# Traçabilité et intégrité de l'information au sein de systèmes critiques : analyse et proposition de méthodes statistiques

Raphaël Larsen

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

# THÈSE DE DOCTORAT DE

l'ÉCOLE NATIONALE SUPÉRIEURE MINES-TÉLÉCOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies de l'Information et de la Communication*
Spécialité : Informatique

Par
**Raphaël M.J.I. LARSEN**

## Traçabilité et intégrité de l'information au sein de systèmes critiques

Analyse et proposition de méthodes statistiques

**Thèse présentée et soutenue à IMT Atlantique, Plouzané, le 7 mars 2022**
**Unité de recherche : LATIM**
**Thèse N° 2022IMTA0291**

### Rapporteurs avant soutenance :

| | |
|---|---|
| Éric Totel | Professeur, Télécom Sud Paris |
| Grégoire Mercier | CTO, eXo maKina |

### Composition du Jury :

| | | |
|---|---|---|
| Président : | Laurent Nana | Professeur, Université de Bretagne Occidentale |
| Examinateurs : | Éric Totel | Professeur, Télécom Sud Paris |
| | Grégoire Mercier | CTO, eXo maKina |
| | Riadh Abdelfattah | Professeur, Ecole Supérieure des Communications de Tunis |
| | Johanne Vincent | Maître de conférence, IMT Atlantique |
| Dir. de thèse : | Gouenou Coatrieux | Professeur, IMT Atlantique |

# Remerciements

Il m'est inconcevable de commencer ces remerciements sans penser à mes proches. Ni ce travail de longue haleine, ni même son auteur ne serait sans eux. Indiscutablement ! Fort de lors soutien, j'ai pu l'achever dans une sérénité éprouvée par les aléas ubuesques des années 2020 et 2021 : merci à mes parents, Laurence et Éric, ma sœur Elin et à mon épouse Diane. Il me faut aussi remercier Solange et Guy dont le toit et l'amour m'ont hébergé des mois durant.

Toujours à l'affût d'idées et de projets, Gouenou Coatrieux fut à l'origine de la problématique de cette thèse. Unique par son intelligence de l'humain et des sciences, Marc-Oliver Pahl m'a aidé à dénouer ce qui aurait pu compromettre l'aboutissement de trois ans de travail.

Réconciliant bienveillance et franchise, Simon Foley m'a lui aussi fait avancer grâce à ses précieux conseils. Bien que n'ayant pas pris part à ce présent travail, deux de mes anciens professeurs de mathématiques le marquent d'une empreinte indélébile, M. Garcin, par sa poésie, et M. Grivaux, par ses mots d'esprit, et bien entendu tous deux par leur pédagogie inoubliable. Aussi, j'aimerais remercier nos collègues d'EDF, Paul Lajoie-Mazenc, Youssef Laarouchi, Franck Bouzon, Maxime Bineau et Sorithy Seng pour leur intérêt dans nos travaux et les fructueux échanges qui nous ont permis de mieux appréhender les préoccupations et les attentes des experts en sécurité industrielle. Nulle thèse n'était aussi peu certaine que la mienne, en effet sans les rencontres de Mme Deutsch et de M. Agbo à l'école primaire, ma scolarité n'aurait pu si bien aboutir ou n'aurait simplement pas abouti. Une autre rencontre déterminante, et plus récente, est celle de Jean-Bernard qui a deviné ma passion pour les mathématiques, ce qui m'a, de fil en aiguille, conduit sur le chemin de la Cybersécurité. Travailler avec Fabien Autrel, l'ingénieur de génie qu'aucune tâche ne peut effrayer, allant du changement d'une pièce du moteur de sa voiture nécessitant son démontage complet à la conception d'un programme informatique vendu des milliers de dollars à des états-uniens en passant par l'assemblage de son imprimante 3D qu'il utilise entre autres pour imprimer les futures pièces de ladite machine, fut une source d'inspiration, en particulier pour mon simulateur de données de système industriel

conçu lorsque Fabien m'aidait à utiliser la maquette reproduisant un système de manufacture miniature. Si je n'avais pas eu les bonnes pratiques de programmation, mes idées n'auraient pas été aussi facilement exploitables, ces bonnes pratiques je les dois à mon ami Jérémy Thomas. Et si Virgile Fritsch n'avait pas été le tuteur de stage bienveillant et à l'écoute qu'il a été, qui sait ce que mon avenir m'aurait réservé. Comme vous l'avez compris, le présent travail ainsi que l'aboutissement des quelques idées qui y ont fleuries n'appartiennent pas qu'à moi seul et sont le résultat de rencontres et de partages dont j'oublie quantité – par exemple, tous mes autres amis que j'ai trop négligé pendant ces trois ans. Gageons que tout ceci n'est qu'un début.

*Je dédie cette thèse à la mémoire d'Ingemann Hagen Boes Larsen, "Farfar", dont la joie de vivre et la bienveillance me guideront toujours.*

# Traçabilité et intégrité de l'information au sein de systèmes critiques

Analyse et proposition de méthodes statistiques

## I Introduction

Notre travail concerne la sécurité des données de systèmes critiques. Nous prenons comme cas d'étude les systèmes de contrôle industriels (ICSs). Un tel système est un ensemble d'appareils qui automatisent les processus industriels par le biais de *software* et de *hardware*. Les ICSs font souvent partie d'une infrastructure critique et sont largement utilisés, par exemple, dans les transports, l'industrie alimentaire, les centrales hydrauliques et électriques. Par conséquent, ils sont la cible d'attaques nécessitant des moyens de sécurité et non plus seulement de sûreté. La sûreté est un état d'un système qui ne peut pas se nuire ou nuire à son environnement dans les limites de sa fonction et de son environnement. La sécurité est l'état d'un système qui ne peut se nuire à lui-même ou à son environnement et auxquels on ne peut nuire, même en cas d'attaque. Le sûreté de fonctionnement, accompagnant les systèmes industriels depuis le début, a été progressivement complétée par des moyens de sécurité de plus en plus sophistiqués dont les systèmes de détection d'intrusion (IDS). Alors que les premiers IDSs étaient uniquement basés sur la signature d'attaques connues, les plus récents utilisent des modèles de détection d'anomalies pour détecter des dysfonctionnements ou des attaques.

## II Problématique

Dans ce travail, nous nous intéressons à la protection des données physiques – c'est-à-dire les données associées à leur processus physique ou autrement dit leur séquence d'actions – en termes de traçabilité et d'intégrité. Ces données sont constituées de valeurs de capteurs, de commandes d'actionneurs et de réponses d'automates et sont très sensibles car elles sont utilisées non seulement pour l'automatisation mais aussi pour la sécurité du système par le biais de la surveillance. L'**intégrité** est définie, dans notre contexte, comme la **quantification de l'altération des données physiques décrivant des actions spécifiques du système**. Nous définissons la **traçabilité** comme l'**authentification de chacun des processus de transformation des données** de leurs créations à leurs dernières utilisations. Les contraintes de tels systèmes telles que la longue vie de leurs composants ou les limites sur la consommation d'énergie nous amènent à considérer des méthodes passives, c'est-à-dire n'ayant besoin que des données.

Après avoir revu les méthodes, potentiellement utiles aux problèmes d'intégrité, issues de domaines de recherches comme le *multimedia forensics* et la détection d'anomalie, nous nous intéressons à des méthodes pouvant servir à notre problème de traçabilité, souvent exposées de façon indépendante dans la littérature et que nous avons regroupées en une nouvelle classe. Cette classe est la caractérisation passive physique de dispositifs, c'est-à-dire le fait de s'appuyer passivement sur les caractéristiques physiques qui résultent de variations propres aux dispositifs en question pour résoudre des problèmes de sécurité. Elle regroupe aujourd'hui six catégories : 1)*Physical Layer Identification* (identification des périphériques pendant la communication en exploitant les caractéristiques uniques de leurs circuits analogiques) 2)*channel fingerprinting* (méthodes permettant de déduire des informations sur l'émetteur et son environnement en utilisant des mesures de canal sans fil spécifiques comme l'angle d'arrivée d'une onde) 3)*device modeling* (modélisation mathématique d'un dispostif pour des questions de sécurité), 4)*physical inconsistencies* (détecter les incohérences par rapport aux lois physiques connues) 5)*environmental signatures* (l'exploitation des signatures de l'environnement) 6)*multimedia forensics specialized techniques* (techniques qui ne peuvent pas encore faire partie d'une catégorie de méthodes suivant un même paradigme mais sont plutôt définies dans un but précis de *multimedia forensics*). Il est aussi utile de mentionner une famille de méthodes de caractérisation actives : les Physically Unclonable Functions (PUFs). Ceux-ci utilisent le hardware de l'appareil à authentifier en vérifiant qu'une réponse de l'appareil à une question correspond bien à une paire de question-réponse d'une base de données pré-établie. Après utilisation, une paire est effacée de la base de données, la rejouer ne peut donc pas constituer une attaque.

## III Nouvelle catégorie de méthodes de caractérisation physique

Pour répondre aux questions de la traçabilité et de l'intégrité, nous nous inspirons des méthodes de multimedia forensics qui détectent la falsification d'image en vérifiant que l'on retrouve le bruit caractéristique des capteurs photographiques dans l'image inspectée.

Dans notre contexte, ce qui caractérise le système est sa variabilité intrinsèque lors de l'exécution des actions pour lesquelles il est programmé. En effet, il répétera un jeux d'actions haut-niveau [1] avec assez peu de variabilité, mais les séquences d'actions sont déterministes. On peut alors appeler presque-déterministe le comportement du système. Pour caractériser la variabilité intrinsèque d'un système industriel et modéliser son fonc-

1. Une action haut-niveau est le résultat du fonctionnement de l'ICS à l'échelle humaine, comme le perçage d'un objet ou sa manipulation par le bras robot, autrement dit ce pour quoi existe le système.

tionnement presque-déterministe, nous pouvons donc considérer ses actions haut-niveau représentées par les données physiques du système. Nous définissons donc le concept d'état d'un système comme l'ensemble des fenêtres temporelles, issues de la série temporelles de données physiques, représentant le même jeux d'actions. Ce concept d'état est différent de celui des modèles espace-état – liant un ensemble de variables d'entrée, de sortie et d'état par des équations différentielles du premier ordre, très utilisé en sûreté de fonctionnement – et va nous servir, à l'aide de méthodes d'apprentissage automatique et statistiques, à caractériser le fonctionnement presque-déterministe d'un système industriel.

Présentons maintenant nos deux solutions utilisant notre concept d'état qui constituent une septième branche de méthodes de caractérisation physique passive.

*a. Intégrité de l'information*

Le contrôle d'intégrité des données physiques est réalisé par l'utilisation d'autoencoders à couches convolutives 1D (une dimension) pour apprendre la représentation normale des états d'un système. Un autoencoder est un réseau de neurones [2] qui reconstruit son entrée tout en faisant face à des contraintes sur ces couches intermédiaires. Originellement utilisé pour la réduction de dimension, ce modèle s'est aussi avéré utile dans le contexte de détection d'anomalie, son erreur de reconstruction servant à construire un score d'anomalie. Dans notre contexte, les autoencoders employés possèdent des couches convolutives 1D pour pouvoir s'appliquer efficacement aux fenêtres temporelles liées aux états.

L'autoencoder apprend à reconstruire les fenêtres temporelles liées à des états du systèmes pour ensuite, pendant la phase d'inférence, fournir une erreur de reconstruction de ces fenêtres servant de scores d'anomalie. Se concentrer sur ces états permet à l'autoencoder de mieux capturer les faibles variations des données physiques et constater lorsqu'elles dévient de la normalité. En outre, la notion d'intégrité que nous proposons peut être pondérée par des connaissances a priori sur la gravité d'une perte d'intégrité des données résultant d'un certain type d'attaque. En effet, un expert qui connaît les besoins et les objectifs spécifiques de son système, voudra donner la priorité à la détection de certains types d'attaques par rapport à d'autres, par exemple si un type d'attaque peut provoquer des désastres mortels alors qu'un autre type d'attaque ne peut provoquer que des dégâts matériels. Nous avons, dans ce but, défini des fonctions de coût permettant un compromis entre la détection d'attaque à long terme et les attaques ponctuelles ainsi

2. C'est-à-dire une succession de couches de neurones artificielles chacune calculant sa sortie à partir des sorties de couches précédentes pour la transmettre aux couches suivantes.

qu'entre la détection d'attaques par rejeu et tout autre type d'attaque.

Nous avons testé notre solution sur les données réelles du banc d'essai Secure Water Treatment (SWaT), faisant progressé l'état de l'art concernant la détection d'anomalies non supervisées pour SWaT. Nous avons analysé notre solution plus en détails grâce aux données générées par un simulateur de système industriel que nous avons conçu pour facilement produire des données provenant d'attaques ou de dysfonctionnements.

*b. Traçabilité de l'information*

En ce qui concerne la traçabilité de l'information, nous nous sommes restreint à l'authentification de deux processus de transformation des données : la création des données et la surveillance des données. Dans le premier cas, nous voulons donc vérifier que les données physiques proviennent bien du système industriel en question. Dans le second cas, nous voulons en revanche vérifier que les données fournies par l'autoencoder de l'IDS, plus précisément, ses matrices d'erreurs de reconstruction, proviennent bien de l'autoencoder. Le deuxième cas consiste donc à davantage sécuriser l'IDS. Le sujet des IDSs engendre beaucoup de recherche visant à améliorer la détection des attaques mais beaucoup moins visant à sécuriser les IDSs eux-mêmes ce qui se révèle être un véritable défi, d'autant plus lorsqu'il s'agit d'IDSs distribués (dIDSs) qui offrent par définition plus de points d'entrées d'attaques. Dans les deux cas, les données sont transformées en une variable sensible à leur changement de distribution, puis le test Wilcoxon-Mann-Whitney (WMW) est effectué sur un échantillon de cette variable pour vérifier qu'elle suit la même distribution qu'un échantillon test, étant lui produit en l'absence d'attaques et de dysfonctionnements.

