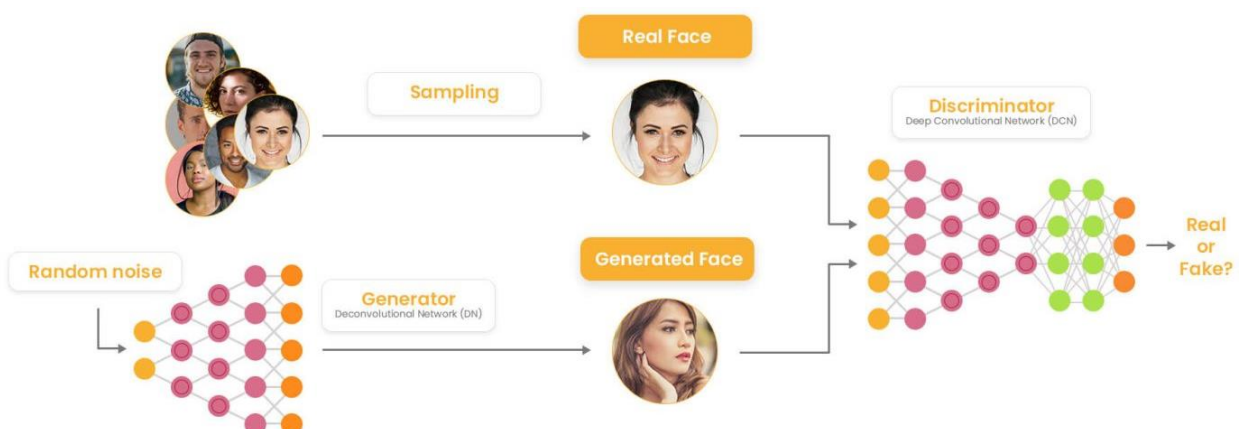


Figure B-1: Simple architecture of GAN - © [244]

# A mostly complete chart of Neural Networks

©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org

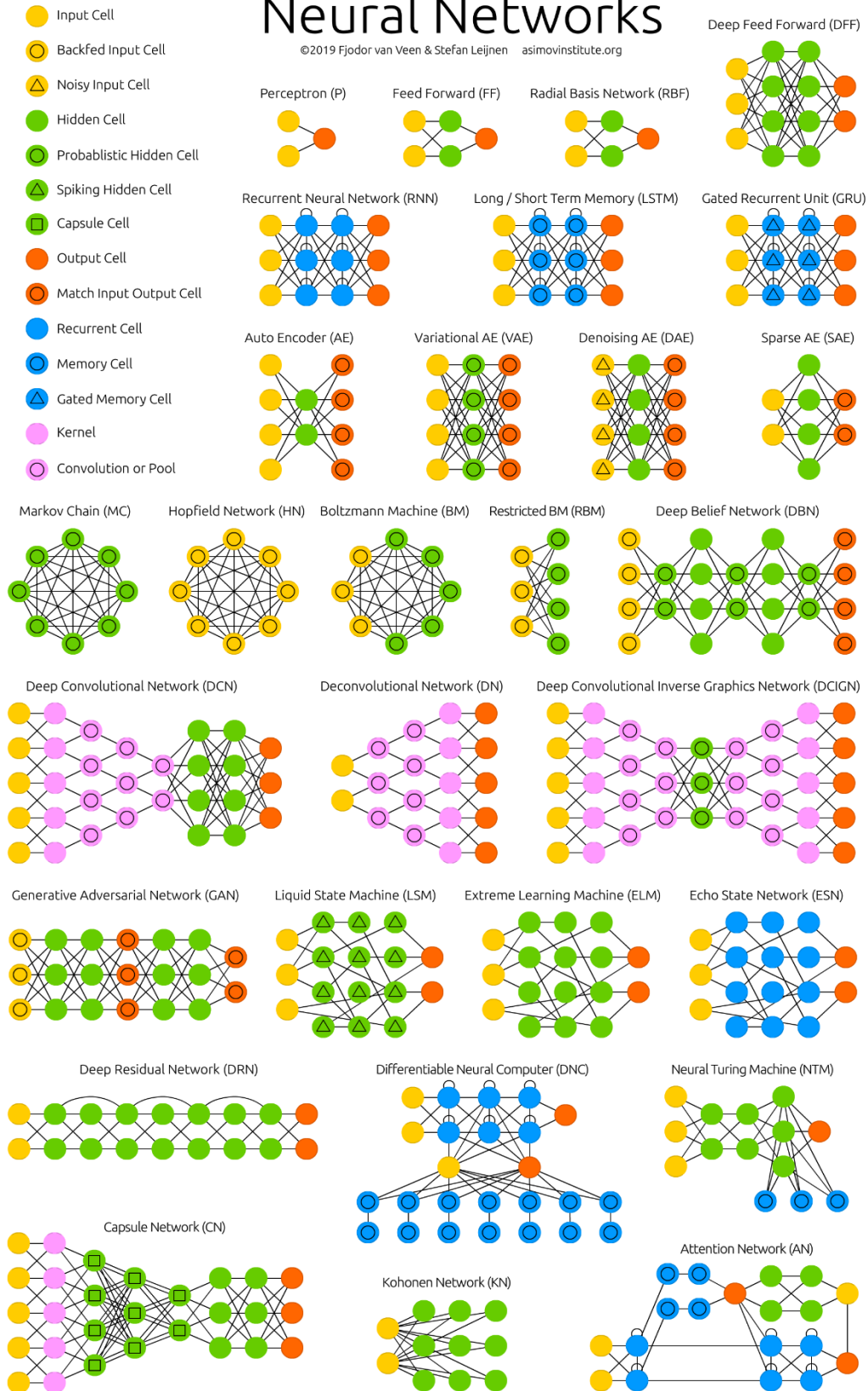


Figure B-2: Infographic with different neural networks architecture (2016) - ©

[242]

# Appendix C. Stochastic Gradient Descent

Gradient descent is a common optimization algorithm used in various ML algorithms as an iterative process that searches for an objective function's optimum value [245].

- **Steps of the gradient descent algorithm**

1. Start with two random points (set of weights / bias)
2. Find the slope / the gradient of the objective / cost function  $gradient = \frac{\delta loss}{\delta \theta}$
3. Calculate the step size and update the parameters as  $\theta \leftarrow \theta - \eta \cdot gradient$
4. Repeat 2 and 3 until the gradient is almost 0

The  $\eta$  parameter is the "learning rate," which is a versatile parameter that impacts the algorithm's convergence. Higher learning rates cause the algorithm to take significant strides along the slope, potentially overshooting the minimum point and failing to converge accurately. Therefore, it is generally advisable to opt for a lower learning rate.

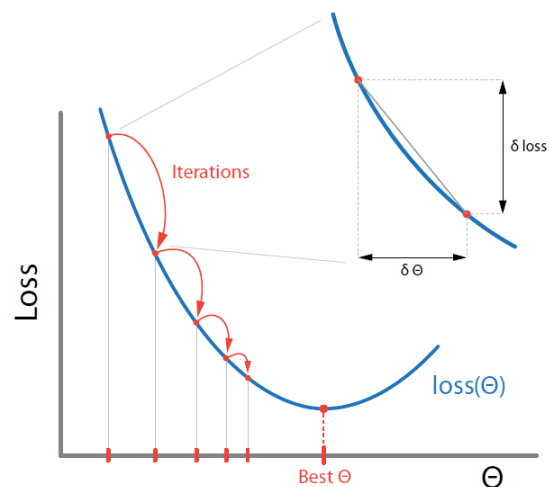


Figure C-1: Gradient Descent

- **The stochastic gradient descent**

Stochastic Gradient Descent (SGD) is a variant of the Gradient Descent algorithm that addresses the computational inefficiency of the latter when dealing with large datasets. In SGD, instead of using the entire dataset for each iteration, only a single random training example (or a small batch) is selected to calculate the gradient and update the model parameters. This random selection introduces randomness into the optimization process, hence the term "stochastic" in stochastic Gradient Descent.