Pour authentifier les données physiques du système industriel nous utilisons, comme variable pour le test WMW, l'erreur quadratique moyennes de la reconstruction d'une fenêtre temporelle dans un état par l'autoencoder. En effet, cette variable est nécessairement sensible au changement de distribution de données physiques puisqu'elle augmente en présence d'anomalies. Nous avons testé notre méthode grâce aux données de notre simulateur de système industriel, d'abord sur l'attaque la plus subtile considérée lors des tests sur le contrôle d'intégrité, puis sur une attaque encore plus subtile. Pour cela, nous avons tracé les courbes ROC (Receiver Operating Characteristic curves) – qui permettent d'évaluer les performances de modèles de classification à deux classes, dont les modèles de détection d'anomalies – de la fonction de score d'anomalie définie comme l'opposée de la p-valeur du test statistique. En effet, la p-valeur est la probabilité d'observer un échantillon étant donné qu'il suit la distribution de l'hypothèse nulle (c'est-à-dire ab-

sence d'anomalies), donc son opposé est bien un score d'anomalie. L'erreur quadratique moyennes s'avère être une bonne variable pour ce test puisque pour ces deux attaques, les courbes ROC croissent vers le coin en haut à gauche du graphique avec la taille de l'échantillon. Bien sûr, pour l'attaque la plus subtile, la taille de l'échantillon permettant la détection, qui est de l'ordre de la centaine, est bien plus important que la taille de l'échantillon, de l'ordre de la dizaine, pour l'attaque moins subtile.

En ce qui concerne l'authentification des données de l'autoencoder, le problème est plus compliqué. En effet, on ne peut pas se reposer sur l'erreur quadratique moyenne des matrices d'erreurs de l'autoencoder pour vérifier que c'est bien l'autoencoder en question qui produit ces matrices d'erreur. Il faut considérer une autre variable pour caractériser ces matrices d'erreurs de façon plus subtile qu'avec une simple moyenne. L'idée est d'entraîner un réseau de neurones classifieur qui, à partir des matrices d'erreurs de reconstruction va prédire l'état auquel est liée la fenêtre temporelle reconstruite. Ensuite, il faudra calculer une mesure de confiance dans la prédiction de l'état. C'est cette mesure de confiance qui sera utilisé par le test WMW. Pour que cette mesure de confiance soit utile au test WMW, il faut que le classifieur capture les erreurs caractéristiques de l'autoencoder. Pour cela, nous proposons le *multipath Neural Network*, un réseau de neurones classifieur qui fournit une mesure de confiance basé sur la redondance entre la prédiction et le chemin de l'information au sein du réseau. Ceci est effectué grâce aux nouvelles unités computationelles que nous avons définies pour bénéficier de la propriété, montrée par Alain et Bengio dans *Understanding intermediate layers using linear classifier probes*, stipulant que le niveau de séparabilité linéaire augmente le long des couches d'un réseau de neurones classifieur supervisé. Cette fois l'expérience consiste à remplacer les sorties de l'autoencoder par celles de l'autoencoder de l'attaquant, le but est de détecter les échantillons provenant du mauvais autoencoder. L'expérience est effectuée sur l'erreur quadratique moyenne pour éprouver notre assertion que cette variable ne suffit pas à bien caractériser l'autoencoder. Puis nous avons répété cette même expérience sur la mesure de confiance de notre *multipath Neural Network* et finalement, par soucis de comparaison, sur la mesure de confiance d'un réseau de neurones classifieur traditionnel. Les résultats montrent que notre mesure de confiance détectent, grâce au test WMW, ce type d'attaque quelque soit l'autoencoder de l'attaquant et quelque soit l'autoencoder authentique même si certains autoencoders nécessiteront qu'on se serve d'échantillons de plus grandes tailles que d'autres autoencoders, alors que les méthodes utilisant les deux premières variables pourront être dupées par certains autoencoders, quelque soit la taille de l'échantillon.

## IV Conclusion et perspectives

Dans cette thèse, nous définissons l'intégrité et la traçabilité des données de processus physiques d'ICSs. Nous abordons le problème de l'évaluation passive de l'intégrité des données – quantification de l'altération des données liée à des actions spécifiques – pour les ICSs grâce au cadre largement utilisé de détection des anomalies par les erreurs de reconstruction d'autoencodeur. Grâce à ce type réseau de neurones et à notre notion d'état d'un système, nous avons pu améliorer l'état de l'art dans la détection des anomalies dans les données physiques du banc d'essai SWaT et spécifier le score d'intégrité de sorte que la détection de certains types d'anomalies soit davantage mise en évidence. Pour la prise de décision, ce score d'intégrité doit être acompagné d'un seuil d'anomalie qui peut être établi grâce à la loi d'extremum généralisée de la théorie des valeurs extrèmes. Nous avons par ailleurs défini l'algorithme Fast Maxima Sampling utile pour l'estimation de la loi d'extremum généralisée et avons prouvé son efficacité et sa cohérence. Quant à la traçabilité des données physiques d'ICSs – qui consiste à vérifier l'authenticité de chaque processus de transformation des données – nous formulons le problème comme un test d'hypothèse et fournissons un nouveau modèle, le *multipath Neural Network*, qui peut être utilisé lors de la surveillance en ligne ou comme un outil d'analyses forensiques pour vérifier que l'autoencoder d'un IDS n'est pas compromis. Enfin, nous proposons de nouvelles mesures pour évaluer la pertinence des fonctions de score d'anomalies. Pour en revenir à notre classification des méthodes de caractérisation physique, nos travaux, tant sur l'intégrité que sur la traçabilité, ajoutent une nouvelle branche qui est la caractérisation du fonctionnement quasi-déterministe d'un système.

Détaillons maintenant les limites de notre étude. Nous proposons une solution au problème d'intégrité qui permet de prendre en compte, dans une certaine mesure, les connaissances des experts. Plus précisément, l'expert peut donner priorité à la détection des attaques par rejeu par rapport aux autres attaques et des attaques à long terme par rapport aux attaques ponctuelles, ou inversement. Bien que ce soit un premier pas vers une plus grande flexibilité dans la définition de la normalité fixé par l'expert, il existe certainement de nombreux autres types d'attaques sur lesquels il pourrait être intéressant de se concentrer.

Notre travail n'aborde pas la question de la sécurité du système de surveillance lui-même en utilisant des méthodes autres que l'ajout d'une couche de sécurité supplémentaire comme c'est le cas pour l'autoencodeur de surveillance qui est protégé par le *multipath*

*Neural Network*. Ainsi, on pourrait dire que nous avons juste déplacer le problème, même si cela peut être un avantage car certains mécanismes de sécurité pourraient être mis en œuvre dans cette deuxième couche qui n'aurait pas pu être mise en œuvre dans la première puisque la sécurité des IDSs est connue pour être difficile. De plus, cette couche de protection peut être elle-même incomplète : il est surprenant de constater que les *adversarial examples* sont transférables d'un réseau de neurones classifieur à un autre réseau de neurones classifieur du même type, une transférabilité différente, sur un échantillon entier, pourrait rendre cette deuxième couche de sécurité inutile face à toute une classe d'attaques. Dans notre cadre, la transférabilité ne concernerait pas la sortie du réseau de neurones classifieur mais l'absence de changement de la distribution de sa mesure de confiance lorsque l'entrée du classifieur subit un changement de distribution. Ainsi, si ce type de transférabilité est maintenu, un attaquant devrait « juste» construire un *multipath Neural Network* et trouver un ensemble d'entrées pour ce réseau de neurones – soi-disant les matrices d'erreur de l'autoencodeur de surveillance – de telle sorte que sa distribution de mesure de confiance ne change pas, afin de rendre inutile le test WMW sur les valeurs de mesure de confiance du véritable *multipath Neural Network*.

Une autre limite de notre travail est que nous n'avons pas testé notre solution de traçabilité sur des données réelles. En effet, les bancs d'essai ne fournissent pas de données physiques dans une plage supérieure à quelques jours, alors que l'absence de traçabilité peut n'être détectable qu'après quelques semaines, voire quelques mois. Enfin, il existe une condition pour que le test WMW soit cohérent et l'attaquant pourrait essayer de contourner cette condition. Nous avons donc exposé un axe de recherche visant à rendre le test WMW cohérent face aux acteurs malveillants grâce à des propriétés des fonctions de densité de probabilité. Mais cet axe de recherche, tout comme nos nouvelles mesures pour les fonctions de score d'anomalies, a encore besoin de résultats théoriques et empiriques plus solides avant d'être adopté comme solution de sécurité potentielle. Une autre façon de rendre le test cohérent face à un attaquant est d'effectuer le test sur une variable différente obtenue par une transformation secrète des données. Le principe serait alors similaire à celui de PUFs. Au lieu d'une base de données de question-réponse tenue secrète pour identifier activement l'objet physique grâce à une question, on utiliserait une transformation secrète d'un jeux de données pour identifier passivement le processus physique du système grâce à un test WMW sur une variable résumant les données transformées.

**Liste d'articles :**

— *Authenticating IDS Autoencoders Using Multipath Neural Networks*, Raphaël M.J.I. Larsen, Marc-Oliver Pahl, Gouenou Coatrieux, CSNet2021

— *Multipath Neural Networks for Anomaly Detection in Cyber-Physical Systems*, Raphaël M.J.I. Larsen, Marc-Oliver Pahl, Gouenou Coatrieux (soumis à Annals of Telecommunications, Springer)

— *Verifying Data Integrity of Industrial Control Systems Using Deep Learning and a New Concept of State*, Raphaël M.J.I. Larsen, Gouenou Coatrieux (en cours de soumission)

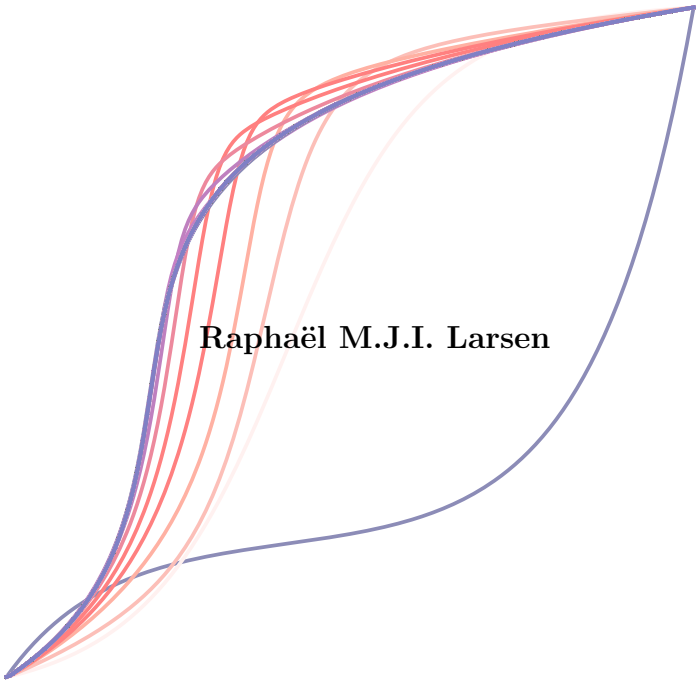
— *Hyper-Neurons and Confidence Through Path Validation*, Raphaël M.J.I. Larsen, click here to access the pdf. [3]

3. ... or copy paste this url : https ://gitlab.imt-atlantique.fr/chaire-cyber-cni-public/host4paper/-/raw/c501b6940030b48c5589f5380930b4f26e77d256/papers/Hyper-NeuronsandCTPV.pdf

# Annexe : version anglaise

# Traceability and integrity of information within critical infrastructures

Study and proposal of statistical methods



**Raphaël M.J.I. Larsen**

A thesis presented for the degree of Doctor of Philosophy
under the supervision of Prof. Gouenou Coatrieux

Institut Mines-Télécom Atlantique

# Table of Contents

**Glossary** **7**

**Introduction** **11**

**Problem statement** **17**

**1 Extreme Value Theory for anomaly detection** **23**

1.1 Introduction to Extreme Value Theory . . . . . 24

1.2 Anomaly detection based on EVT . . . . . 26

1.2.1 Entailment of EVT in cybersecurity . . . . . 26

1.2.2 Methods from EVT for anomaly detection challenges in industrial security . . . . . 27

1.3 Fast Maxima Sampling . . . . . 29

1.3.1 Presentation of the method . . . . . 30

1.3.2 Properties and coherence of Fast Maxima Sampling . . . . . 31

1.4 Conclusion . . . . . 39

**2 Integrity of physical processes from industrial systems** **41**

2.1 Data integrity threats . . . . . 42

2.2 Simulation of industrial systems . . . . . 43

2.3 A novel notion of state in industrial systems . . . . . 53

2.4 Setup of the experiment . . . . . 56

2.5 A baseline integrity score based on states and a 1D-CNN autoencoder . . . 59

2.6 Losses and architectures of autoencoders to refine the integrity specification 70

2.6.1 Rational Sampling . . . . . 70

2.6.2 The Tempered Center Loss on a Fully Connected layer . . . . . 72

2.6.3 The Similarity Transfer Loss and the swelling topology . . . . . 80

2.7 Conclusion . . . . . 86

**3 Traceability of physical processes from industrial systems 87**
3.1 Anomaly detection with the WMW test . . . 88
3.2 Confidence in neural networks predictions . . . 92
3.2.1 State of the art . . . 94
3.2.2 Dissector Layers and clustering . . . 95
3.2.3 Hyper-Neuron and Confidence Through Path Validation . . . 100
3.3 Setup of the experiments . . . 109
3.4 Traceability thanks to the Wilcoxon-Mann-Whitney (WMW) test and the state recovery . . . 111
3.4.1 Is it our Industrial Control System (ICS) ? . . . 111
3.4.2 Is it our Intrusion Detection System (IDS) ? . . . 112
3.5 Conclusion . . . 116

**4 Metrics for anomaly scoring functions 117**
4.1 The disparity . . . 118
4.2 The susceptibility . . . 120
4.3 Conclusion and perspective . . . 122

**Conclusion 123**

**Index of ideas 126**

**Bibliography 129**

**Appendices 145**

**A Details on Experimental data 146**
A.1 Real data from the testbed Secure Water Treatment (SWaT) . . . 146
A.2 Visualization of cycles simulated with Simulation of ICSs with Binary-valued Registers (Sibriz) . . . 147