1. Choose a batch size  $1 \leq n < \text{number of the training data samples}$
2. Select an initial vector of parameters and a learning rate
3. Repeat until an approximate minimum is reached:
  - a. Randomly shuffle samples in the training set
  - b. Find the slope using  $n$  samples and update the parameters  $\theta$



## *Appendix D. Survey on anomaly detection using AI in the hydrological field.*

In green, traditional Machine Learning approaches / In bleu, Deep Learning approaches

<b>Date</b>	<b>Article</b>	<b>Objective</b>	<b>Database</b>	<b>Parameters</b>	<b>Models</b>
<b>2012</b>	[246]	Anomaly detection for on-line monitoring	Municipal wastewater treatment plant in Northwest China	flow, effluent quality parameters (COD, NH <sub>3</sub> -N, pH), dissolved oxygen	Support Vector Machine (SVM), Radial Basis Function (RBF)
	[247]	Simulation of E.coli contamination in water distribution system	Water Security Initiative pilot study in Cincinnati	pH, conductivity, turbidity	Bayesian Belief Network (BBN)
	[135]	Anomaly detection in water distribution system	CANARY database	Chlorine, electrical conductivity, pH, temperature, total organic carbon, and turbidity	ANN and Bayes rule for finding the probability of an anomalous event
<b>2014</b>	[136]	Detection of chlorine decay in drinking water systems	Water supply data from distribution system in Czech Republic	Chlorine, flow, pH, temperature, turbidity	Artificial Neural Network (ANN)
<b>2015</b>	[248]	Water quality assessment	Data acquired from DanJiangKou reservoir	COD, DO, Pt, DBO <sub>5</sub> , Ph, turbidity, temperature, Ecoli, oil, chlorophyll, N-NH <sub>3</sub> , N-NO <sub>3</sub>	Sparse autoencoder + SoftMax Classifier

Date	Article	Objective	Database	Parameters	Models
	[249]	Anomaly detection for smart water monitoring	Water monitoring testbed dataset	Tank level, pump status, flow, aggregate and differential demand	Deep Belief Network (DBN)
2017	[21]	Detection of cyber-attacks on drinking water treatment systems	Secure Water Treatment (SWaT) testbed at Singapore university	network traffic, sensor data over 11 days of continuous operation	LSTM-DNN and OCSVM
	[168]	attack detection in water distribution systems	BATtle of the Attack Detection ALgorithms (BATADAL) dataset	tank water level, pump flowrate, and pumping station discharge pressure	CNN-VAE
2018	[129]	Anomaly detection in water consumption data	Experience in collaboration with the Colruyt Group	water consumption data	kNN, LOF, CBLOF, OCSVM
	[119]	Remote monitoring of water quality assessment via a mobile app	A recording system of water quality	10 parameters including ph, turbidity & temperature	SVM, kNN, single layer neural network and deep neural network
	[160]	Anomaly detection approaches for drinking water quality	Dataset of the GECCO Challenge 2018	pH, Redox potential, conductivity, turbidity, chlorine dioxide	Manual feature engineering Vs LSTM neural network
	[131]	Anomaly detection approaches for drinking water quality	Dataset of the GECCO Challenge 2018	pH, Redox potential, conductivity, turbidity, chlorine dioxide	LR, LDA, SVM, ANN
	[140]	Anomaly detection for drinking water quality	Not tested		CNN + deep BiLSTM

Date	Article	Objective	Database	Parameters	Models
2019	[145]	Anomaly detection in water supply data	Water supply data provided by the Ministry of Water Resources of China	water quantity data of 738 days in 204 different sources	OC-SVM
	[116]	In-situ wastewater systems monitoring data	Wastewater data in the municipality of Fehrltorf (Urban Water Observatory)	only dry weather data	CNN- AE
	[155]	Attack detection in water distribution systems	BATtle of the Attack Detection ALgorithms (BATADAL) dataset	tank water level, pump flowrate, and pumping station discharge pressure	OCSVM, LOF, ensemble models
	[130]	Hydrological time series anomaly pattern detection	Measured data of the Chuhe River Basin	water level of Jinniuhu Reservoir and Chuhe River	K-Means, kNN, Isolation Forest
	[149]	Anomaly detection in water quality monitoring	Database from the Dundas wastewater treatment facility, Hamilton, Canada.	Ammonia, potassium, chloride, temperature	LOF, IForest + Statistical approaches
2020	[124]	Intrusion and pollution detection into drinking water distribution systems	Database from the water treatment station "Ghadir El Golla" of Tunis	38 physicochemical and microbiological water quality parameters	Decision Trees + SVM
	[164]	Cyber security attacks on water treatment plans	SWaT, an operational laboratory water treatment plant	Multiple parameters	Autoencoders

Date	Article	Objective	Database	Parameters	Models
	[117]	Leaks detection in water distribution networks	Dataset from an experimental testbed at the University of Waterloo	Hydrophone data	CNN-VAE
2021	[250]	Automatic fault detection	Real WWTP data in South Korea	COD, TP, TN, and SS, pH, Temperature, Turbidity	stacked denoising autoencoder (SDAE)
	[165]	Data validation and anomaly detection	IT-based infrastructure company located in Seoul, South Korea	Water level from 3 sensor sites	Deep autoencoder
	[154]	real-time water quality monitoring	Raw water quality data from the NYEWASCO water treatment plant	Ph, Turbidity	LOF, Random Forest
2023	[118]	Validation and reconciliation of sensor data	Sensor data from Amsterdam WWTP, issued from the aerobic tank	Concentration of nitrate ( $\text{NO}_3^-$ ) and ammonium ( $\text{NH}_4^+$ )	LSTM-AE

# Appendix E. Commonly used loss functions

The loss function serves as a method for assessing the performance of a machine learning algorithm in modeling a given dataset. Its role encompasses several key aspects:

- **Performance Evaluation:** Loss functions provide a quantifiable metric for evaluating the model's performance by measuring the disparity between predicted outcomes and actual results.
- **Iterative Improvement Guidance:** Loss functions direct model refinement by instructing the algorithm to iteratively adjust parameters (weights) to minimize loss and enhance predictive accuracy.
- **Bias-Variance Trade-off:** Effective loss functions aid in striking a balance between model bias (oversimplification) and variance (overfitting), crucial for the model's ability to generalize to unseen data.

Loss functions can be broadly categorized into two major groups based on the types of problems encountered in real-world scenarios: **classification and regression**. In classification problems, the objective is to predict the probabilities of each class involved in the problem. Conversely, in regression tasks, the goal is to predict continuous values based on a given set of independent features provided to the learning algorithm.

## Classification loss functions

- **Binary Cross-Entropy Loss / Log Loss**

This is the most common loss function used in classification problems. The cross-entropy loss decreases as the predicted probability converges to the actual label. It measures the performance of a classification model whose predicted output is a probability value between 0 and 1.

Equation 19: Binary Cross Entropy Loss function

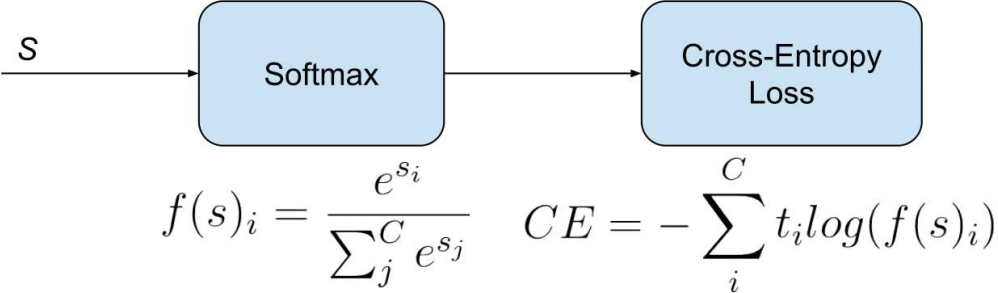
$$L = -\frac{1}{m} \sum_{i=1}^m (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$$

- **Categorical Cross Entropy Loss / SoftMax Loss**

Cross-Entropy Loss is an extension of log loss to multi-class classification problems. Here, the loss is computed over all classes, emphasizing the divergence of the predicted class probabilities from the true class distribution. The complexity of multi-class cross-entropy

escalates with an increase in the number of classes. A fundamental challenge is ensuring that the predicted probabilities across all classes aggregate to one. This normalization is typically achieved using the SoftMax function, which exponentiates each class score and then normalizes these values to yield a valid probability distribution.

**Equation 20: Categorical Cross Entropy Loss Function**



**Loss functions for regression**

- **Mean Square Error (MSE) / L2 Loss**

The Mean Square Error (MSE) quantifies the magnitude of the error between a prediction and an actual output by taking the average of the squared difference between the predictions and the target values. Squaring the difference between the predictions and actual target values results in a higher penalty assigned to more significant deviations from the target value. A mean of the errors normalizes the total errors against the number of samples in a dataset or observation.

**Equation 21: L2 Loss Function**

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$$

- **Mean Absolute Error (MAE) / L1 Loss**

Mean Absolute Error (MAE) is a loss function used in regression tasks that calculates the average absolute differences between predicted values from a machine learning model and the actual target values. Unlike Mean Squared Error (MSE), MAE does not square the differences, treating all errors with equal weight regardless of their magnitude.

**Equation 22: L1 Loss Function**

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

## Appendix F. Time series decomposition

Time series decomposition is a statistical process that breaks down a time series into various components, each representing distinct patterns [251]. A typical decomposition includes:

- **Trend Component** ( $T_t$ ): Reflects the long-term progression of the series, indicating a persistent increasing or decreasing direction. The trend can take non-linear forms.
- **Seasonal Component** ( $S_t$ ): Reflects seasonality, capturing patterns influenced by seasonal factors occurring over fixed and known periods (e.g., quarters, months, or days of the week). There might be multiple seasonal components in the same time series, corresponding to different seasonal periods
- **Irregular Component** ( $R_t$  or "noise"): Describes random and irregular influences, representing the residuals or the remaining part of the time series after accounting for other components.

Thus, a time series can be conceptualized as consisting of three main components: a trend-cycle component, a seasonal component, and an irregular component (which encompasses any other elements in the time series). For an additive decomposition, the time series ( $y_t$ ) is expressed as the sum of its components:

$$y_t = S_t + T_t + R_t$$

Alternatively, a multiplicative decomposition is represented as:

$$y_t = S_t \times T_t \times R_t$$

The choice between additive and multiplicative decomposition depends on the relationship between the magnitude of seasonal fluctuations or variation around the trend-cycle and the level of the time series. Additive decomposition is suitable when these variations do not vary with the level, while multiplicative decomposition is more appropriate when the variations are proportional to the level, a common scenario in economic time series.

Automatic decomposition methods exist, and the statsmodels library offers an implementation of the classical decomposition method through the `seasonal_decompose()` function [252]. One needs to specify whether the model is additive or multiplicative when utilizing this function. Here, we will choose the additive model since multiplicative seasonality is not appropriate for zero and negative values.

# Appendix G. Matrix Profile algorithms

There are a handful of algorithms and different implementations to use the Matrix Profile model. Here, we introduce the matrix profile algorithms differentiating the univariate and multivariate time series. In fact, even if the principle is the same, the algorithms are different.

The main time complexity of the matrix profile approach comes from the calculation of the distance profile. Hence, the challenge is to find an optimized algorithm that allows such computation efficiently. In the next section, we introduce an “ultra-fast” way to compute distance profiles.

## Mueen’s Algorithm for Similarity Search (MASS)

Many algorithms in the literature have been developed to answer the question of similarity search in time series data. Because of the daunting nature of the input, optimization techniques have been adopted to make the calculation efficient and reduce the time complexity. These approaches include indexing structures [253] and early abandoning [254], [186]. However, in the case of complex time series with important levels of noise for example, these approaches degrade to brute force search.

In 2011, [202] introduced “The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance”. The objective is to create the distance profile of a query to a long time series, exploiting the overlap between subsequences using the classic Fast Fourier Transform (FFT) algorithm. The formula to calculate the z-normalized Euclidean distance between two time series subsequences Q (query) and  $T_{i,m}$  a subsequence of T using their dot product  $QT[i]$  is (for demonstration, see [255]):

### Equation 23: Normalized Euclidean Distance

$$D[i] = \sqrt{2m \left( 1 - \frac{QT[i] - m\mu_Q\mu_T[i]}{m\sigma_Q\sigma_T[i]} \right)}$$

Where:

- $m$  is the subsequence length
- $\mu_Q$  and  $\mu_T$  the mean of Q and  $T_{i,m}$  respectively
- $\sigma_Q$  and  $\sigma_T$  the standard deviation of Q and  $T_{i,m}$  respectively
- $QT[i]$  the dot product of Q and  $T_{i,m}$



The full algorithm is outlined in Table 57.

**Table 57: Mueen's Algorithm for similarity Search (MASS) -© [179]**

<b>Procedure MASS(Q,T)</b>	
Input: A query $Q^{16}$ and a time series T	
Output: A distance profile D of the query Q	
1	$n = \text{length}(T)$ , $m = \text{length}(Q)$
2	$T_a \leftarrow \text{Append } T \text{ with } n \text{ zeros}$
3	$Q_r \leftarrow \text{Reverse}(Q)$
4	$Q_{ra} \leftarrow \text{Append } Q_r \text{ with } 2n-m \text{ zeros}$
5	$Q_{raf} \leftarrow \text{FFT}(Q_{ra})$ , $T_{af} \leftarrow \text{FFT}(T_a)$
6	$QT \leftarrow \text{inverseFFT}(\text{ElementwiseMultiplication}(Q_{raf}, T_{af}))$
7	$\mu_Q, \mu_T, \sigma_Q, \sigma_T \leftarrow \text{ComputeMeanStd}(Q,T)$
8	$D \leftarrow \text{CalculateDistanceProfile}(Q,T,QT, \mu_Q, \mu_T, \sigma_Q, \sigma_T)$
9	<b>return</b> D

In line 5, the algorithm calculates Fourier Fast Transform. The resulting  $Q_{raf}$  and the  $T_{af}$  are vectors of complex numbers representing frequency components of the two-time series. In line 6, the algorithm calculates their element-wise multiplication and performs inverse FFT on the product. In line 7, we calculate the mean and standard deviation of each subsequence. This algorithm represents the basis for the calculation of the matrix profile.

### STAMP Algorithm

Once the distance profile is calculated for each subsequence of a time series T, the matrix profile becomes a simple loop that extracts the minimum of each row. The algorithm used for this is Scalable Time series Anytime Matrix Profile (STAMP) and its syntax is presented in Table 58.

**Table 58: The Stamp Algorithm - © [195]**

<b>Procedure STAMP(T,m)</b>	
Input: A time series T and interested subsequence length m	
Output: A matrix profile P and matrix index I	
1	$n \leftarrow \text{Length}(T)$
2	$P \leftarrow \text{infs}$ , $I \leftarrow \text{zeros}$ , $\text{idxes} \leftarrow 0 : n-m$
3	<b>for</b> each idx in idxes:

<sup>16</sup> In our case, the query is a fixed subsequence of T

4	$D \leftarrow \text{MASS}(T[\text{idx}, \text{idx}+m], T)$
5	$P, I \leftarrow \text{ElementWiseMin}(P, I, D, \text{idx})$
6	<b>end for</b>
7	<b>return P, I</b>

In line 5, we perform pairwise minimum for each element in D with the paired element in P (i.e.,  $\min(D[i], P[i])$  for  $i = 0$  to  $\text{length}(D) - 1$ .) We also update  $I[i]$  with  $\text{idx}$  when  $D[i] \leq P[i]$  as we perform the pairwise minimum operation.

The STAMP algorithm is characterized by its anytime nature. In fact, in line 3, the indexes are selected randomly, allowing an approximate solution (if needed). To illustrate the utility of this feature, let's imagine that all anomalies are located in the last subsequences of a time series. Hence, we are obliged to wait for the end of the algorithm to have an appropriate solution. Thus, the anytime feature enables an early stop of the algorithm without compromising the representativeness of the results.

The STAMP Algorithm can also be used in different fashions according to the problem's requirement. Among these different scenarios, we mention [179]:

- Parallelizable STAMP: it allows using multicore machines by distributing the indexes in line 3 on several cores. Once all the secondary cores are done, the main core merge the results using a similar function to ElementWiseMin
- Incremental STAMP (ISTAMP): The aim of this approach is to use Matrix Profile incrementally: as long as new data arrives; it is embedded to the dataset and the matrix profile is adjusted accordingly.

This algorithm is able to process up to a million data points in tenable time. However, for larger datasets, there is a need to upgrade this algorithm. That is the reason [200] developed STOMP.

### **STOMP Algorithm**

By giving up the anytime feature of STAMP algorithm and performing an ordered evaluation of the distance profiles, the Scalable Time series Ordered-search Matrix Profile (STOMP) reduces the time complexity. This speedup factor becomes interesting for large datasets (with millions of datapoints) but makes minor difference for smaller ones.

The main added value of this algorithm is to exploit the link between two consecutive dot products.

**Table 59: The STOMP Algorithm - © [195]**

<b>Procedure STOMP(T,m)</b>	
Input: A time series T and interested subsequence length m	
Output: A matrix profile P and matrix index I	
1	$n \leftarrow \text{Length}(T), l \leftarrow n - m + 1$
2	$\mu, \sigma \leftarrow \text{ComputeMeanStd}(T, m)$
3	$D, QT \leftarrow \text{MASS}(T_{1,m}, T)$
4	$QT\text{-first} \leftarrow QT$
5	$P \leftarrow D, I \leftarrow \text{ones}$ //initialization
6	<b>for</b> $i = 2$ to $l$ //in-order evaluation
7	<b>for</b> $j = l$ downto $2$ //update dot product
8	$QT[j] \leftarrow QT[j-1] - T[j-1] \times T[i-1] + T[j+m-1] \times T[i+m-1]$
9	<b>end for</b>
10	$QT[1] \leftarrow QT\text{-first}[i]$
11	$D \leftarrow \text{CalculateDistanceProfile}(QT, \mu, \sigma, i)$
12	$P, I \leftarrow \text{ElementWiseMin}(P, I, D, i)$
13	<b>end for</b>
14	<b>return</b> $P, I$

The loop in lines 6-13 calculates the distance profile of every subsequence of T in sequential order, with lines 7-9 updating QT. In line 10, we complete the QT with the precomputed QT\_first in line 4.