**B A simple method to approximate the anti-function of a bijection 150**

**C A measure of the level of linear separability 154**

**D Details for reproducibility 156**
D.1 Initialization and regularization of Dissector Layers (Diss-Layers) . . . 156

D.2 Datasets and Neural Networks for the experiment on confidence measures . 159

**E Analysis and future research on the multipah NN** **162**

E.1 Analysis of the multipath Neural Network (NN) confidence measure . . . . 162

E.2 Future Research . . . . . 164

# Glossary

**ADALINE** ADAptive LInear NEuron. 14

**Adam** Adaptive Moment Estimation. 58

**AUC** Area Under the Curve. 58, 59, 67, 68, 77, 79

**CDF** cumulative distribution function. 24–26, 28–31, 35, 37, 90

**CNN** Convolutional Neural Network. 15, 57–61, 64–68, 70–73, 75, 76, 78, 80–85, 115, 126, 161

**CPS** Cyber-Physical System. 11, 13, 41, 54, 69, 123

**CRP** Challenge-Response Pair. 21, 89

**CTPV** Confidence Through Path Validation. 92, 100, 101, 104, 127

**DCS** Distributed Control System. 43

**dIDS** distributed Intrusion Detection System. 112

**Diss-Layer** Dissector Layer. 4, 94, 96–106, 110, 127, 156–160, 163

**DL** Deep Learning. 11, 13–16, 19, 21, 59, 70, 87, 92, 94, 123, 125, 127, 166

**ECE** Expected Calibration Error. 94, 159, 160, 162, 166

**ELU** Exponential linear unit. 58

**EM** Excess-Mass curve. 117, 118, 122

**ENF** Electrical Network Frequency. 20

**EVT** Extreme Value Theory. 14, 16, 23–25, 27–29, 39, 123

**FC** Fully Connected. 15, 58, 60, 62, 64–68, 70, 72, 73, 75–81, 83–86, 110, 126, 160

**FMP** Fractional Max-Pooling. 71

**FPR** False Positive Rate. 23, 26, 27, 59, 67, 68, 79

**GEV** Generalized Extreme Value. 25, 28, 29, 39, 123, 126

**GPD** Generalized Pareto Distribution. 28

**i.i.d.** independent and identically distributed. 24–29, 35

**ICS** Industrial Control System. 11–13, 17–19, 21, 23, 27, 41–46, 52–55, 59, 87, 112, 115, 123, 126

**IDS** Intrusion Detection System. 13, 87, 109, 112, 115, 123, 124

**LSTM** Long Short-Term Memory. 18

**M-SGf** Multivariate Savitzky-Golay filter. 121

**ML** Machine Learning. 14, 15, 18, 126

**MLP** multilayer perceptron. 15

**MSE** Mean Squared Error. 62, 70, 109–111, 113–116

**MSLE** Mean Squared Logarithmic Error. 158

**MV** Mass-Volume curve. 117, 118, 122

**NIPS** Neural Information Processing Systems. 15

**NLL** Negative Log-Likelihood. 94, 105, 160

**NLP** Natural Language Processing. 104

**NN** Neural Network. 5, 14–16, 18, 57–66, 68, 70–73, 75–80, 82, 83, 85–88, 92–97, 100–107, 109–111, 113–116, 123–125, 127, 156, 157, 159, 160, 162–166

**NSS** Non-integer Strided Sampling. 71

**OC-SVM** One-Class Support Vector Machine. 23, 81, 82

**PLC** Programmable Logic Controller. 12, 43

**PLI** Physical Layer Identification. 19

**PR** Precision Recall. 58, 59, 68

**PUF** Physically Unclonable Function. 21, 89

**PWM** Probability Weighted Moment. 28

**ReLU** rectified linear unit. 106

**ROC** Receiver Operating Characteristics. 52, 57–59, 63, 79, 83, 99, 100, 109–111, 113–116, 118

**RTU** Remote Terminal Unit. 43

**SGEV** Skew Generalized Extreme Value. 28

**SHA** Secure Hash Algorithm. 17

**Sibriz** Simulation of ICSs with Binary-valued Registers. 4, 43, 51, 52, 56, 57, 63, 147

**STL** Similarity Transfer Loss. 80, 82–86

**SWaT** Secure Water Treatment. 4, 42, 53, 56–58, 67, 69, 86, 123, 146

**TCL** Tempered Center Loss. 72, 73, 76–78, 82, 85, 86

**TDA** Topological Data Analysis. 108, 116, 124, 127, 163–165

**TDNN** Time Delay Neural Network. 15, 60

**TNR** True Negative Rate. 82

**TPR** True Positive Rate. 23, 58, 79, 82

**WMW** Wilcoxon-Mann-Whitney. 4, 16, 21, 87–89, 91, 109–116, 124, 126, 127

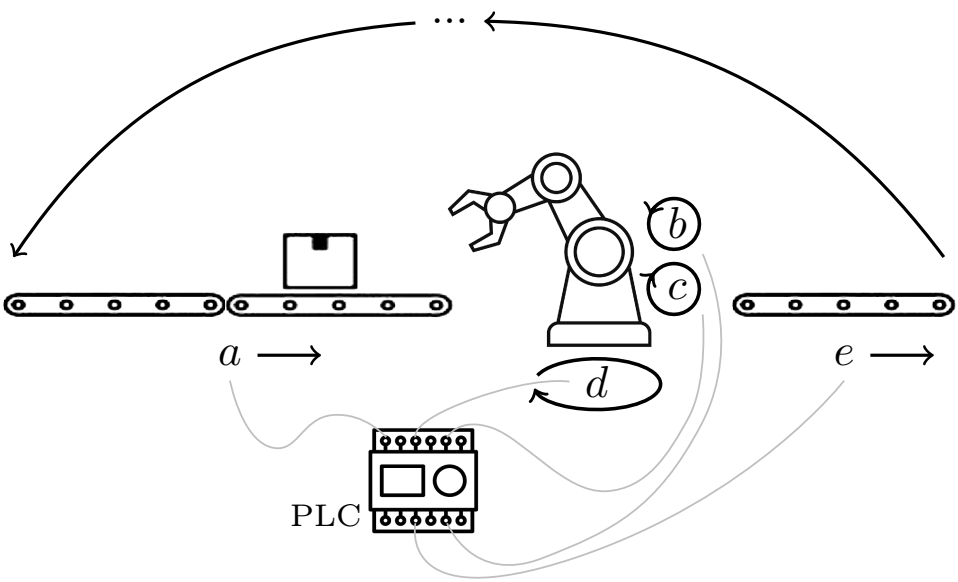
# Introduction

**Industrial Control Systems.** Industrial Control Systems (ICSs) refers to computers within a network of machines and electronic components, i.e. sensors and actuators, used to automate industrial processes through specific hardware and software. They are often part of critical infrastructures and are members of the family of Cyber-Physical System (CPS). CPSs are systems of which the physical components are controlled by software to operate in the real world. In that respect, ICSs are critical infrastructures because the health of people working around needs to be ensured and the system itself must be sustainable. Data within ICSs are not only used for automation but also for safety and security through monitoring. Safety is a state of a system that cannot harm itself or its surroundings within the limits of its function and its environment. Security is the state of a system that cannot harm itself or its surroundings and which one cannot harm, even under attack. ICSs represent one of the main source of wealth of a country. They are widely used in transport industries, food industries, power plants and water plants to name a few. This is the second aspect of the criticality of ICSs, they are vital to our way of life. Hence, it is also a strategic target for hostile actors.

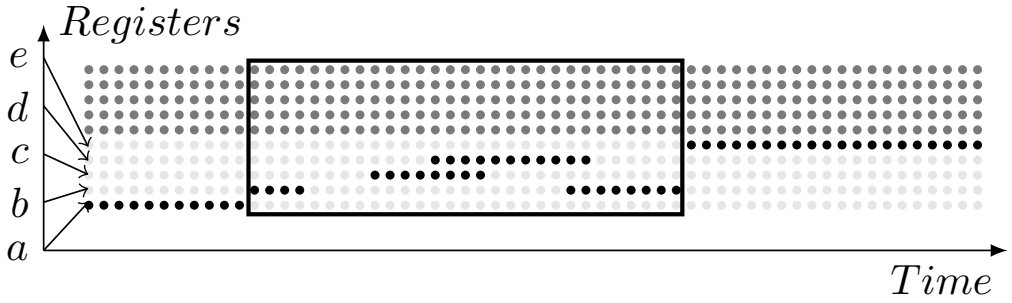
Yet, most of these systems are not secure by design. More generally, research and development in cyber security increased continuously since the invention of the microprocessor used in modern computers. [1]

When one is concerned with the security of such systems, one can monitor data stemming from the communication protocol or one can monitor sensors and actuators data, we call physical data, that are expressed in time series. In this study, we are interested in the second option. We assume Physical data times series are sampled with a regular time step and the same time. In other words, we consider a multidimensional time series with as many dimensions as sensors and actuators whose values are sampled at the same time every $x$ seconds, with $x$ relatively small and constant over time. For instance, when

1. It is interesting to remark that the idea of a microprocessor – that is to have all functions of a central processing unit on a single integrated circuit – comes from the same Marcian Edward Ted Hoff that contributed to the advent of Deep Learning (DL) as explained in the end of this introduction. He, of course, is one of the co-author that engineered the first microprocessor.




Cycle 1




Cycle 2

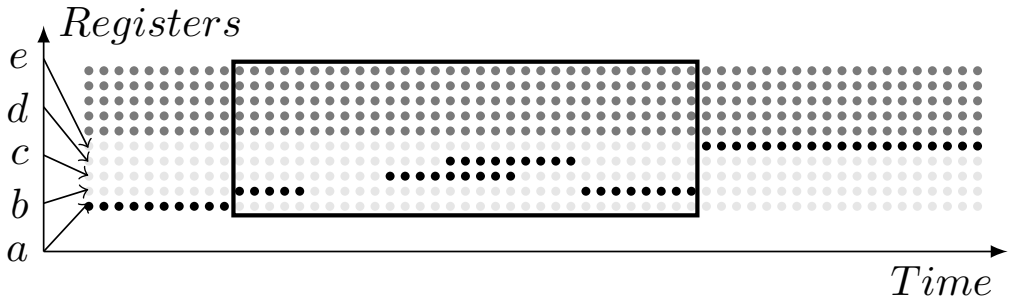



Figure 1 – Illustration of an ICS composed of a Programmable Logic Controller (PLC) with binary registers and safety motor sensors that detect whether a motor works at a certain speed (value 1) or not (value 0) ($a$ and $e$ for the 1st and 2d conveyor belt motors respectively) and actuators for motions of the robot arm ($b$, $c$ and $d$ are for "down", "up" and "turn right" motions respectively). The time series is the registers data loggings at evenly spaced instants (every seconds for instant), black points are for value 1, light gray points for value 0 and dark gray points for values of registers not mentioned in the scheme. Black rectangles delineate the time windows containing the register values representing the action of the robot arm transporting the object from the first conveyor belt to the second.

actuators and sensors are binary one can easily illustrate the time series as in Figure 1.

The CIA triad (Confidentiality, Integrity, Availability) and other security goals emerged in the literature. Safety engineering, which is part of the implementation of industrial systems since the beginning, has been progressively complemented by Intrusion Detection Systems (IDSs) more and more sophisticated. While the firt IDSs were only based on signature of known attacks, more recent ones employ anomaly detection models, that are models usually unsupervisedly trained to detect anomalies such as dysfunctions or attacks. Thus, after presenting a wide literature on the security of physical systems in the next chapter, we decide to look at anomaly detection methods to propose security solutions for ICSs.

**Anomaly Detection.** In anomaly detection for ICS, there exists three research axes on according to [154]: 1) protocols content, 2) network characteristics and 3) values stemming from the industrial system physical process. The first axis only treats network packets (formatted units of data) and tries to model protocol rules in order to detect attack that might deviate from them. [2] The second axis treats sequences of network packets and tries to model the communication relationship between devices. In this dissertation, we are interested in the third axis, that is to modeling the physical relationship between devices as well as the normal range of a system behavior. Therefore, we deal with the data related to the physical process of the system, that is the set of actions of the system that take place in the physical environment of the CPS. We do not look after other data types such as protocols content. The data we are interested in naturally appear as time series in $\mathbb{R}^n$ from sensors and actuators as already mentioned, while in the two other cases, one has to intensively preprocess data in order to be able to use mathematical models.

There exist a lot of anomaly detection techniques, nevertheless Chandola et al. [16] identified six categories of anomaly detection techniques: classification based, nearest neighbor based, clustering based, statistical, information theoretic and spectral techniques. In the context of industrial systems, data can appear in large dimensional spaces as the number of sensors can be of the order of thousand for example [141]. The capacity of DL models to handle high dimensional data and to adapt to different situations, in other words flexibility, are the reasons why we choose to rely on them for developing methods useful for ICS cyber security problems. Autoencoders are frequently used for

2. Authors of [154] are the first to model unknown binary protocols, which is appreciable since most of industrial protocol are still proprietary.

anomaly detection and the industrial system security domain is no exception. Indeed, their ability to deal with high dimensional space and the fact that no labeled data are needed make this model an attractive solution for security of industrial systems. They fall within the first category of anomaly detection techniques, that is classification based anomaly detection techniques, more particularly, one-class anomaly detection technique. We are therefore interested in extending and specifying these tools for industrial systems so that it benefits as many security teams as possible to tackle the issue of integrity of information. We are also interested in the confidence of a neural network prediction so we develop a new kind of computational unit, Hyper-Neuron, crucial to our answer to the issue of traceability of information. Both of these terms, integrity and traceability, will be clarified in the next section. Finally, Extreme Value Theory (EVT) plays an important role in our anomaly detection context so we develop a statistical method useful for this theory, called Fast Maxima Sampling.

**Artificial Neural Networks.** Whereas the subject of the present dissertation is not a priori linked to DL theory, its class of algorithms turned out essential for our work through the controllability of their architectures and loss functions.