STOMP is also suitable for parallel computing on several processors. In addition, there is also a GPU-based version of this algorithm which offers a considerable speedup. Table 60 shows a performance comparison of the algorithms presented so far. Note that unlike STAMP where the exclusion zone is  $w/2$ , the default exclusion zone for STOMP is  $w/4$ .

**Table 60: Time required for motif discovery varying the dataset length n (m = 256) – © [200]**

Value of n	$2^{17}$	$2^{18}$	$2^{19}$	$2^{20}$	$2^{21}$
<b>STAMP</b>	15.1 min	1.17 hours	5.4 hours	24.4 hours	4.2 days
<b>STOMP</b>	4.21 min	0.3 hours	1.26 hours	5.22 hours	0.87 days
<b>GPU-STOMP<sup>17</sup></b>	10 sec	18 sec	46 sec	2.5 min	9.25 min

<sup>17</sup> The GPU used here is an NVIDIA Tesla K80

### mSTAMP Algorithm

The multidimensional matrix profile algorithm is an adaptation of the algorithms presented above for univariate time series to include the different dimensions. Table 61 outlines the multivariable STAMP algorithm (mSTAMP). For this study, we used the accelerated version of it, i.e., mSTOMP.

**Table 61: The mSTAMP Algorithm - © [205]**

<b>Procedure mSTAMP (T,m)</b>	
Input: A time series T and interested subsequence length m	
Output: A set of k-dimensional matrix profile P	
1	$P \leftarrow \text{inf matrix of shape } (d \times n-m+1)$
2	idxes integers from 1 to $n-m+1$
3	<b>for</b> each idx in idxes:
4	$D \leftarrow \text{zero matrix of shape } (d \times n-m+1)$
5	<b>for</b> i from 1 to d:
6	$Q \leftarrow T[i, \text{idx}:\text{idx}+m-1]$
7	$D[i, :] \leftarrow \text{distanceProfile}(Q, T[i, :])$
8	<b>end for</b>
9	
10	$D \leftarrow \text{columnWiseAscendingSort}(D)$
11	$D' \leftarrow \text{zero array of length } n-m+1$
12	<b>for</b> l from 1 to d:
13	$D' \leftarrow D' + D[l, :]$
14	$D'' \leftarrow D' / l$
15	$P[l, :] \leftarrow \text{elementWiseMin}(P[l, :], D'')$
16	<b>end for</b>
17	<b>end for</b>
18	<b>return</b> P

In mSTAMP, the query (line 3) is selected in a random order and the distance profile is calculated using MASS in order to have the anytime feature. In the mSTOMP version, the query is selected in order, and the profile distance is calculated using the method proposed by [200]. For the demonstration of the correctness of this approach, we refer the interested reader to [205].

## Conclusion

Table 62 provides a concise description of each of the algorithms presented above.

**Table 62: Concise description of the matrix profile algorithms used in this study.**

Algorithm	Description
<b>Naïve</b>	Inefficient technique for Matrix Profile computation, characterized by a "brute force" approach
<b>STAMP</b>	Among the initial algorithms derived from Keogh's research group, STAMP is recognized as an anytime algorithm.
<b>STOMP</b>	An exact ordered algorithm, STOMP exhibits significantly enhanced speed compared to STAMP.
<b>SCRIMP++</b>	This algorithm combines the anytime component of STAMP with the speed of STOMP

The algorithm that we used in this research is PYSCAMP<sup>18</sup>, which is a python implementation of matrix profile based on SCRIMP++ [203] and with high computational optimizations that are beyond the scope of this work [204]. According to [256], "It is the fastest code in existence for computing the matrix profile". For the multidimensional approach, we used the python implementation of mSTOMP, which is publicly available in [205].

---

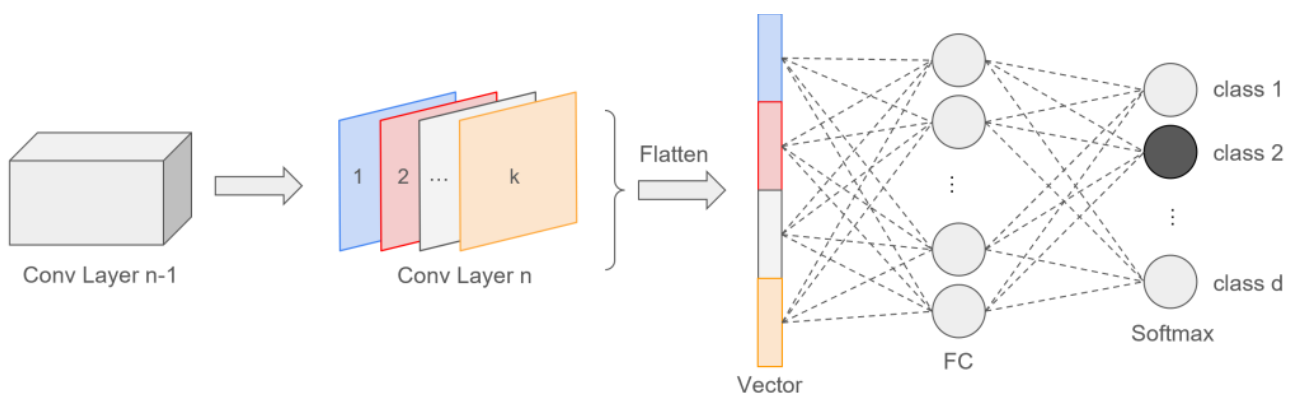
<sup>18</sup> Source code accessible online <https://scamp-docs.readthedocs.io/en/latest/pyscamp/intro.html>

## Appendix H. Class Activation Maps

**Class Activation Maps (CAMs)** are a novel approach in computer vision, developed to improve the understanding of convolutional neural networks (CNNs) by identifying the regions of an image that contribute most to a particular prediction. This method was introduced by [257] in 2016. The main aim of using CAMs is to bring interpretability to neural network models, which are often regarded as black boxes, by providing visual tools for interpreting and validating the decisions made by convolutional networks.

- **Background**

During the training of CNN, filters are learned for convolution operations, generating feature maps at each layer. Typically, filters close to the input layer detect low-level features like edges or lines, while deeper layers combine these low-level features into higher-level concepts. [257] proposed to enhance the interpretability of feature maps by making the number of feature maps in the last convolutional layer equal to the number of classes. This enables each feature map to be interpreted as a confidence map for a specific class, with the strongest activation indicating the region in the original image where the corresponding object is present.



**Figure H-1: Standard approach, where the feature maps are flattened in order to fed them into a dense layer - © [258]**

However, the traditional approach of flattening feature maps for input into fully connected layers diminishes the direct correspondence between feature maps and the output (see Figure H-1). New CNN architectures aim to mitigate this issue by avoiding FC layers. Instead of flattening, they employ global average pooling, preserving the correspondence and localization ability of the network (see Figure H-2).

Global average pooling involves taking the spatial average of each feature map, creating a vector with scalar values representing the mean activation of each feature map (see Figure

5-21). This resulting vector can then be input into a classification (SoftMax) layer, where the activation for a specific class output is a linear combination of average feature map activations multiplied by the corresponding class weights. These weights signify the importance of each feature map for the respective class. This approach maintains interpretability, enhances spatial awareness, and facilitates a clearer understanding of CNN's decision-making process.