Machine Learning (ML) and more precisely artificial Neural Networks (NNs) was pioneered by the psychologist Frank Rosenblatt in 1957 with his perceptron [129], the first algorithm applying the paradigm of *learning-by-examples*, now referred as supervised learning—though the first machine learning program is the Samuel's Checker Program presented a year before on TV. This breakthrough relied on the work of Donald Olding Hebb [65] whose the neuroscientific theory is used as a learning principle for artificial NNs. The learning principle is to strengthen an existing connection between two neurons if they are activated simultaneously. The perceptron is the first artificial neuron supported by a rule to automatize its contruction but the first model of the nerve net in the brain is the one of Warren Sturgis McCulloch and Walter Harry Pitts [107] of which the weights needed to be specified by humans. Marvin Minsky and Seymour Papert proved the perceptron is able to learn to separate two linearly separable clusters in a finite number of connection weight updates [110]. Group method of data handling (GMDH) was introduced by Ivakhnenko and Lapa [72] and involves polynomial NNs. They were trained by regression analysis and a height layers network was presented in a paper of Ivakhnenko in 1971 as explained in [135]. Bernard Widrow and Marcian Edward Ted Hoff proposed an improved version of the perceptron, called ADAptive LInear NEuron (ADALINE), that resists noise in the training

set [152]. To perform more complex tasks, NNs with several non-linear layers are needed but a practical updating rule was lacking until the famous backpropagation algorithm of which the first NN version was proposed by Paul Werbos in 1981 [151] – according to [135] – and the second and prevailing version was proposed and implemented by Yann Lecun in 1987 [92]. The backpropagation algorithm uses the chain rule to update the connection weights between neurons in hidden layers, in contrast with neurons of the ouput layer whose weights updates only need the error from the output against the example target.

In 1982, a content-addressable memory system, now called Hopfield net, revived the interest in NNs. Another breakthrough in the same period, the neocognitron proposed by Kunihiko Fukushima and Sei Miyake [42], inspired Convolutional Neural Networks (CNNs) and Time Delay Neural Networks (TDNNs), a special case of 1D-CNN, as explained in the paper of the *AI conspirators trio* [93]. Their own papers having been refused at the Neural Information Processing Systems (NIPS) conference, they organized the NIPS 2007 Satellite Meeting. It took this unusual event to initiate a trend that gave ML conferences a renewed focus on DL, the part of ML which involves NNs. CNNs achieve some spatial or temporal translation invariance, as the case may be, in the data representation with high computational efficiency. Since then, a multitude of NNs took shape and proved themselves useful with modern computing capabilities and vast quantities of data. However, many challenges remain in DL which lacks unsupervised learning and learning from small training sets methods, as well as interpretability and confidence. More details on the DL's history can be found in a historical survey [135] and in the introduction of the book *Deep Learning* [54]. The standard NN type is the Fully Connected (FC) NN, also known as multilayer perceptron (MLP), which consists of an input layer of which every unit is connected to every unit of the next layer and so on until the output layer. Each connection is assigned a weight and each unit output of a layer, different from the input layer, is obtained with a weighted sum, of the outputs of the previous layer, on which is applied an activation function. At first, the weights are randomly initialized, then they are updated during training thanks to the backpropagation of the output layer error obtained using a loss function. We will also use CNNs already mentionned and autoencoders which are NNs whose the main objective is to reconstruct the input while undergoing some constraints on their hidden layers.

The challenges cited above are herein tackled in varying degrees thanks to a new computational unit we introduce, the Hyper-Neuron.

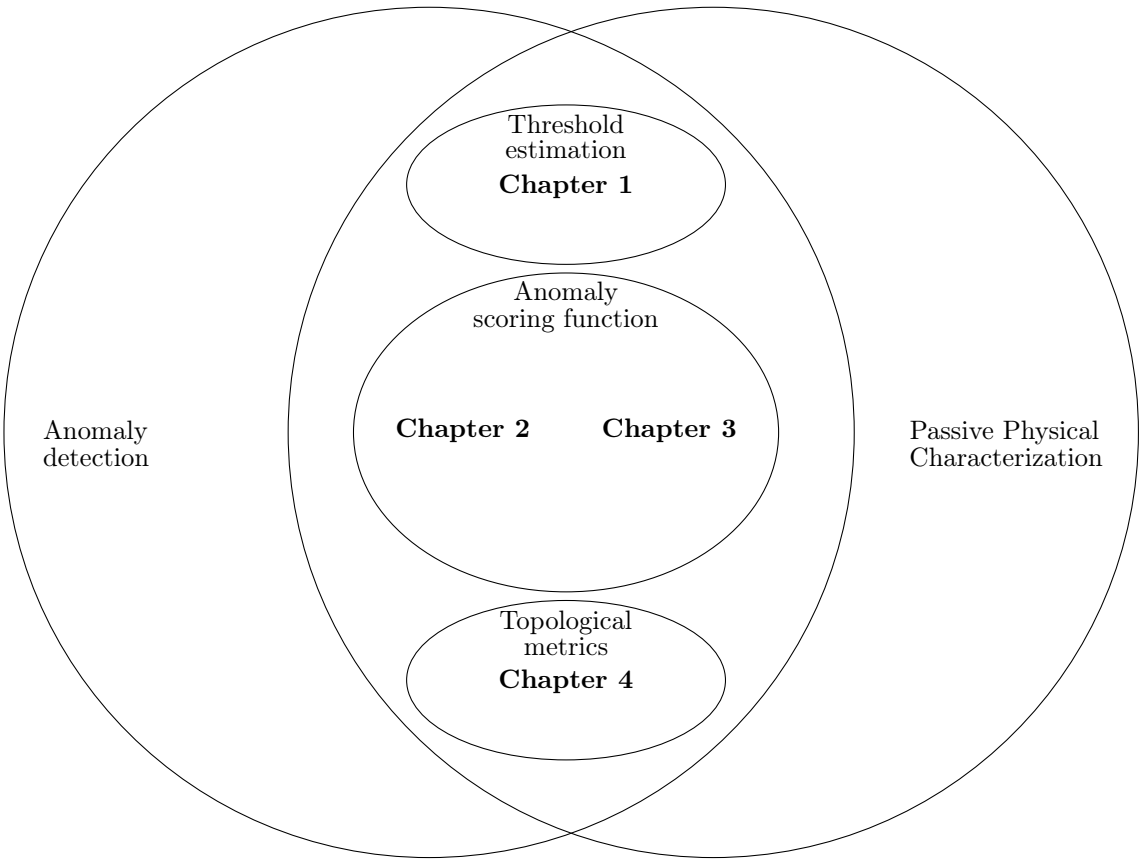



Figure 2 – Thesis outline

**Layout of the thesis.** [3] In the sequel, we state the problem of characterization of physical systems, browse through solutions, expose challenges and specify the subject into the anomaly detection problem. Chapter 1 presents Extreme Value Theory (EVT) and its link to anomaly detection, in particular in industrial systems data, and our contribution to this theory. Chapter 2 is about our solution to integrity of information based on our novel notion of state in industrial systems and also on NNs losses we defined, besides the simulation model of industrial systems we used for the experiments. Chapter 3 is about our solution to traceability of information based on the WMW test and on our DL model called multipath NN that provides a measure of confidence in its prediction. Chapter 4 presents a new topological metric about the performance of anomaly scoring functions.

3. For an optimal reading experience, there are hyperlinks for abbreviations and references but there are also hyperlinks for pages where they appear so that one can go from one page to the glossary or the bibliography and come back instantly. Likewise, the numbering of chapters, sections, figures, tables footnotes, definitions, equations and some list numbers uses hyperlinks.

# Problem statement

As explained in the introduction, only the physical process data of an ICS – that are sensors measurements, automata commands and actuators responses –, or transformations of these data, are taken into consideration in this work. Let us explain why. First of all, data reliability is critical in such systems as it enables ensuring safety and security. Also, it is necessary to guarantee the adequacy of data with the systems for digital forensics analyzes. Secondly, because of the long life cycle of these systems [31] and their complexity, solutions that tackle data reliability should use as little as possible specific functionalities like additional devices and should not interfere too much with the monitored system. As stated in [106], it is even more valid for online monitoring. Solutions that require neither of these conditions are called passive solutions throughout this dissertation [1].

In the context of physical processes, data reliability relies on two important aspects. The first is data integrity which, in the general context, refers to the ability to have confidence in the fact that a piece of data has not been unexpectedly modified. In our context, we refine this definition for ICS physical process data, **integrity is the quantification of the alteration of a piece of data describing specific actions of the system**. Our goal is thus to bind common actions of the system to their normal representation. There are two fields with approaches potentially useful to meet our problem. The first is the digital forensics, that is the process of uncovering and interpreting electronic data. Authors of [7], classify digital forensics into two categories that are computer forensics and multimedia forensics. In the second category, the goal is to make reliable and authentic «digital representations of parts of reality» so that they can serve as probative facts [7], while computer forensics is only interested in the state of a computer after a crime to provide proofs of a crime. For instance, some approaches use the inherent characteristic noise from camera to identify an object or to assess data falsification. As for approaches from this field to assess data integrity, cryptography is a classic one. Cryptographic hash functions, such as Secure Hash Algorithm (SHA)-256, compute a short length data digest:

1. «*It is important that no measuring device or monitoring system interferes with the ICS environment under scrutiny.*» [106]

a hash, but it just states whether pieces of data are the same as their original version. Watermarking constitutes an interesting alternative as it can measure the alteration of a piece of data. It consists in the embedding of a message into the data to protect such as an image, an audio signal or a database, by transforming them via imperceptible modifications based on the principle of controlled distortion. At the detection phase, one compares the recomputed signature with the extracted one to decide about data integrity. But, as cryptography, it needs computational resources at both communication ends for data integrity to be assessed. This is not the case of passive approaches. Such methods were proposed in the context of digital forensics with as purpose to detect that an image has been tampered [69], for instance. They usually rely on the extraction of some image features that are sensitive to image modifications. It is then possible thanks to a classifier to detect the kind of modification an image undergone, this one being local or global. The second field that can help us stipulating data integrity is anomaly detection. Recent strategies are based on ML techniques like the well-known NNs towards anomaly detection, in images [5] for instance. As mentionned in the introduciton, Anomaly detection in ICS operates on three axes [154]: 1) the protocol content, 2) the network characteristics and 3) modeling the values stemming from the industrial system physical process; and only the third axis interests us. According to [154], there are many ways to approach axis 3), now referred to as model-based approaches. Model-based approaches can be inspired from safety engineering, as envelope escalation for sensor readings for instance [101]; or can rely on describing the underlying the fundamental process of attacks as with a graph-theoretic characterization of attacks [124], for instance; or or by directly trying to detect manipulated physical data like clustering methods with a Gaussian mixture model [83], for instance. To go further, NNs based anomaly detection methods have not been proposed only in the domain of intrusion detection but also in safety of industrial systems. For example, In [99], a method aims to optimize the connection between neurons to benefit from the event ordering relationship for ICS safety purposes as predictive maintenance. These proposals aim at going further than state space representation models – that are models consisting of a set of input, output and state variables linked by first-order differential equations – used in operational safety of ICS. Indeed, when these state space representation models are not sufficient for a proper monitoring due to a limitation inherent to the complexity of the system and the achievable sampling rate [32], one needs a more complex model. Both in safety and security, autoencoders have been widely studied for ICS monitoring. For example, in [103] a Long Short-Term Memory (LSTM) autoencoder is applied

on continuous multidimensional time series for anomaly detection. However, they have not specifically been employed to answer our problem of data integrity, that is to bind common actions of the system to their normal representation. Model-based approaches can also rely on physics theories. For example, in [38], characteristics of ICS networks are seeked and discussed so that passive fingerprinting approaches suited to ICS environment can be developed. More precisely, the authors support that the «*mechanical composition of a device can usually be obtained from manual inspection, available drawings/pictures, or manufacturer's specifications*» in order to derive a mathematical model of its behavior, with equations for example like for their Latch Relay modeling. This is part of a broader class we call physical characterisation methods summarized in Figure 3. In this work, we aim at using DL methods drawn from image analysis and anomaly detection to the integrity verification of ICS data with a certain degree of confidence. Notably by adopting a strategy from digital forensics which consists in focusing on the physical characteristics of the system.

The second aspect is data **traceability**, which we define as **the authentication of each process of transformation of the data from its creation to its end use**. Of course, with cryptographic or watermarking tools, one could achieve such a goal, but the need for passive solutions remains with the same reasons than above. That is why we further discuss (passive) physical characterisation methods, that is all the methods that use (passively) the physical aspect of devices or their environment for security purposes (e.g. identification, authentication, integrity or traceability of information). Physical characterization methods can be grouped into six categories: 1) device modeling, 2) Physical Layer Identification, 3) channel-based fingerprinting, 4) physical inconsistencies, 5) environmental signatures and 6) multimedia forensics specialized techniques. We already mentioned a paper [38] of which one of the method consists of a mathematical model of the device from strong prior knowledge about the system, this constitutes the first category. The second category is the one of Physical Layer Identification (PLI). Danev et al. present an exhaustive overview of the wireless device's PLI in [8]. They define it as follows: «*Physical-layer device identification aims at identifying wireless devices during radio communication by exploiting unique characteristics of their analog (radio) circuitry*». The work of [130] allows us to generalize the definition of the term PLI to wired communication and digital equipment while keeping the passive aspect, that is to say the fact of passively identifying the network devices based on the physical layer of the Open Systems Interconnection model. PLI then becomes synonymous with passive hardware fingerprint-

ing, the expression "hardware fingerprinting" being used in certain paper, in particular that which first inspired our classification [78] (Figure 3). This category appeared in 2005, when Daniels et al. introduced a paradigm called DILON (Detecting Intrusions at Layer ONe) [26], which is based on the characteristics of analog signals used to identify instances of electronic circuits, to initiate the project of the same name. This was a generalization of fingerprinting methods from old radar systems developed since World War II. Although the name DILON has not flourished in the area of security research, the paradigm has been adopted in many works. The third category is Channel-based fingerprinting which refers to the methods of inferring information about the transmitter and its environment using specific wireless channel measurements such as the angle of arrival of a wave. These methods usually uses Received Signal Strength Indication (RSSI), the Channel State Information (CSI) or the Angle Of Arrival (AOA). There also exist Channel based fingerprinting methods that require supplementary equipment, and so out of the scope of this dissertation. The fourth category searches for inconsistencies within the multimedia files with respect to our knowledge of physical laws. For example, in [76] a method was propose to calculate the lighting angle for an object within an image and verify that all objects are illuminated by the same source. Another example is to detect video forgeries thanks to the trajectory of projectile that does not match a parabolic path dictated by the laws of physics [23]. The fifth category, environmental signatures, «*exploit signatures from a sensing environment*»[142]. For example, the signal of Electrical Network Frequency (ENF), the supply frequency of power distribution networks, is known to be embedded in multimedia recordings because of the interference from electromagnetic fields generated by the power source. So it can be used to estimate the time of recording or to detect forgery in multimedia recordings by comparing a database of ENF recordings of the power grid near the recording device. The sixth category, Multimedia forensics specialized techniques, are techniques that cannot yet be grouped in a category based on a common paradigm but are rather specialized for a precise purpose like the issue of identifying imaging type (from a scanner, a computer, a graphic rendering, a digital camera, or a low-end cellphone camera) [118] as cited in [142], the second paper that inspired our classification. Let us note that multimedia forensics is a growing research domain that use techniques from almost every category, thus new categories might appear in the next few years. Finally, these methods do use physical aspect of the device or the environment to the same degree, for example, mentioned methods using environmental signatures does not use at all physical properties of the device. In Figure 3, these six categories are detailed and supported by

examples of works dealing with the type of the corresponding method.