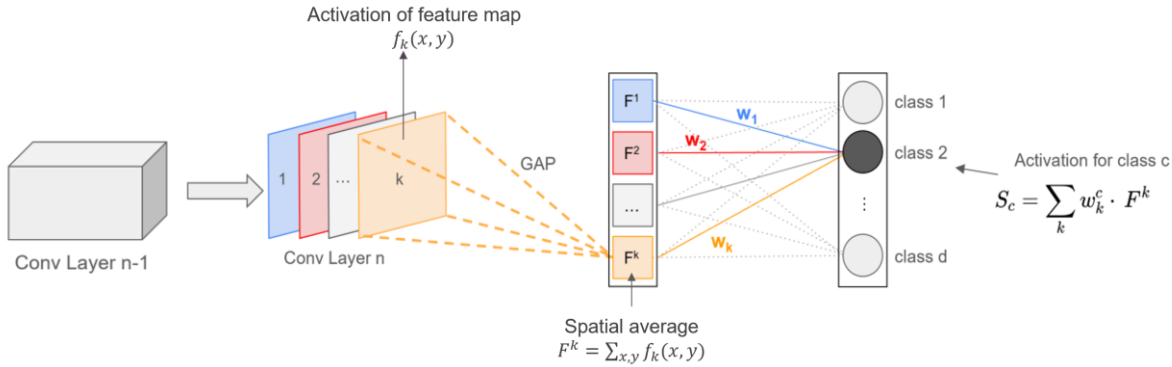


Figure H-2: Global average pooling operation - © modified from [258]

- **Class activation mapping principle**

[257] leverage the concept of global average pooling in neural networks to discern the most discriminative regions within an image. Rather than computing the product of weights and global averages of feature maps, the authors suggest a direct multiplication of feature maps with class-specific weights. This operation results in the creation of a Class Activation Map (CAM), effectively highlighting the image regions crucial for discrimination. The CAM serves as an indicator of where the network focuses when predicting a particular class (see Figure H-3).

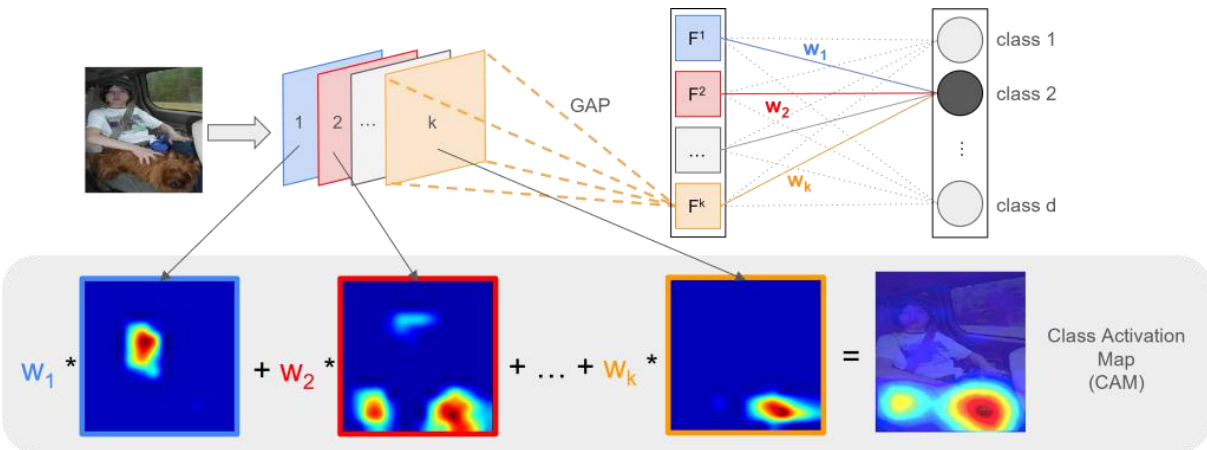


Figure H-3: Linear combination of the weights and feature maps to obtain the class activation map - © [258]

- **CAM for time series**

The adaptation of CAMs to time series represents an extension of this technique to the field of sequential data. Unlike the classical application of CAMs to images (2D or 3D input data), time series present a temporal dimension (1D) that requires a distinct approach. So, instead of working with spatial activation maps, adapting CAMs to time series involves generating 1D activation maps. The underlying logic is based on the idea that each point in time contributes in a different way to class prediction, and temporal CAMs make it possible to visualize the time periods crucial to model decision-making. By applying the same principle of linear weighting of temporal features, temporal CAMs thus offer a visual interpretation of key moments in a temporal sequence, improving the comprehensibility and explicability of deep learning models. Our implementation of CAMs in this study was mainly inspired by [134].



# ***Appendix I. Visualizing Data using t-SNE***

t-distributed Stochastic Neighbor Embedding (t-SNE), developed in 2008 by [221], is a dimension reduction method widely used in high-dimensional data mining. This non-linear technique aims to represent a set of points in a two- or three-dimensional space, while simultaneously preserving the distances between them.

- **Dimension reduction principle**

The reduction of dimensionality is a studied process in mathematics and computer science, involving the transformation of data from a high-dimensional space to a lower-dimensional one while preserving most of the information from the original set. This is essentially an effort to construct fewer variables while retaining maximum information.

There are two main approaches to achieve dimensionality reduction: removing variables and combining variables. Removing variables involves techniques like regularization in certain models, or variable selection based on their relationship with the output through statistical tests like correlation coefficients or mean absolute differences. On the other hand, combining variables is done using methods like Principal Component Analysis (PCA), which transforms correlated variables into new uncorrelated ones.

- **T-SNE principle**

The role of t-SNE is to nonlinearly reduce dimensionality, allowing the separation of data that cannot be linearly separated. This algorithm provides insight into how data is organized in a high-dimensional space, producing distinct and well-defined groups. Once the data structure is understood, t-SNE compresses it by projecting it into a lower-dimensional space (2D or 3D).

The algorithm begins by constructing a probability distribution for pairs of points in the original space, based on their similarities. This distribution represents the relative probability of two points being neighbors of each other. A similar distribution is then created in low-dimensional space. The second step involves projecting the points into the low-dimensional space in such a way as to minimize the Kullback-Leibler divergence between the two probability distributions. Concretely, t-SNE seeks to ensure that pairs of points that are similar in the original space remain close in the reduced space, and that pairs of different points are far apart. The choice of t-distribution reduces sensitivity to outliers and avoids the problem of points being concentrated in certain areas of space.

- **Comparison between t-SNE and PCA**

**Table 63: Comparison between t-SNE and PCA for dimension reduction - © [259]**

	<b>PCA</b>	<b>t-SNE</b>
1.	It is a linear Dimensionality reduction technique.	It is a non-linear Dimensionality reduction technique.
2.	It tries to preserve the global structure of the data.	It tries to preserve the local structure(cluster) of data.
3.	It does not work well as compared to t-SNE.	It is one of the best dimensionality reduction techniques.
4.	It does not involve Hyperparameters.	It involves Hyperparameters such as perplexity, learning rate and number of steps.
5.	It gets highly affected by outliers.	It can handle outliers.
6.	PCA is a deterministic algorithm.	It is a non-deterministic or randomized algorithm.
7.	It works by rotating the vectors for preserving variance.	It works by minimizing the distance between the point in a gaussian.
8.	We can find decide on how much variance to preserve using eigen values.	We cannot preserve variance instead we can preserve distance using hyperparameters.
9.	PCA is computationally less expensive than t-SNE, especially for large datasets.	t-SNE can be computationally expensive, especially for high-dimensional datasets with a large number of data points.
10.	It can be used for visualization of high-dimensional data in a low-dimensional space.	It is specifically designed for visualization and is known to perform better in this regard.
11.	It is suitable for linearly separable datasets.	It is more suitable for non-linearly separable datasets.
12.	It can be used for feature extraction	It is mainly used for visualization and exploratory data analysis.
13.	PCA can be sensitive to the ordering of the data points	t-SNE is less sensitive to the ordering of the data points.

# Appendix J. Pairwise F1 score results between annotators



Figure J-1: Pairwise confusion matrices issued from the validation pool per month

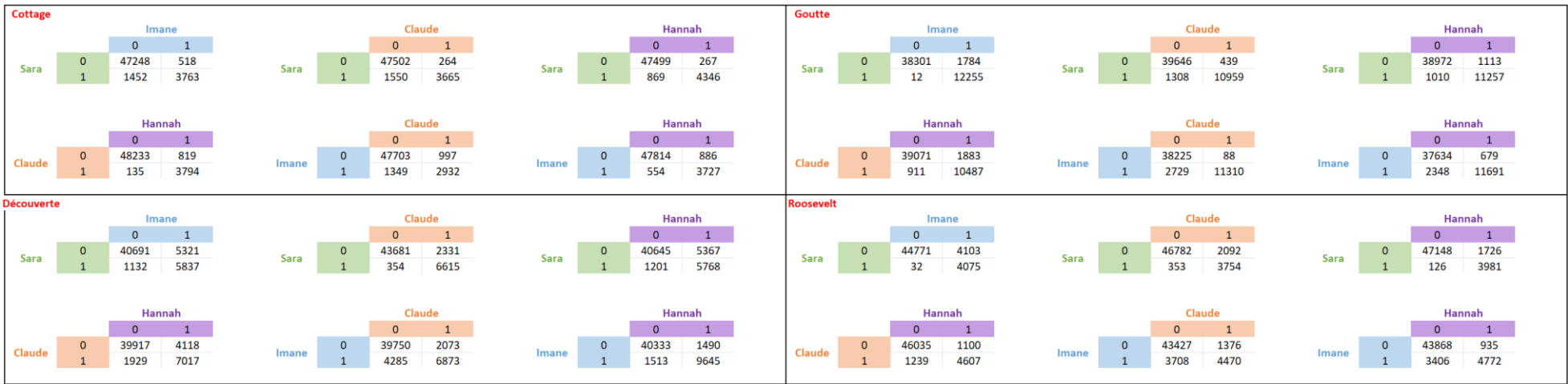


Figure J-2: Pairwise confusion matrices issued from the validation pool per site

## Appendix K. ANOVA test

**Analysis of variance (ANOVA)** is a statistical methodology for assessing significant differences between the means of three or more groups. Its aim is to discern whether the variability observed in the data can be attributed to real differences between the groups, rather than simple random fluctuation. ANOVA operates by comparing between-group variance with within-group variance. More precisely, it subdivides the total variance into two distinct components: the variance attributed to disparities between groups and the variance attributed to random fluctuations within groups.