Although not passive, a particularly interesting technique, the Physically Unclonable Functions (PUFs), from the cyber security literature allows us to explore analogous passive solutions. PUF technology is the equivalent of biometrics but applied to physical systems. The PUF procedure takes place in two phases [20]: 1) an enrolment phase to collect numerous Challenge-Response Pairs (CRPs) in a secret CRPs database 2) a verification phase which consists in comparing a new response to a known challenge and erasing the corresponding CRP from the database. Once the database exhausted, the first phase needs to be done again. This way an eavesdropper cannot spoof the system and the PUF provides a key or an ID.

The solutions listed above are most of the time specific to a system or does not take into account the physical process data. We would like to develop general solutions to the problems of integrity and traceability of information from a physical process of a system by focusing on characteristic variations of the system physical process data.

As depicted in Figure 2, we specify the problem of passively characterizing ICSs for security purpose into anomaly detection. The first chapter gives a result useful for the anomaly threshold estimation. The second and third chapters propose different kinds of anomaly scoring functions. More precisely, in the second chapter, anomaly scoring functions are defined based on a pre-processing of physical data that is specific to ICSs and on a widely used anomaly detection model, the autoencoder. However, one could have chosen another machine learning model to benefit from this pre-processing that is the key component for data integrity. In the third chapter, which concerns data traceability, anomaly scoring functions are defined to detect very long term anomalies, of the month order, thanks to the WMW test and a new DL computational unit, the Hyper-Neuron. Finally, the last chapter proposes a new topological metric, that is a metric that does not use label like statistical metric, but that evaluates the performance of an anomaly scorig function based on topological properties.

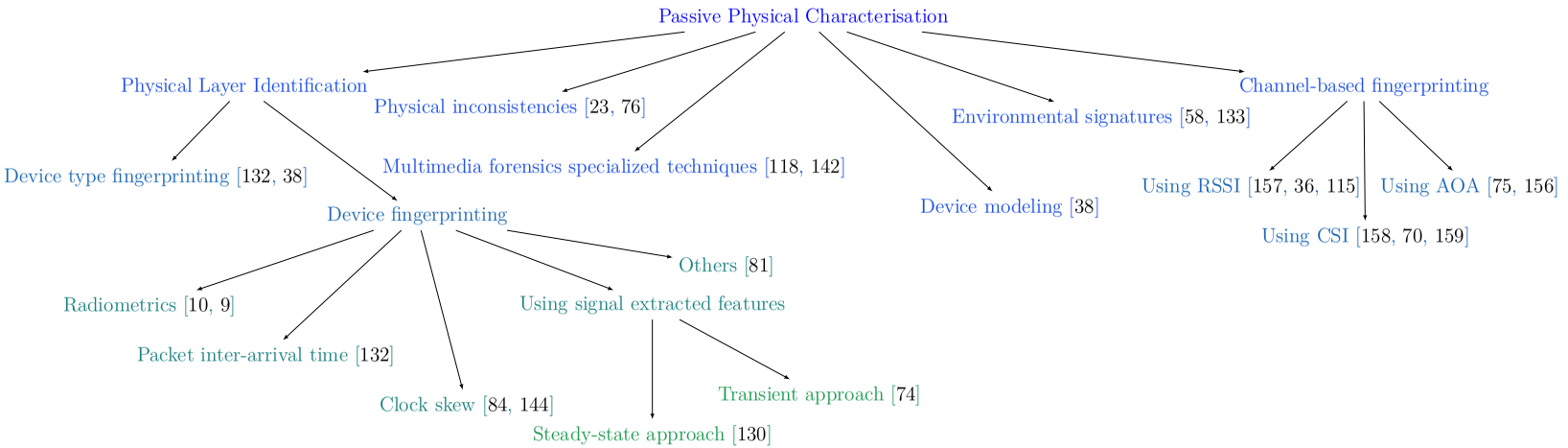



Figure 3 – Classification of passive physical characterisation methods. Some papers present different types of methods and are thus mentionned in several categories.

Chapter 1

# Extreme Value Theory for anomaly detection

we needFor many anomaly detection methods, one needs to first define an anomaly scoring function that can incorporate the experts vision on what is a normal piece of data, then one uses a statistical outlier detection method. This way of proceeding offers a flexibility in the definition of normality, unlike other anomaly detection methods such as One-Class Support Vector Machine (OC-SVM) [136] for example, which, incidentally, is «*the state of the art for estimating* [high density] *regions from high dimensional data*» [145]. Once an anomaly score has been constructed in an unsupervised fashion the most basic solution to complete the anomaly detection method – that is to provide a decision rule – is to define a threshold thanks to a histogram based density estimation. Any score above the threshold would trigger an alert. An expert would decide an acceptable False Positive Rate (FPR) [1] not to be exceeded depending on the number of alerts the security team can tolerate during the system normal regime and find the threshold that maximizes, under this constraint, the True Positive Rate (TPR) [2]. However, if this method is often used in network security for example because of time constraints [53], it suffers from imprecision, especially when relatively small dataset is provided as it can be the case for an ICS. In our case, time is not a constraint. Even the histogram continuous counterpart, the kernel density estimation, is not appropriate for defining the threshold in a well-grounded theoretical framework since only the tail of the distribution is of interest. A more suited method would be the Grubbs's test [60], unfortunately it relies on the assumption that the distribution is Gaussian which is an improper assumption in the case where the random variable is an anomaly score. That is why it may be preferable to use methods from EVT like the one proposed in [140] to limit FPR while maximizing the TPR. Let

1. ... that is the ratio of the number of false alerts over the total number of normal events.
2. ... that is the ratio of the number of alerts for actual abnormal events over the total number of abnormal events.

us start by explaining the general purpose of EVT and the historical context, then we will give an outline of the main results of EVT detailed in the course of Laurent Gardes [45] and their importance in cybersecurity anomaly detection. Then, we will present the more appropriate methods from EVT for our problem. Finally, we will introduce our contribution to this theory, the method called Fast Maxima Sampling.

## 1.1 Introduction to Extreme Value Theory

EVT cares about the modelization of extreme events. The first example that comes to mind is the one of catastrophic events, like freak waves, but we will see that there are many applications of this theory. EVT is built upon the work of Fréchet [40] in 1927 and Fisher and Tippett [37] in 2918. Fréchet first studied the law of largest member between independent and identically distributed (i.i.d.) random variables following a truncated Gaussian distribution. Then Fisher and Tippett characterized all the possible forms of distribution of the extremes in any i.i.d. sample. Tippett first applied the theory to the problem of yarn breakage rates in weaving at the Shirley Institute for which he begun to work in 1925 [143]. In 1939, Weibull proposed a theory of the strength of materials [149]. In 1943, Gnedenko [49] finished the work initiated by Von Mises [111] in 1936, by proving the necessary and sufficient conditions for the characterization found by Fisher and Tippett, giving rise to the now called Fisher–Tippett–Gnedenko theorem. Finally, Gumbel unified EVT with the rework of its own papers he wrote for twenty years [61] where he gave many examples of applications he found in the literature: floods (magnitude of a daily discharge of river), climatology, aeronautics (gust loads), breaking strength of materials, extreme duration of human life, extinction times for bacteria, radioactivity and stock market. In the sequel, "iff" means if and only if and "i.i.d." means independent and identically distributed.

**Definition 1.1.** *Let $H$ be a non-degenerate cumulative distribution function (CDF), and $F$ another CDF. $F$ is said to belong to the maximum domain of attraction of $H$ ($F \in \mathcal{D}_{\mathcal{M}}(H)$) iff there exist two sequences $a_n > 0$ and $b_n$ s.t. $F^n(a_n x + b_n) \to H(x)$ as $n \to \infty$ $\forall x \in \mathbb{R}$. It is equivalent to say that $\frac{Z_n - b_n}{a_n} \xrightarrow{\mathcal{L}} Y$ where $Y$ is a random variable whose CDF is $H$, $Z_n = \max_k(X_k)_{1 \leq k \leq n}$, $X_1, X_2, \ldots, X_n$ i.i.d. with CDF $F$ and $\xrightarrow{\mathcal{L}}$ is the convergence in law.*

EVT mainly aims to study the law of $Z_n = \max_k(X_k)_{1 \leq k \leq n}$ with $X_1, X_2, \ldots, X_n$

a sequence of $n$ random variables with the same marginal CDF $F$. Usually, $(X_i)_i$ are supposed i.i.d. or the sequence is supposed stationary. Let us present, in Definition 1.1, the core concept of EVT which is the maximum domain of attraction of a CDF that is not degenerate—i.e. associated with law whose support has a single value.

**Definition 1.2.** *Let $G$ and $H$ be two non-degenerate CDFs. $G$ and $H$, or their distributions, are of same type if and only if there exist $a > 0$ and $b \in \mathbb{R}$ such that $H(x) = G(ax + b), \forall x \in \mathbb{R}$.*

The maximum domain of attraction is a well defined concept as the CDF associated to it is unique except for a change in the scale and position parameters (cf. Proposition 1.1). Although it seems intuitive, the proof is not straightforward [45]. Moreover, Proposition 1.2 indicates that it represents well the notion of tail of distribution.

**Proposition 1.1.** *Let $H$ and $G$ be two non-degenerate CDFs. If $H$ and $G$ are of the same type, $\mathcal{D}_{\mathcal{M}}(H) = \mathcal{D}_{\mathcal{M}}(G)$, otherwise $\mathcal{D}_{\mathcal{M}}(H) \cap \mathcal{D}_{\mathcal{M}}(G) = \emptyset$.*

**Definition 1.3.** *Let $F$ and $G$ be two CDFs. $F$ and $G$ are said to have proportional tails if they have the same endpoint $x^* = \inf\{x \mid F(x) = 1\} \in \mathbb{R} \cup \{+\infty\}$ and $\lim_{x \to x^*} \frac{1-F(x)}{1-G(x)} \in \mathbb{R}^+$.*

**Proposition 1.2.** *Let $F$ and $G$ be two CDFs with proportional tails and endpoint $x^*$. Let $H$ be a non-degenerate CDF. Then $F \in \mathcal{D}_{\mathcal{M}}(H) \Leftrightarrow G \in \mathcal{D}_{\mathcal{M}}(H^{\frac{1}{c}})$, with $c = \lim_{x \to x^*} \frac{1-F(x)}{1-G(x)}$.*

Now we can formulate the famous theorem of EVT.

**Theorem 1.1** (Fisher–Tippett–Gnedenko)**.** *Let $G$ be a non-degenerate CDF. If there is another CDF $F$ such that $F \in \mathcal{D}_{\mathcal{M}}(G)$, then there exists an extreme-value index $\gamma \in \mathbb{R}$ such that $G$ is of the same type as*

$$H_\gamma : x \in \mathbb{R} \to \mathbb{1}_{1+\gamma x>0}.e^{-(1+\gamma x)^{-\frac{1}{\gamma}}} + \mathbb{1}_{1+\gamma x \leq 0, \gamma<0}$$

*with $H_0 = \lim_{\gamma \to 0} H_\gamma = x \to e^{-e^{-x}}$, i.e. $H_\gamma \propto e^{-(1+\gamma x)^{-\frac{1}{\gamma}}}$, $\propto$ meaning "proportional to".*

Theorem 1.1 tells us that, although there is an infinity of maximum domains of attraction, we can regroup their extreme distributions types in a single formula that depends only the extreme-value index $\gamma$. This family is called the Generalized Extreme Value (GEV) distributions $\{G_{\gamma,\mu,\sigma} : x \in \mathbb{R} \to H_\gamma((x-\mu)/\sigma) \mid \gamma \in \mathbb{R}, \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$. The formulation given in Theorem 1.1 actually comes from Jenkinson and Von Mises, while in

the original theorem, three classes of maximum domains of attraction are represented by three families of extreme distributions with formulations that had not allowed to directly recognize a unified formula. Without giving these formulations, let us present the three classes: the first $\{G_{0,\mu,\sigma} \mid \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$ is referred to as the Gumbel extreme value distributions, the second $\{G_{\gamma,\mu,\sigma} \mid \gamma \in \mathbb{R}^+, \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$ as the Fréchet extreme value distributions, and the third $\{G_{\gamma,\mu,\sigma} \mid \gamma \in \mathbb{R}^-, \mu \in \mathbb{R}, \sigma \in \mathbb{R}^+\}$ as the reversed Weibull extreme value distributions. It is worth noting that not every CDF belongs to a maximum domain of attraction. For example, no CDF with a finite endpoint and a discontinuity in this point belongs to a maximum domain of attraction. It is also the case for the Poisson distribution [45]. However, most continuous CDFs belong to a maximum domain of attraction.

## 1.2 Anomaly detection based on EVT

### 1.2.1 Entailment of EVT in cybersecurity

One can appreciate the importance of Theorem 1.1 as it allows the estimation of the distribution of the random variable $Z_n = \max_k(X_k)_{1 \leq k \leq n}$ with $X_1, X_2, \ldots, X_n$ a sequence of i.i.d. random variables following the law with a CDF $F$ that belongs to a maximum domain of attraction, when $n$ is large enough. Then, once in possession of the estimated distribution $G_{\hat{\gamma},\hat{\mu},\hat{\sigma}}$, one can estimate the extreme quantile (inverse CDF) values for $Z_n$. In the context of anomaly detection, that is when the random variable is the output of an anomaly scoring function, one would naturally favor the estimation of the quantile on the right tail of $G'_{\hat{\gamma},\hat{\mu},\hat{\sigma}}$ to set an anomaly threshold. That is, the threshold would be $q(p) = {G_{\hat{\gamma},\hat{\mu},\hat{\sigma}}}^{-1}(p)$ with $G_{\hat{\gamma},\hat{\mu},\hat{\sigma}}(z) = \mathbb{P}(Z_n \leq z)$, if the security team can tolerate a FPR equal to $1 - p$, and any anomaly score above $q(p)$ would trigger an alert. The value $p$ expresses both the practicality of the anomaly detection method and the practicality of the security team analysis. The more $p$ is close to 1, the more the anomaly scoring function is considered reliable, while the more $p$ is close to 0, the more content analysis the security team have to support during the normal regime. However, this supposes that any abnormal activities would increase the anomaly score. While this is indeed the purpose of such a function, we have to keep in mind that a malicious actor will try to bypass the security. Every security expert agrees now that security through obscurity is

a naive paradigm. So one has to assume that the attacker knows what model, that is what type anomaly scoring function in our case, is used. On this basis, the attacker can replay the features that the anomaly scoring function accepts as input and that he has previously recorded from the normal regime of the system, so that the anomaly score never exceeds $q(p)$. Yet, the random variable $Z_n$ is likely to have a different distribution under this attack. In particular, if the features replayed are from a relatively small record, $Z_n$ is likely to have abnormaly small values, so the left tail of $G'_{\hat{\gamma},\hat{\mu},\hat{\sigma}}$ can also be considered. The decision for the alerts would be when either $Z_n$ is below a certain $q(p_1)$ or above a certain $q(p_2)$, with $p_1 < p_2$ and $1 - p_2 + p_1$ would be the acceptable FPR. Having $p_1 = 0$, that is ignoring the left tail of $G'_{\hat{\gamma},\hat{\mu},\hat{\sigma}}$, is tempting though, because then the procedure to trigger an alert would not require to check if $Z_n > q(p_2)$ but only if $X_n > q(p_2)$, so it would not be necessary to store the past anomaly scores, and the evidence of some anomalies is exactly the $X_n$ that trigger the alert, whereas if the anomaly is detected thanks to $Z_n < q(p_1)$ the pieces of evidence are to be found among $(X_k)_{1 \leq k \leq n}$. To decide about $p_1$, the experts in charge of the online monotoring of a system should therefore assess the capacity of its team to investigate in large log files as well as the usefulness of such task, that is whether it can change the final result of the attack or it is too late. Of course, the security experts do not know which attack they will face, but they can determine the class of the more redoubtable attacks and the practicality of a reaction to them. In conclusion, $p_2 - p_1$ expresses the practicality of the anomaly detection method and the practicality of the security team analysis in the same way than explained above for $p$, and $\frac{p_1}{1-p_2}$ expresses the choice to put more effort on the analysis of the long-term anomalies ($\frac{p_1}{1-p_2} > 1$) or on the one of prompt anomalies ($\frac{p_1}{1-p_2} < 1$).

### 1.2.2 Methods from EVT for anomaly detection challenges in industrial security

Now that we have presented the entailment of EVT in cybersecurity, we can promote methods we have found in the EVT literature and their use for our problem. The main challenge in anomaly detection from physical processes of ICSs is the small size of available datasets. Another challenge is that time series from ICSs processes can have relatively long range dependence, making online anomaly detection more difficult.

Until now, we have assumed that the random variables were i.i.d.. While the fact that

random variables follows the same distribution is a reasonable assumption, their independence can be too restrictive. An interesting result proved by Leadbetter provides weaker assumptions than the independence: for a stationary sequence $(X_k)_{1\leq k\leq n}$ of random variables under a weak condition, noted $D((a_n)_n, (b_n)_n, (X_i)_i)$, on the normalization sequences $(a_n)_n, (b_n)_n$ and on $(X_i)_i$ restricting long range dependence, if $\frac{Z_n - b_n}{a_n} \xrightarrow{\mathcal{L}} Y$, then $Y$ follows a GEV distribution, with $Z_n$ defined as previously. This result is known as the extremal limit theorem [91, 6]. From there, one could directly use the GEV distribution to model the right tail distribution of the anomaly scores. However, the issue of small dataset has to be tackled too since it is a constraint of lots of industrial systems and this is exacerbated by the nature of the random variable $Z_n$. Here comes a related result also proved by Leadbetter. It states that if the weak condition $D((a_n)_n, (b_n)_n, (\hat{X}_i)_i)$ holds for the associated, independent sequence $(\hat{X}_i)_i$ with the same marginal distribution as $(X_i)_i$, and $\frac{\hat{Z}_n - b_n}{a_n} \xrightarrow{\mathcal{L}} \hat{Y} \sim \hat{G}$, then $\frac{Z_n - b_n}{a_n} \xrightarrow{\mathcal{L}} Y \sim G \Rightarrow G = \hat{G}^\theta$ with $\theta \in [0, 1]$ [90, 6]. The parameter $\theta$ is called the extremal index. One can verify that if $\theta > 0$, then $G$ is also GEV distribution. The reader could then object that since the family of distribution is not changed, this new result is not of interest for our problem. But that is not including the fact that adding this new parameter gives the model more freedom so that it can efficiently capture the right tail distribution of the random variable of interest despite the small size of the dataset available as showed in [126]. Moreover, as mentionned in [6], «*the extremal index characterizes the change in the distribution of samples' maxima due to dependence in the sequence* [and] *measures the tendency of extreme values to occur in clusters*». Hence the work of Ribereau et al. [126] tackles both the dependence and the small size of datasets. Their family of distributions $\{G^{1+\lambda}_{\gamma,\mu,\sigma} \mid \gamma \in \mathbb{R}, \mu \in \mathbb{R}, \sigma > 0, \lambda > -1\}$ is called the Skew Generalized Extreme Value (SGEV) distribution [126]. For $-1 < \lambda \leq 0$, the SGEV distributions model the dependency of the data with the extremal index $\theta = \lambda + 1$. Moreover, the authors linked their family of distribution to the Generalized Pareto Distribution (GPD): $G^{1+\lambda}_{\gamma,\mu,\sigma}$ equals, on the interval $[\mu, +\infty[$, the CDF of the maximum $\max_{1\leq k\leq N}(X_k)$ of a random Poisson size sample $(N \sim \mathcal{P}(\lambda + 1))$ of i.i.d. random variables $(X_k)_k$ following a GPD.

The parameter estimation method is detailed in [126] and relies on the Probability Weighted Moments (PWMs) method introduced by Greenwood et al. [57]. Special cases of PWMs used in EVT are $\nu_a = \mathbb{E}(XF^a(X))$ with $F$ the CDF of $X$, since their analytical forms are known when $F = G_{\gamma,\mu,\sigma}$. Ribereau et al. derive the analytical forms of $\nu_a$ when $F = G^{\lambda+1}_{\gamma,\mu,\sigma}$ and explain how to estimate $\gamma, \mu, \sigma$ and $\lambda$ thanks to the estimates of $\nu_0, \nu_1, \nu_2$ and $\nu_3$ [126]. They proposed to use $\frac{1}{n}\sum_{i=1}^{n} X_{(i)}(\frac{i}{n})^a$, but some authors prefer

the unbiased estimator from Landwehr et al.: $\hat{\nu}_a := \frac{1}{n}\sum_{i=a+1}^{n} X_{(i)} \prod_{k=1}^{a} \frac{i-k}{n-k}$ [88, 100] [3], with $(X_{(i)})_i$ the ordered sample and $n > a$. Finally, one can use bootstrap methods to overcome the smallness of the dataset used to estimate $G_{\gamma,\mu,\sigma}^{\lambda+1}$. If $(X_i)_i$ is stationary, under some conditions on the resample size and the long range dependence of $(X_i)_i$ detailed by Fukuchi in Theorem 3.6 in [41], the Efron's bootstrap method is appropriate. In the case of i.i.d. random variables, one can use the Efron's bootstrap method or the permutation bootstrap optimized for the EVT framework as described by Mefleh et al [108]. In the next section, we propose an optimized bootstrap method for maxima sampling.

## 1.3 Fast Maxima Sampling

The Efron's bootstrap method is the baseline for non-parametric sampling [33]. Let us say we have the observations $x_1, x_2, \ldots, x_n$ of a $n$ i.i.d. random variables $(X_i)_{1\leq i\leq n}$. A bootstrap sample of size $m$ is a set of $m$ points $x_1^*, x_2^*, \ldots, x_m^*$ each drawn from $x_1, x_2, \ldots, x_n$ with probility $\frac{1}{n}$. More formally, it is a sample from the empirical CDF $\hat{F}_n$ defined as $x \to \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{x_i\leq x}$. In the context of EVT, we are interested in the maximum $Z$ among a relatively large set of random variables. The traditional method, called the block maxima method [6], is to divide the available sample into $k$ blocks of size $b$, $k \times b$, with $b$ large enough so that the distribution of $Z$ approach a GEV distribution, but not too large so that one can have enough observations to carry out the estimation. As mentionned in the previous section, Mefleh et al propose to use permutation bootstrap optimized for block maxima method [108]. Instead of computing $m$ permutations, and then block maxima on these permutations, to have $m$ estimates they can aggregate, the authors found a way to directly sample these block maxima by computing no permutations but only the block maxima ranks thanks to the rank distribution they have derived on ordered statistics. While this an elegant and efficient way of applying the block maxima method, it needs to generate the rank distribution and estimates come from only $k$ maxima. To palliate these drawbacks, one can use the Efron's bootstrap method to sample a larger set of maxima used to compute an estimate. That is with $k' > k$ bootstrap sample' maxima $\{x_1^*, x_2^*, \ldots, x_b^*\}$, one takes their $k'$ maxima which are given to an estimator. Of course this

3. Lubes et al. made a mistake in their last formulation, they wrote $\frac{1}{n}\sum_{i=1}^{n} X_{(i)} \prod_{k=1}^{a} \frac{i-k}{n-k}$ starting the sum with $i = 1$ instead of $i = a + 1$ (cf. the very last equation in their appendix [100]). This formulation error seems to last in the literature, but it may not have change the experiments' results because one may directly use the following formulation $\frac{1}{n}\sum_{i=1}^{n} X_{(i)} \binom{i-1}{a} / \binom{n-1}{a}$ with the convention $\binom{n}{k} = 0$ if $k > n$.

naive approach is not computationally efficient because it needs $k' \times b$ samples point from the empirical distribution, but one can think of a method to sample the maxima without going through all this process in the same way as maxima on permutation bootstrap are computed without the need of computing the permutations. This the subject of this section. The overall idea is the same as the one of Mefleh et al., that is to use the order.

### 1.3.1 Presentation of the method

Let $F$ be a CDF from which one has to generate samples maxima. If the distribution is discrete, one can consider the values of the support of the distribution in an ordered fashion $x_1 \leq x_2 \ldots \leq x_n$, and generate $x_I$, directly with $\mathbb{P}(I = k) = \frac{k^b-(k-1)^b}{n^b}$, the probability of $I = \max_{i=1:b}(J_i)$ and $(J_i)_i$ i.i.d. uniform random variables on the set of indices $\{1, ..., n\}$. In general, one can use inverse transform sampling with the CDF $F^b$, i.e., $F^{-1}(U^{\frac{1}{b}})$ represents the maximum of a virtual sample of size $b$ with $U \sim \mathcal{U}(0, 1)$. Yet, in some cases, a closed form of $F^{-1}$ does not exist and it can be difficult to approximate $F^{-1}$, or even impossible, thus alternative methods to inverse transform sampling are used, like rejection sampling [147], the Metropolis–Hastings algorithm [109], slice sampling [116], the Bailey's method [4], the simulation of a mixture [47], etc. Here, we propose an algorithm that can transpose such techniques to efficiently generate samples maxima. To introduce our algorithm, let us first present the naive algorithm, $P_{\text{NaiveMS}}(F, b)$, when the sample points are successively discovered such that one can keep track of the latest maximum value. In this naive algorithm, a random variable, noted $\theta$, represents the number of times one recalls the latest maximum value during the successive sampling.

**Procedure** (Naive Maxima Sampling)**.** $P_{\text{NaiveMS}}(F, b)$.
*The naive procedure to generate a maximum from a sample of size b is to generate a first sample point, $x_1$, from the law with F as CDF, store it in memory, then generate the same way $x_2$ and store it in memory if and only if $x_1 < x_2$ and so on until b points have been observed and one stores $x_b$ if $x_{b-1} < x_b$. Then the sample's maximum is the last point stored in memory, $x_{j_\theta}$, where $(j_k)_k$ denote the indices of the points once stored in memory and $\theta$ is the number of times a point has been stored.*

The goal now is to find the law, $\Theta(b)$, of the random variable $\theta$ defined in $P_{\text{NaiveMS}}(F, b)$. [4]

4. Incidentally, knowing $\Theta(n)$ leads to an independence test for ordinal 1rst-order stationary processes: the Z-test can check whether the average number of indices in samples $(s_i)_i$ of size $n$ marked during the

This way one can skip the laborious process of the naive procedure by making the reverse operation in a fast procedure, with $b < n$: we first generate $\theta \sim \Theta(b)$ and then sample, in an increasing order, $\theta$ points.