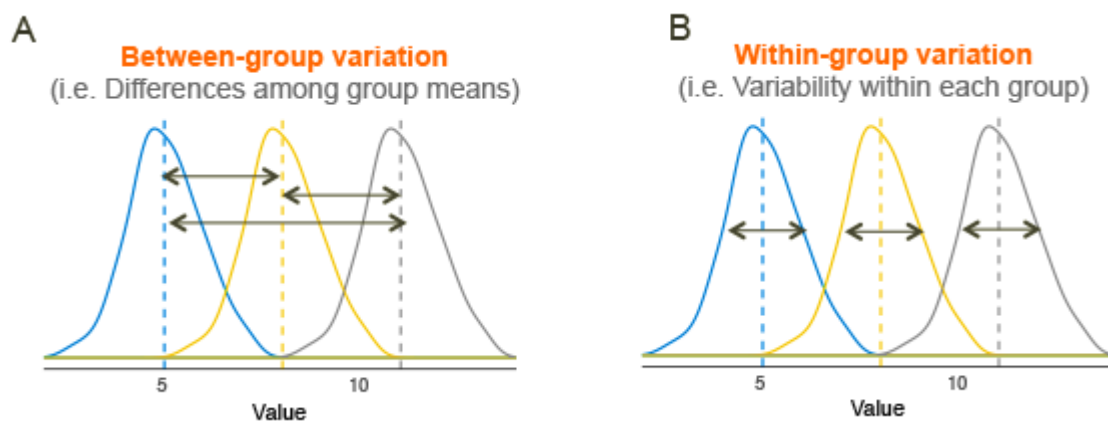


Figure K-1: Types of variation analyzed using the ANOVA test

Calculating the ratio between these two components generates a test statistic. Statistical significance of the latter indicates the existence of significant differences between at least two groups. One-way ANOVA extends the independent t-test to more than two groups or samples.

The null and alternative hypotheses arising from a one-way ANOVA are formulated as follows:

- Null hypothesis  $H_0$ : Mean values are uniform across all groups.
- Alternative hypothesis  $H_1$ : Differences remain between group means.

The ANOVA results indicate whether or not there are differences between at least two groups. However, they do not specifically identify which groups have significant differences.

A number of prerequisites must be met before unifactorial ANOVA can be applied, including: the level of metric scaling of the dependent variable in relation to the nominal scaling of the independent variable, the homogeneity of variances within each group, and the normal distribution of data within groups.

# Appendix L. ResNet results using the multivariable approach

F1 score	Prediction																			
Targets	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
0.05	0.5020	0.5325	0.5333	0.5394	0.5202	0.5094	0.4752	0.4211	0.3978	0.3258	0.2775	0.2471	0.2367	0.1939	0.1500	0.1384	0.1026	0.0654	0.0654	0.0000
0.1	0.4822	0.5629	0.5783	0.5909	0.5743	0.5654	0.5304	0.4734	0.4485	0.3694	0.3158	0.2819	0.2703	0.2222	0.1727	0.1594	0.1185	0.0758	0.0758	0.0000
0.15	0.4691	0.5578	0.5726	0.5849	0.5670	0.5574	0.5202	0.4969	0.4713	0.3893	0.3333	0.2979	0.2857	0.2353	0.1832	0.1692	0.1260	0.0806	0.0806	0.0000
0.2	0.4178	0.5600	0.6036	0.6321	0.6171	0.6098	0.5844	0.5634	0.5362	0.4462	0.3840	0.3443	0.3306	0.2735	0.2143	0.1982	0.1481	0.0952	0.0952	0.0000
0.25	0.4090	0.5630	0.6083	0.6489	0.6353	0.6289	0.6040	0.5839	0.5564	0.4640	0.4000	0.3590	0.3448	0.2857	0.2243	0.2075	0.1553	0.1000	0.1000	0.0000
0.3	0.3853	0.5670	0.6154	0.6592	0.6584	0.6533	0.6286	0.6250	0.5968	0.5000	0.4324	0.3889	0.3738	0.3107	0.2449	0.2268	0.1702	0.1099	0.1099	0.0000
0.35	0.3788	0.5581	0.6146	0.6591	0.6709	0.6667	0.6423	0.6400	0.6116	0.5133	0.4444	0.4000	0.3846	0.3200	0.2526	0.2340	0.1758	0.1136	0.1136	0.0000
0.4	0.3636	0.5512	0.6070	0.6512	0.6623	0.6573	0.6466	0.6612	0.6325	0.5321	0.4615	0.4158	0.4000	0.3333	0.2637	0.2444	0.1839	0.1190	0.1190	0.0000
0.45	0.3404	0.5403	0.6154	0.6747	0.6892	0.6861	0.6772	0.6957	0.6667	0.5631	0.4898	0.4421	0.4255	0.3556	0.2824	0.2619	0.1975	0.1282	0.1282	0.0000
0.5	0.3165	0.5289	0.6138	0.6875	0.7042	0.7023	0.6942	0.7156	0.6857	0.5773	0.5217	0.4719	0.4545	0.3810	0.3038	0.2821	0.2133	0.1389	0.1389	0.0000
0.55	0.2836	0.4872	0.5967	0.6842	0.7164	0.7317	0.7257	0.7525	0.7216	0.6067	0.5476	0.4938	0.4750	0.3947	0.3099	0.3143	0.2388	0.1563	0.1563	0.0000
0.6	0.2716	0.4783	0.5876	0.6757	0.7231	0.7563	0.7523	0.7835	0.7527	0.6353	0.5750	0.5195	0.5000	0.4167	0.3284	0.3333	0.2540	0.1667	0.1667	0.0000
0.65	0.2630	0.4649	0.5714	0.6575	0.7187	0.7521	0.7477	0.8000	0.7692	0.6506	0.5897	0.5333	0.5135	0.4286	0.3385	0.3438	0.2623	0.1724	0.1724	0.0000
0.7	0.2544	0.4513	0.5549	0.6389	0.6984	0.7304	0.7429	0.7957	0.7640	0.6667	0.6053	0.5479	0.5278	0.4412	0.3492	0.3548	0.2712	0.1786	0.1786	0.0000
0.75	0.2456	0.4375	0.5497	0.6338	0.6935	0.7257	0.7379	0.7912	0.7586	0.6582	0.6216	0.5634	0.5429	0.4545	0.3607	0.3667	0.2807	0.1852	0.1852	0.0000
0.8	0.2234	0.4018	0.5181	0.6131	0.6723	0.7037	0.7347	0.7907	0.7805	0.6757	0.6377	0.5758	0.5538	0.4590	0.3929	0.4000	0.3077	0.2041	0.2041	0.0000
0.85	0.2143	0.3871	0.5000	0.5926	0.6496	0.6792	0.7083	0.7619	0.7500	0.6667	0.6269	0.5625	0.5714	0.4746	0.4074	0.4151	0.3200	0.2128	0.2128	0.0000
0.9	0.2051	0.3721	0.4815	0.5714	0.6261	0.6538	0.6809	0.7317	0.7179	0.6571	0.6154	0.5806	0.5902	0.4912	0.4231	0.4314	0.3333	0.2222	0.2222	0.0000
0.95	0.2051	0.3721	0.4815	0.5714	0.6261	0.6538	0.6809	0.7317	0.7179	0.6571	0.6154	0.5806	0.5902	0.4912	0.4231	0.4314	0.3333	0.2222	0.2222	0.0000
1	0.2005	0.3645	0.4720	0.5606	0.6140	0.6408	0.6667	0.7160	0.7013	0.6377	0.5938	0.5574	0.5667	0.4643	0.3922	0.4000	0.3404	0.2273	0.2273	0.0000