**Procedure** (Fast Maxima Sampling with $\hat{F}_n$)**.** $P_{\text{FastMS}}(\hat{F}_n, b)$.
*Let $x_1 \leq x_2 \leq \ldots \leq x_n$ the ordered dataset for which $\hat{F}_n$ is the empirical CDF. To generate a maximum from a sample of size $b$, one can first generate $\theta \sim \Theta(b)$, then, $(j_k)_{1 \leq k \leq \theta}$ such that $j_1$ is randomly chosen among $[\![1, n]\!]$ (i.e., with the uniform distribution on $[\![1, n]\!]$), if possible $j_2$ is randomly chosen among $[\![j_1 + 1, n]\!]$ otherwise $\theta' = 1, \ldots$, if possible $j_\theta$ is randomly chosen among $[\![j_{\theta-1} + 1, n]\!]$ otherwise $\theta' = \theta - 1$, always independently of $\theta$. Then if $\theta' \neq \theta$ start again the same procedure with the same $\theta$ until $\theta' = \theta$. Otherwise, the point $x_{j_{\theta'}}$ can be considered as the maximum of a sample of size $b$ from the law with $\hat{F}_n$ as CDF (cf. Corollary 1.1).*

**Procedure** (Fast Maxima Sampling with any CDF $F$)**.** $P_{\text{FastMS}}(F, b)$.
*Let $\theta \sim \Theta(b)$ and $(X_i)_{1 \leq i \leq \theta}$ be a random variable whose CDF is $F \times F_{>X_1} \times \ldots F_{>X_{\theta-1}}$, with $F_{>X} = \frac{F(.)}{1-F(X)} \mathbb{1}_{X<.}$ and $\forall i \in [\![1, \theta]\!]$, $\theta \perp\!\!\!\perp X_i$. The observation of $X_\theta$ can be considered as the maximum of a sample of size $b$ from the law with $F$ as CDF (cf. Corollary 1.1).*

**Remark 1.1.** *When $F^{-1}$ is known, in particular when $F$ is discrete like the empirical CDF $\hat{F}_n$, Fast Maxima Sampling is useless since the inverse transform sampling can be applied with $F^n$. Yet, we still describe $P_{\text{FastMS}}(\hat{F}_n, b)$ as the intuition is easier to grasp that in its general form and it shows how to handle a CDF $F$ that is from a mixture of discrete and continuous distribution. In some cases, one must use an alternative method $M$ to inverse transform sampling. If so, Fast Maxima Sampling is useful because, with the help of $M$, it produces the increasing sample of size $\theta \sim \Theta(b)$ exponentially faster than taking the maximum of a bootstrap sample of size $b$ produced with $M$ (cf. Proposition 1.4).*

### 1.3.2 Properties and coherence of Fast Maxima Sampling

In this section, we prove that Fast Maxima Sampling $P_{\text{FastMS}}(F, b)$ is exponentially faster than the naive canonical method $P_{\text{NaiveMS}}(F, b)$ and that their outputs follow the same law, besides elucidating practical considerations.

similar procedure to $(P_{\text{NaiveMS}}(s_i, n))_i$—with the difference that the sequence from which the indices are gathered is not a random size bootstrap sample from $(s_i)_i$ but exactly $(s_i)_i$—is $\mathbb{E}_{\Theta(n)}(\theta)$. The variables are believed to be dependent when the null hypothesis is rejected.

**Lemma 1.1.** *The support of $\Theta(n)$ is $[\![1,n]\!]$. Let $\theta \sim \Theta(n)$, we have $\mathbb{P}(\theta=1)=\frac{1}{n}$ and:*

$$\forall i \in [\![2,n]\!], \mathbb{P}(\theta=i) = \frac{1}{n} \sum_{1\leq j_1<j_2<\ldots<j_{i-1}\leq n-1} \prod_{k=1}^{i-1} \frac{1}{j_k}$$

**Proposition 1.3.** *Let $\theta \sim \Theta(n)$. Then $\mathbb{E}(\theta) = \sum_{1\leq k\leq n} \frac{1}{k}$.*

**Proposition 1.4.** *The average duration of $P_{\text{FastMS}}(\hat{F}_n, b)$ (resp. $P_{\text{FastMS}}(F,b)$ with $F$ continuous) is upper bounded by $\sum_{1\leq k\leq b} \frac{1}{k} + (b-1)\sum_{k=2}^{b} \mathbb{P}(\theta_b=k)\frac{\mathbb{P}(\theta_n<k)}{\mathbb{P}(\theta_n\geq k)}$, with $\theta_k \sim \Theta(k), \theta_n \perp\!\!\!\perp \theta_b$ (resp. is $\sum_{1\leq k\leq b} \frac{1}{k}$).*

*Proof of Proposition 1.4.* Note $T(b,n)$ the running time of $P_{\text{FastMS}}(\hat{F}_n, b)$, $(\theta_{n,j})_j \overset{i.i.d.}{\sim} \Theta(n)$ and $\tilde{\theta}_{n,i} = \max_{j=1,..,i}(\theta_{n,j})$. We have $T(b,n) = \theta_b + \sum_{i\geq 1} \tilde{\theta}_{n,i} \mathbb{1}_{\tilde{\theta}_{n,i}<\theta_b}$, so the expected time is: $\mathbb{E}(T(b,n)) \leq \mathbb{E}(\theta_b) + (b-1)\mathbb{E}(\sum_{i\geq 1}\prod_{j=1}^{i}\mathbb{P}(\theta_{n,j}<\theta_b \,|\, \theta_b)) = \mathbb{E}(\theta_b) + (b-1)\mathbb{E}(\sum_{i\geq 1}\mathbb{P}(\theta_n<\theta_b \,|\, \theta_b)^i)$ So with Proposition 1.3: $\mathbb{E}(T(b,n)) \leq \sum_{1\leq k\leq b} \frac{1}{k} + (b-1)\sum_{k=2}^{b} \mathbb{P}(\theta_b=k)\frac{\mathbb{P}(\theta_n<k)}{\mathbb{P}(\theta_n\geq k)}$.

As for $P_{\text{FastMS}}(F,b)$, the duration is $\theta_b$, so the expected duration is $\sum_{1\leq k\leq b} \frac{1}{k}$. □

For $P_{\text{FastMS}}(\hat{F}_n, b)$ to be fast, one has to make $\mathbb{P}(\theta_n < \theta_b)$ small to efficiently compute $\Theta$. This is the goal of Lemma 1.2, thanks to which one can quickly compute $\Theta(k+1)$ thanks to $\Theta(k)$. See Remark 1.3 for dealing with very large $n$.

**Lemma 1.2.** *Noting $\theta_k \sim \Theta(k)$, $\forall (i,n), \mathbb{P}(\theta_n=i) = \frac{n-1}{n}\mathbb{P}(\theta_{n-1}=i) + \frac{1}{n}\mathbb{P}(\theta_{n-1}=i-1)$.*

*Proof of Lemma 1.2.* Using Lemma 1.1, we have:

$$\begin{aligned}
&\mathbb{P}(\theta_n=i) = \mathbb{P}(\theta_n=i \wedge \theta_n>1) + \mathbb{P}(\theta_n=i\wedge\theta_n=1) = \mathbb{1}_{i>1}\mathbb{P}(\theta_n=i) + \mathbb{1}_{i=1}\mathbb{P}(\theta_n=1)\\
&= \mathbb{1}_{i>1}\frac{1}{n} \sum_{1\leq j_1<j_2<\ldots<j_{i-1}\leq n-1} \prod_{k=1}^{i-1}\frac{1}{j_k} + \mathbb{1}_{i=1}\frac{1}{n}\\
&= \frac{n-1}{n}\mathbb{1}_{i>1}\Big(\sum_{1\leq j_1<j_2<\ldots<j_{i-1}=n-1} \frac{1}{n-1}\prod_{k=1}^{i-1}\frac{1}{j_k} + \sum_{1\leq j_1<j_2<\ldots<j_{i-1}<n-1} \frac{1}{n-1}\prod_{k=1}^{i-1}\frac{1}{j_k}\Big) + \mathbb{1}_{i=1}\frac{1}{n}\\
&= \frac{n-1}{n}\mathbb{1}_{i>1}\Big(\sum_{1\leq j_1<j_2<\ldots<j_{i-2}\leq n-2} \frac{1}{(n-1)^2}\prod_{k=1}^{i-2}\frac{1}{j_k} + \sum_{1\leq j_1<j_2<\ldots<j_{i-1}\leq n-2} \frac{1}{n-1}\prod_{k=1}^{i-1}\frac{1}{j_k}\Big) + \frac{\mathbb{1}_{i=1}}{n}\\
&= \frac{n-1}{n}\mathbb{1}_{i>1}\Big(\frac{1}{(n-1)}\mathbb{P}(\theta_{n-1}=i-1) + \mathbb{P}(\theta_{n-1}=i)\Big) + \frac{\mathbb{1}_{i=1}}{n} = \frac{n-1}{n}\mathbb{P}(\theta_{n-1}=i) + \frac{1}{n}\mathbb{P}(\theta_{n-1}=i-1)
\end{aligned}$$

The last equation comes from $\mathbb{P}(\theta_{n-1}=1)=\frac{1}{n-1}, \mathbb{P}(\theta_{n-1}=0)=0$ when $i=1$. □

*Proof of Lemma 1.1.* With $J_i$ the random variable for the $i$-th rank marked during the naive procedure, $\mathbb{P}(\theta = 1) = \mathbb{P}(J_1 = n) = \frac{1}{n}$. Since $\hat{F}_n$ is used for the sampling, we have $J_1 \sim \mathcal{U}(\llbracket 1, n \rrbracket)$ and for all $i \in \llbracket 2, \theta \rrbracket$, $(J_i \mid J_{i-1}) \sim \mathcal{U}(\llbracket J_{i-1} + 1, n \rrbracket)$, so:

$$\begin{aligned}
\mathbb{P}(\theta = i) &= \sum_{j_1=1}^{n-1} \mathbb{P}(\theta = i \mid J_1 = j_1)\mathbb{P}(J_1 = j_1) \\
&= \sum_{j_1=1}^{n-1} \sum_{j_2=j_1+1}^{n-1} \mathbb{P}(\theta = i \mid J_1 = j_1, J_2 = j_2)\mathbb{P}(J_2 = j_2 \mid J_1 = j_1)\frac{1}{n} \\
&= \sum_{j_1=1}^{n-1} \sum_{j_2=j_1+1}^{n-1} \mathbb{P}(\theta = i \mid J_2 = j_2)\frac{1}{n-j_1}\frac{1}{n} = \dots \\
&= \sum_{1 \le j_1 < j_2 < \dots < j_{i-1} \le n-1} \mathbb{P}(\theta = i \mid J_{i-1} = j_{i-1})\frac{1}{n-j_{i-2}} \dots \frac{1}{n-j_1}\frac{1}{n} \\
&= \sum_{1 \le j_1 < j_2 < \dots < j_{i-1} \le n-1} \frac{1}{n-j_{i-1}}\frac{1}{n-j_{i-2}} \dots \frac{1}{n-j_1}\frac{1}{n}
\end{aligned}$$

Finally, we can perform successive changes of variable:

$$\begin{aligned}
\mathbb{P}(\theta = i) =& \frac{1}{n} \sum_{1 \le j_1 < j_2 < \dots < j_{i-1} \le n-1} \frac{1}{n-j_1} \dots \frac{1}{n-j_{i-2}}\frac{1}{n-j_{i-1}} \\
=& \frac{1}{n} \sum_{1 \le j_1 < j_2 < \dots < j_{i-2} \le n-1} \frac{1}{n-j_1} \dots \frac{1}{n-j_{i-2}} \sum_{j_{i-1}=j_{i-2}+1}^{n-1} \frac{1}{n-j_{i-1}} \\
=& \frac{1}{n} \sum_{1 \le j_1 < j_2 < \dots < j_{i-2} \le n-1} \frac{1}{n-j_1} \dots \frac{1}{n-j_{i-2}} \sum_{j'_1=1}^{n-j_{i-2}-1} \frac{1}{j'_1} \\
=& \frac{1}{n} \sum_{1 \le j_1 < j_2 < \dots < j_{i-2} \le n-1} \frac{1}{n-j_1} \dots \frac{1}{n-j_{i-2}} \sum_{j'_1=1}^{n-1} \frac{1}{j'_1} \mathbb{1}_{j'_1 < n-j_{i-2}} \\
=& \frac{1}{n} \sum_{1 \le j_1 < j_2 < \dots < j_{i-3} \le n-1} \frac{1}{n-j_1} \dots \frac{1}{n-j_{i-3}} \sum_{j_{i-2}=j_{i-3}+1}^{n-1} \frac{1}{n-j_{i-2}} \sum_{j'_1=1}^{n-1} \frac{1}{j'_1} \mathbb{1}_{j'_1 < n-j_{i-2}} \\
=& \frac{1}{n} \sum_{1 \le j_1 < j_2 < \dots < j_{i-3} \le n-1} \frac{1}{n-j_1} \dots \frac{1}{n-j_{i-3}} \sum_{j'_2=1}^{n-1} \sum_{j'_1=1}^{n-1} \frac{1}{j'_2}\frac{1}{j'_1} \mathbb{1}_{j'_1 < j'_2 < n-j_{i-3}} \\
=& \dots \\
=& \frac{1}{n} \sum_{1 \le j'_1 < j'_2 < \dots < j'_{i-1} \le n-1} \prod_{k=1}^{i-1} \frac{1}{j'_k}
\end{aligned}$$

□

*Proof of Proposition 1.3.* Let us note $\theta \sim \Theta(n)$ and $\phi_n = \mathbb{E}_{\Theta(n)}(\theta)$. Let us make an induction on the size of the dataset $n$. Obviously, $\mathcal{P}_{\Theta(2)}(\theta = 1) = \mathcal{P}_{\Theta(2)}(\theta = 2) = \frac{1}{2}$ and $\mathcal{P}_{\Theta(1)}(\theta = 1) = 1$.