Figure L-1: Heatmap using both raw turbidity data (2T) as input

F1 score	Prediction																			
Targets	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
0.05	0.391931	0.566845	0.595318	0.550388	0.534483	0.476636	0.457711	0.382979	0.324022	0.285714	0.224852	0.181818	0.159509	0.160494	0.161491	0.102564	0.065359	0.065359	0.065359	0
0.1	0.362556	0.589235	0.625899	0.599156	0.587678	0.528497	0.511111	0.431138	0.367089	0.324675	0.256757	0.208333	0.183099	0.184397	0.185714	0.118519	0.075758	0.075758	0.075758	0
0.15	0.348872	0.57971	0.614815	0.593886	0.581281	0.52973	0.511628	0.440252	0.386667	0.342466	0.271429	0.220588	0.19403	0.195489	0.19697	0.125984	0.080645	0.080645	0.080645	0
0.2	0.303406	0.527607	0.581673	0.619048	0.630435	0.590361	0.575163	0.5	0.442748	0.393701	0.31405	0.25641	0.226087	0.22807	0.230088	0.148148	0.095238	0.095238	0.095238	0
0.25	0.290172	0.529595	0.585366	0.634146	0.648045	0.608696	0.594595	0.518519	0.460317	0.409836	0.327586	0.267857	0.236364	0.238532	0.240741	0.15534	0.1	0.1	0.1	0
0.3	0.265823	0.525641	0.590717	0.642857	0.658824	0.631579	0.618705	0.555556	0.495726	0.442478	0.35514	0.291262	0.257426	0.26	0.262626	0.170213	0.10989	0.10989	0.10989	0
0.35	0.260731	0.517799	0.589744	0.642487	0.658683	0.644295	0.632353	0.569106	0.508772	0.454545	0.365385	0.3	0.265306	0.268041	0.270833	0.175824	0.113636	0.113636	0.113636	0
0.4	0.2496	0.498361	0.582609	0.634921	0.650307	0.648276	0.636364	0.588235	0.527273	0.471698	0.38	0.3125	0.276596	0.27957	0.282609	0.183908	0.119048	0.119048	0.119048	0
0.45	0.232633	0.468227	0.580357	0.644809	0.675159	0.676259	0.666667	0.619469	0.557692	0.5	0.404255	0.333333	0.295455	0.298851	0.302326	0.197531	0.128205	0.128205	0.128205	0
0.5	0.215334	0.43686	0.559633	0.655367	0.688742	0.691729	0.683333	0.654206	0.591837	0.531915	0.431818	0.357143	0.317073	0.320988	0.325	0.213333	0.138889	0.138889	0.138889	0
0.55	0.191736	0.4	0.52381	0.639053	0.699301	0.704	0.714286	0.686869	0.622222	0.55814	0.45	0.368421	0.324324	0.328767	0.333333	0.238806	0.15625	0.15625	0.15625	0
0.6	0.183028	0.391459	0.514563	0.630303	0.690647	0.694215	0.722222	0.694737	0.627907	0.585366	0.473684	0.388889	0.342857	0.347826	0.352941	0.253968	0.166667	0.166667	0.166667	0
0.65	0.176962	0.379928	0.5	0.613497	0.686131	0.689076	0.716981	0.709677	0.642857	0.6	0.486486	0.4	0.352941	0.358209	0.363636	0.262295	0.172414	0.172414	0.172414	0
0.7	0.170854	0.368231	0.485149	0.596273	0.666667	0.666667	0.692308	0.681319	0.658537	0.615385	0.5	0.411765	0.363636	0.369231	0.375	0.271186	0.178571	0.178571	0.178571	0
0.75	0.164706	0.356364	0.47	0.578616	0.646617	0.66087	0.686275	0.674157	0.65	0.605263	0.485714	0.424242	0.375	0.380952	0.387097	0.280702	0.185185	0.185185	0.185185	0
0.8	0.149153	0.325926	0.430769	0.545455	0.625	0.672727	0.701031	0.690476	0.666667	0.619718	0.492308	0.42623	0.372881	0.37931	0.385965	0.269231	0.204082	0.204082	0.204082	0
0.85	0.142857	0.313433	0.414508	0.526316	0.603175	0.648148	0.673684	0.658537	0.657534	0.608696	0.47619	0.40678	0.350877	0.357143	0.363636	0.28	0.212766	0.212766	0.212766	0
0.9	0.136519	0.300752	0.397906	0.506667	0.580645	0.622642	0.645161	0.625	0.647887	0.597015	0.491803	0.421053	0.363636	0.37037	0.377358	0.291667	0.222222	0.222222	0.222222	0
0.95	0.136519	0.300752	0.397906	0.506667	0.580645	0.622642	0.645161	0.625	0.647887	0.597015	0.491803	0.421053	0.363636	0.37037	0.377358	0.291667	0.222222	0.222222	0.222222	0
1	0.133333	0.29434	0.389474	0.496644	0.569106	0.609524	0.630435	0.607595	0.628571	0.575758	0.466667	0.392857	0.333333	0.339623	0.346154	0.255319	0.227273	0.227273	0.227273	0

Figure L-2: Heatmap using raw turbidity and reconstructed turbidity data (3T) as input

F1 score	Prediction																			
Targets	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
0.05	0.506616	0.638298	0.619926	0.594142	0.590308	0.559633	0.548077	0.50495	0.484848	0.463918	0.406417	0.389189	0.353591	0.335196	0.27907	0.238095	0.184049	0.126582	0.052632	0
0.1	0.484252	0.668831	0.664	0.651376	0.650485	0.619289	0.609626	0.563536	0.542373	0.520231	0.457831	0.439024	0.4	0.379747	0.317881	0.272109	0.211268	0.145985	0.061069	0
0.15	0.468	0.653333	0.652893	0.67619	0.676768	0.645503	0.636872	0.589595	0.568047	0.545455	0.481013	0.461538	0.421053	0.4	0.335664	0.28777	0.223881	0.155039	0.065041	0
0.2	0.407484	0.640569	0.690583	0.722513	0.73743	0.705882	0.7125	0.662338	0.64	0.616438	0.546763	0.525547	0.481203	0.458015	0.387097	0.333333	0.26087	0.181818	0.076923	0
0.25	0.394958	0.623188	0.697248	0.741935	0.758621	0.727273	0.735484	0.684564	0.662069	0.638298	0.567164	0.545455	0.5	0.47619	0.403361	0.347826	0.272727	0.190476	0.080808	0
0.3	0.364026	0.59176	0.698565	0.757062	0.787879	0.75641	0.767123	0.714286	0.705882	0.681818	0.608	0.585366	0.537815	0.512821	0.436364	0.377358	0.29703	0.208333	0.088889	0
0.35	0.353448	0.583333	0.699029	0.758621	0.802469	0.771242	0.783217	0.729927	0.721805	0.697674	0.622951	0.6	0.551724	0.526316	0.448598	0.38835	0.306122	0.215054	0.091954	0
0.4	0.33913	0.576923	0.693069	0.764706	0.810127	0.778523	0.791367	0.75188	0.744186	0.72	0.644068	0.62069	0.571429	0.545455	0.466019	0.40404	0.319149	0.224719	0.096386	0
0.45	0.321586	0.559055	0.673469	0.756098	0.802632	0.769231	0.81203	0.771654	0.780488	0.756303	0.678571	0.654545	0.603774	0.576923	0.494845	0.430108	0.340909	0.240964	0.103896	0
0.5	0.299107	0.524194	0.663158	0.746835	0.794521	0.759124	0.80315	0.760331	0.769231	0.761062	0.716981	0.692308	0.64	0.612245	0.527473	0.45977	0.365854	0.25974	0.112676	0
0.55	0.268182	0.483333	0.626374	0.733333	0.782609	0.75969	0.806723	0.778761	0.788991	0.780952	0.755102	0.729167	0.673913	0.644444	0.554217	0.506329	0.405405	0.289855	0.126984	0
0.6	0.252294	0.466102	0.606742	0.726027	0.776119	0.768	0.817391	0.788991	0.8	0.792079	0.787234	0.76087	0.704545	0.674419	0.582278	0.533333	0.428571	0.307692	0.135593	0
0.65	0.24424	0.452991	0.602273	0.722222	0.772727	0.780488	0.831858	0.803738	0.815534	0.808081	0.804348	0.777778	0.72093	0.690476	0.597403	0.547945	0.441176	0.31746	0.140351	0
0.7	0.236111	0.439655	0.586207	0.704225	0.753846	0.760331	0.810811	0.8	0.831683	0.824742	0.822222	0.795455	0.738095	0.707317	0.613333	0.56338	0.454545	0.327869	0.145455	0
0.75	0.227907	0.426087	0.569767	0.685714	0.734375	0.739496	0.788991	0.776699	0.808081	0.8	0.795455	0.767442	0.707317	0.675	0.575342	0.521739	0.46875	0.338983	0.150943	0
0.8	0.207059	0.391111	0.526946	0.651852	0.699187	0.719298	0.769231	0.77551	0.808511	0.8	0.795181	0.765432	0.701299	0.666667	0.588235	0.53125	0.474576	0.333333	0.166667	0
0.85	0.198582	0.376682	0.509091	0.631579	0.694215	0.714286	0.764706	0.770833	0.804348	0.795455	0.790123	0.759494	0.693333	0.657534	0.575758	0.516129	0.45614	0.346154	0.173913	0
0.9	0.190024	0.361991	0.490798	0.610687	0.672269	0.690909	0.74	0.765957	0.8	0.790698	0.78481	0.753247	0.712329	0.676056	0.59375	0.533333	0.472727	0.36	0.181818	0
0.95	0.190024	0.361991	0.490798	0.610687	0.672269	0.690909	0.74	0.765957	0.8	0.790698	0.78481	0.753247	0.712329	0.676056	0.59375	0.533333	0.472727	0.36	0.181818	0
1	0.185714	0.354545	0.481481	0.6	0.661017	0.678899	0.727273	0.752688	0.786517	0.776471	0.769231	0.736842	0.694444	0.657143	0.571429	0.508475	0.444444	0.367347	0.186047	0

Figure L-3: Heatmap using raw turbidity and conductivity data (2TC) as input



F1 score		Prediction																			
Targets	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1	
0.05	0.448161	0.613982	0.607143	0.591093	0.606061	0.56621	0.540284	0.5	0.469388	0.429319	0.421053	0.369565	0.344444	0.316384	0.277457	0.206061	0.11465	0.090323	0.065359	0	
0.1	0.422877	0.636364	0.648649	0.646018	0.666667	0.626263	0.6	0.557377	0.525714	0.482353	0.473373	0.417178	0.389937	0.358974	0.315789	0.236111	0.132353	0.104478	0.075758	0	
0.15	0.407733	0.626667	0.653386	0.651376	0.673267	0.642105	0.626374	0.582857	0.550898	0.506173	0.496894	0.43871	0.410596	0.378378	0.333333	0.25	0.140625	0.111111	0.080645	0	
0.2	0.36	0.626335	0.681034	0.703518	0.73224	0.701754	0.687117	0.641026	0.621622	0.573427	0.56338	0.5	0.469697	0.434109	0.384	0.290598	0.165138	0.130841	0.095238	0	
0.25	0.348624	0.623188	0.687225	0.711134	0.752809	0.722892	0.708861	0.662252	0.643357	0.594203	0.583942	0.519084	0.488189	0.451613	0.4	0.303571	0.173077	0.137255	0.1	0	
0.3	0.320896	0.59176	0.678899	0.735135	0.781065	0.751592	0.738255	0.690141	0.671642	0.635659	0.625	0.557377	0.525424	0.486957	0.432432	0.330097	0.189474	0.150538	0.10989	0	
0.35	0.311445	0.583333	0.67907	0.736264	0.783133	0.766234	0.753425	0.705036	0.687023	0.650794	0.64	0.571429	0.53913	0.5	0.444444	0.34	0.195652	0.155556	0.113636	0	
0.4	0.298677	0.576923	0.672986	0.741573	0.790123	0.773333	0.774648	0.725926	0.708661	0.672131	0.661157	0.591304	0.558559	0.518519	0.461538	0.354167	0.204545	0.162791	0.119048	0	
0.45	0.279159	0.559055	0.653659	0.732558	0.782051	0.791667	0.794118	0.744186	0.743802	0.706897	0.695652	0.623853	0.590476	0.54902	0.489796	0.377778	0.219512	0.175	0.128205	0	
0.5	0.259188	0.532258	0.633166	0.722892	0.773333	0.782609	0.784615	0.731707	0.765217	0.727273	0.715596	0.640777	0.606061	0.583333	0.521739	0.404762	0.236842	0.189189	0.138889	0	
0.55	0.231827	0.483333	0.586387	0.696203	0.774648	0.784615	0.786885	0.747826	0.785047	0.764706	0.752475	0.673684	0.637363	0.613636	0.547619	0.421053	0.264706	0.212121	0.15625	0	
0.6	0.217822	0.466102	0.566845	0.688312	0.768116	0.777778	0.779661	0.756757	0.796117	0.795918	0.783505	0.703297	0.666667	0.642857	0.575	0.444444	0.28125	0.225806	0.166667	0	
0.65	0.210736	0.452991	0.562162	0.684211	0.764706	0.774194	0.775862	0.770642	0.811881	0.8125	0.8	0.719101	0.682353	0.658537	0.589744	0.457143	0.290323	0.233333	0.172414	0	
0.7	0.203593	0.439655	0.546448	0.666667	0.746269	0.754098	0.754386	0.785047	0.828283	0.829787	0.817204	0.735632	0.698795	0.675	0.605263	0.470588	0.3	0.241379	0.178571	0	
0.75	0.196393	0.426087	0.530387	0.648649	0.727273	0.733333	0.732143	0.761905	0.804124	0.804348	0.791209	0.705882	0.666667	0.641026	0.594595	0.484848	0.310345	0.25	0.185185	0	
0.8	0.178138	0.391111	0.5	0.615385	0.692913	0.713043	0.728972	0.76	0.804348	0.804598	0.790698	0.7	0.657895	0.630137	0.57971	0.491803	0.301887	0.235294	0.204082	0	
0.85	0.170732	0.376682	0.482759	0.595745	0.672	0.707965	0.72381	0.755102	0.8	0.8	0.785714	0.692308	0.648649	0.619718	0.567164	0.474576	0.313725	0.244898	0.212766	0	
0.9	0.163265	0.361991	0.465116	0.57554	0.650407	0.684685	0.699029	0.729167	0.795455	0.795181	0.780488	0.710526	0.666667	0.637681	0.584615	0.491228	0.326531	0.255319	0.222222	0	
0.95	0.163265	0.361991	0.465116	0.57554	0.650407	0.684685	0.699029	0.729167	0.795455	0.795181	0.780488	0.710526	0.666667	0.637681	0.584615	0.491228	0.326531	0.255319	0.222222	0	
1	0.159509	0.354545	0.45614	0.565217	0.639344	0.672727	0.686275	0.715789	0.781609	0.780488	0.765432	0.693333	0.647887	0.617647	0.5625	0.464286	0.333333	0.26087	0.227273	0	

Figure L-4: Heatmap using raw turbidity, reconstructed turbidity and conductivity data (3TC) as input

## Advanced Data Validation Methods for Wastewater Sensors Using Artificial Intelligence

### Résumé

La fiabilité des données dans la gestion des réseaux d'eaux usées est cruciale en raison des implications directes sur les opérations. Cependant, les approches actuelles de validation des données sont souvent coûteuses et manquent d'objectivité. Cette thèse explore les avancées en intelligence artificielle pour instaurer une validation robuste. La mise en place d'un pôle de validation humaine montre que le F1 score moyen entre experts reste à 0.81, soulignant l'inévitable biais humain. Les modèles testés, à savoir Matrix Profile, ResNet et l'autoencodeur, présentent des résultats prometteurs, avec un F1 score de 0.96 pour ce dernier, indiquant une capacité à détecter efficacement les séquences anormales dans les séries temporelles. Matrix Profile excelle en non-supervisé, idéal pour des sites à faible défaillance, tandis que ResNet se montre utile dans des contextes plus problématiques, pouvant justifier une phase de validation manuelle a priori. Ces conclusions ouvrent des perspectives pour une gestion améliorée des réseaux d'eaux usées, basée sur des données fiabilisées grâce à l'IA.

**Mots-clés :** Assainissement, Intelligence artificielle, Capteurs, Séries temporelles, Validation, Anomalies, Matrix Profile, ResNet, autoencodeur

### Résumé en anglais

Data reliability in wastewater system management is crucial because of the direct implications on operations. However, current approaches to data validation are often costly and lack objectivity. This thesis explores advances in artificial intelligence to establish robust validation. The establishment of a human validation pool shows that the average F1 score between experts remains at 0.81, highlighting the inevitable human bias. The models tested, namely Matrix Profile, ResNet and the Auto-encoder, show promising results, with an F1 score of 0.96 for the latter, indicating an ability to effectively detect abnormal sequences in the time series. Matrix Profile excels in non-supervised, ideal for low failure sites, while ResNet is useful in more problematic contexts, which can justify a manual validation phase a priori. These findings open up prospects for improved management of wastewater networks, based on data made more reliable thanks to AI.

**Keywords:** Wastewater networks, Artificial intelligence, Sensors, Time series, Validation, Anomalies, Matrix Profile, ResNet, Autoencoder