With Lemma 1.1 and the convention $\sum_{\emptyset} = 1$ for $i = 1$ in the sums:

$$
\begin{aligned}
\phi_n = \mathbb{E}_{\Theta(n)}(\theta) &= \sum_{i=1}^{n} i\mathbb{P}_{\Theta(n)}(\theta = i) = \sum_{i=1}^{n} \frac{i}{n} \sum_{1\le j_1<j_2<\ldots<j_{i-1}\le n-1} \prod_{k=1}^{i-1} \frac{1}{j_k} \\
&= \sum_{i=1}^{\boldsymbol{n-1}} \frac{i}{n} \sum_{1\le j_1<j_2<\ldots<j_{i-1}\le n-1} \prod_{k=1}^{i-1} \frac{1}{j_k} + \frac{n}{n} \sum_{1\le j_1<j_2<\ldots<j_{n-1}\le n-1} \prod_{k=1}^{n-1} \frac{1}{j_k} \\
&= \frac{n-1}{n} \sum_{i=1}^{n-1} \frac{i}{n-1} \sum_{1\le j_1<j_2<\ldots<j_{i-2}<j_{i-1}<\boldsymbol{n-1}} \prod_{k=1}^{i-1} \frac{1}{j_k} \\
&\quad + \sum_{\boldsymbol{i=2}}^{n-1} \frac{i}{n} \sum_{1\le j_1<j_2<\ldots<j_{i-2}<j_{i-1}=\boldsymbol{n-1}} \prod_{k=1}^{i-1} \frac{1}{j_k} + \prod_{k=1}^{n-1} \frac{1}{k} \\
&= \frac{n-1}{n}\phi_{n-1} + \frac{1}{n} \sum_{i=2}^{n-1} \frac{\boldsymbol{i-1}}{n-1} \sum_{1\le j_1<j_2<\ldots<j_{i-2}\le n-2} \prod_{k=1}^{i-2} \frac{1}{j_k} \\
&\quad + \frac{1}{n} \sum_{i=2}^{n-1} \frac{\mathbf{1}}{n-1} \sum_{1\le j_1<j_2<\ldots<j_{i-2}\le n-2} \prod_{k=1}^{i-2} \frac{1}{j_k} + \prod_{k=1}^{n-1} \frac{1}{k} \\
&= \frac{n-1}{n}\phi_{n-1} + \frac{1}{n} \sum_{\boldsymbol{l=1}}^{n-2} \frac{l}{n-1} \sum_{1\le j_1<j_2<\ldots<j_{l-1}\le n-2} \prod_{k=1}^{l-1} \frac{1}{j_k} \\
&\quad + \frac{1}{n} \sum_{\boldsymbol{l=1}}^{n-2} \frac{1}{n-1} \sum_{1\le j_1<j_2<\ldots<j_{l-1}\le n-2} \prod_{k=1}^{l-1} \frac{1}{j_k} + \prod_{k=1}^{n-1} \frac{1}{k} \\
&= \frac{n-1}{n}\phi_{n-1} + \frac{1}{n}(\phi_{n-1} - \prod_{k=1}^{n-2} \frac{1}{k}) + \frac{1}{n}(1 - \prod_{k=1}^{n-1} \frac{1}{k}) + \prod_{k=1}^{n-1} \frac{1}{k} \\
&= \phi_{n-1} - \frac{1}{n} \prod_{k=1}^{n-2} \frac{1}{k} + \frac{1}{n} - \frac{1}{n} \prod_{k=1}^{n-1} \frac{1}{k} + \prod_{k=1}^{n-1} \frac{1}{k} \\
&= \phi_{n-1} + \frac{1}{n} - (n-1) \prod_{k=1}^{n} \frac{1}{k} - \prod_{k=1}^{n} \frac{1}{k} + n \prod_{k=1}^{n} \frac{1}{k} = \phi_{n-1} + \frac{1}{n}
\end{aligned}
$$

Another way to prove this is to use Lemma 1.2.

By induction, we have $\phi_n = \sum_{1\le k\le n} \frac{1}{k}$. □

**Remark 1.2.** *As a comparison, the naive procedure $P_{\text{NaiveMS}}(F, b)$ exactly takes $b$ iterations whereas the expected time for our procedure $P_{\text{FastMS}}(F, b)$ is $\sum_{1\le k\le b} \frac{1}{k} \approx \ln(b)$ when $F$ has continuous right tail, hence the name Fast Maxima Sampling.*

**Proposition 1.5.** *Let $F$ a continuous-tailed CDF. Let $X_1, X_2, \ldots, X_m$ be i.i.d. random variables with $F$ as CDF. Let $\theta \sim \Theta(m)$ and $(\tilde{X}_i)_{1\leq i\leq\theta}$ be a multidimensional random variable with a CDF equal to $F \times F_{>\tilde{X}_1} \times F_{>\tilde{X}_2} \times \ldots \times F_{>\tilde{X}_{\theta-1}}$, with $F_{>X} = \frac{F(.)}{1-F(X)}\mathbb{1}_{X<.}$ such that $\forall i \in [\![1, \theta]\!]$, $\theta \perp\!\!\!\perp \tilde{X}_i$. Then, we have $\max_{1\leq i\leq m}(X_i) \sim \max_{1\leq i\leq\theta}(\tilde{X}_i)$.*

*Proof of Proposition 1.5.* To prove that two random variables $V_1$ and $V_2$ follows the same law, one can construct a third random variable $V_3$ such that $\mathcal{L}(V_3) = \mathcal{L}(V_2)$ and $V_3 = V_1$. Let $X_1, X_2, \ldots, X_m$ be i.i.d. random variables with $F$ as CDF. Let $R_i$ be the random variables such that $X_{R_1} \leq X_{R_2} \leq \ldots \leq X_{R_m}$. Let us define $Y_1 = X_{R_{J_1}}$ with $J_1 \sim \mathcal{U}([\![1, n]\!])$, and $\forall i \in [\![2, S]\!]$: $Y_i = X_{R_{J_i}}$ with $(J_i \mid J_{i-1}) \sim \mathcal{U}([\![J_{i-1}+1, m]\!])$ and $J_S$ the last index that can be defined this way, i.e. $J_S = m$. By definition, $S \sim \Theta(m)$. Since $X_1, X_2, \ldots, X_m$ are i.i.d., $J_i \perp\!\!\!\perp (J_k)_{1\leq k<i-1} \mid J_{i-1}$, $\forall i \in [\![1, S]\!]$, $S \perp\!\!\!\perp Y_i$, and the CDF of $(Y_i)_{1\leq i\leq S}$ is given by $F \times F_{>Y_1} \ldots \times F_{>Y_{S-1}}$. Yet, by construction, $\max_{1\leq i\leq m}(X_i) = \max_{1\leq i\leq S}(Y_i)$. □

**Corollary 1.1** (of Proposition 1.5)**.** *The output of the procedure $P_{\text{FastMS}}(F, b)$ and of its naive counterpart, $P_{\text{NaiveMS}}(F, b)$, follow the same law.*

*Proof of Corollary 1.1.* If $F$ has a continuous right tail, it is a direct application of Proposition 1.5. Otherwise, one can assume w.l.o.g. that $F = \hat{F}_n$ with $D = \{(x_i, i) \mid 1 \leq i \leq n\}$ such that $\forall k \in [\![1, n-1]\!], x_k \leq x_{k+1}$, and consider, instead of $\hat{F}_n$, the following CDF $\tilde{G} : x \to \frac{nG(x)}{n+1}\mathbb{1}_{x\leq n+\frac{n+1}{n}} + \mathbb{1}_{0<x-n\leq\frac{n+1}{n}}\frac{n(x-n)}{(n+1)^2} + \mathbb{1}_{x>\frac{n+1}{n}+n}$ with $G : x \to \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{i\leq x}$. Proposition 1.5 ensures that $\max_{1\leq i\leq m}(I_i)$ and $\max_{1\leq i\leq\theta}(I'_i)$ follow the same distribution if $I_1, \ldots, I_m$ are i.i.d. random variables with $\tilde{G}$ as CDF and $\theta \sim \Theta(m)$ and $(I'_i)_{1\leq i\leq\theta}$ a multidimensional random variable with a CDF equal to $\tilde{G} \times \tilde{G}_{>I'_1} \ldots \times \tilde{G}_{>I'_{\theta-1}}$ such that $\forall i \in [\![1, \theta]\!], \theta \perp\!\!\!\perp I'_i$. If we discard values of $I = \max_{1\leq i\leq m}(I_i)$ strictly above $n$, we have $(x_I \mid I \leq n) \sim \mathcal{L}(P_{\text{NaiveMS}}(D, m))$. Likewise, $(x_{I'} \mid I' \leq n) \sim \mathcal{L}(P_{\text{FastMS}}(\hat{F}_n, m))$, for $I' = \max_{1\leq i\leq\theta}(I'_i)$. Yet $\mathcal{L}(I) = \mathcal{L}(I')$, so $\mathcal{L}(P_{\text{FastMS}}(\hat{F}_n, m)) = \mathcal{L}(P_{\text{NaiveMS}}(D, m))$. □

With Fast Maxima Sampling one can therefore efficiently generate maxima from *virtual* – that is not actually generated – bootstrap samples in order to benefit from the famous theorem Fisher–Tippett–Gnedenko when using an alternative method to inverse transform sampling is mandatory.

**Proposition 1.6.** *Let $\theta \sim \Theta(n)$. Then $\mathbb{V}(\theta) = \sum_{k=1}^{n}\frac{1}{k}(1-\frac{1}{k})$.*

*Proof of Proposition 1.6.* With Lemma 1.2 and Proposition 1.3:

$$\begin{aligned}
\mathbb{E}_{\Theta(n)}(\theta_n^2) =& \sum_{i=1}^{n} i^2 \mathbb{P}_{\Theta(n)}(\theta_n = i) = \sum_{i=1}^{n} i^2 \left(\frac{n-1}{n}\mathbb{P}_{\Theta(n-1)}(\theta_{n-1} = i) + \frac{1}{n}\mathbb{P}_{\Theta(n-1)}(\theta_{n-1} = i-1)\right) \\
=& \frac{n-1}{n}\mathbb{E}_{\Theta(n-1)}(\theta_{n-1}^2) + \sum_{i=1}^{n}((i-1)^2 + 2i - 1)\frac{1}{n}\mathbb{P}_{\Theta(n-1)}(\theta_{n-1} = i-1) \\
=& \frac{n-1}{n}\mathbb{E}_{\Theta(n-1)}(\theta_{n-1}^2) + \sum_{i=1}^{n}(i-1)^2\frac{1}{n}\mathbb{P}_{\Theta(n-1)}(\theta_{n-1} = i-1) \\
&+ \frac{2}{n}\sum_{i=1}^{n} i\mathbb{P}_{\Theta(n-1)}(\theta_{n-1} = i-1) - \frac{1}{n} \\
=& \mathbb{E}_{\Theta(n-1)}(\theta_{n-1}^2) + \frac{2}{n}(\mathbb{E}_{\Theta(n-1)}(\theta_{n-1}) + 1) - \frac{1}{n} = \mathbb{E}_{\Theta(n-1)}(\theta_{n-1}^2) + \frac{1}{n}(2\sum_{k=1}^{n-1}\frac{1}{k} + 1) \\
=& \ldots = \sum_{j=1}^{n}\frac{1}{j}(2\sum_{k=1}^{j-1}\frac{1}{k} + 1); \text{ with the convention } \sum_{\emptyset} = 0
\end{aligned}$$

Since $\mathbb{V}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2$: $\mathbb{V}(\theta_n) = \sum_{j=1}^{n}\frac{1}{j}(2\sum_{k=1}^{j-1}\frac{1}{k} + 1) - (\sum_{k=1}^{n}\frac{1}{k})^2$, finally we have $\mathbb{V}(\theta_n) = \sum_{j=1}^{n}\frac{1}{j} + 2\sum_{j=1}^{n}\sum_{k=1}^{j-1}\frac{1}{jk} - \sum_{\substack{j,k=1 \\ j\neq k}}^{n}\frac{1}{jk} - \sum_{k=1}^{n}\frac{1}{k^2} = \sum_{k=1}^{n}\frac{1}{k} - \sum_{k=1}^{n}\frac{1}{k^2}$. □

**Remark 1.3.** *Corollary 1.2 tells us that $\Theta(n)$ is in fact a subfamily of the family of Poisson binomial distributions that model the sum of $n$ independent Bernoulli trials. Incidentally, it means that $\frac{1}{n}\sum_{1\leq j_1<j_2<\ldots<j_{i-1}\leq n-1}\prod_{k=1}^{i-1}\frac{1}{j_k} = \sum_{A\in\mathscr{P}([\![1,n]\!]),|A|=i}\prod_{k\in A}\frac{1}{k}\prod_{k'\in A^c}(1-\frac{1}{k'})$. Yet, when $n$ increases, these distributions are known to be close to a Poisson distribution. Since $\Theta(n)$ requires an involved computation when $n$ is large, it is natural to search for an approximation of $\Theta(n)$ thanks to a Poisson distribution. We know that the variance of Poisson $\mathcal{P}(\lambda)$ is $\lambda$ and $\Theta(n)$ first increases and then decreases (Lemma 1.4), thus $\Theta(n)$ and $\mathcal{P}(\mathbb{V}_{\Theta(n)}(\theta_n))$ might have similar shapes, but different means. One can shift the Poisson distribution, so that both the variances and the means are respectively close to each other. Figure 1.1 show that a properly shifted Poisson distribution is a good approximation even for relatively small $n$. Hereafter, the appropriate shift is derived from the asymptotic difference between the mean and the variance of our law $\Theta(n)$ when $n$ increases.*

**Lemma 1.3.** *Let $\theta_n \sim \Theta(n)$, $\mathbb{E}_{\Theta(n)}(e^{t\theta_n}) = \mathbb{E}_{\Theta(n-1)}(e^{t\theta_{n-1}})(\frac{n-1}{n} + \frac{e^t}{n}) = \prod_{k=1}^{n}(1 + \frac{e^t-1}{k})$.*

*Proof of Lemma 1.3.* Direct induction with the result of Lemma 1.2. □

**Corollary 1.2** (of Lemma 1.3)**.** *$\Theta(n)$ is a Poisson binomial distribution of parameters $(\frac{1}{k})_{1\leq k\leq n}$ (same moment-generating function) and we retrieve the mean and the variance.*

Let us define $g_n$ as follows $g_n(u,0) = (1-u)e^{-\lambda_n}$, with $\lambda_n = \mathbb{V}_{\Theta_n}(\theta_n)$ and $\forall k > 0$, $u \in [0,1]$, $g_n(u,k) = (1-u)\frac{e^{-\lambda_n}\lambda_n^k}{k!} + u\frac{e^{-\lambda_n}\lambda_n^{k-1}}{(k-1)!}$, and $f_n(x) = \mathbb{1}_{x>-1} g_n(\lceil x \rceil - x, \lceil x \rceil)$. It follows that $\forall c > 0, \sum_k f_n(k-c) = 1$, i.e. $f_n(.-c)$ is a density function, and of course the mean is shifted by $c$, $\mathbb{E}_{X\sim f_n(.-c)}(X) = \lambda_n + c$. The fact that $k \to f_n(k-c)$ is a density function is immediate, let us prove that the mean is indeed $\lambda_n + c$, with $c = m + u, u \in [0,1]$:

$$\begin{aligned}
\mathbb{E}_{X\sim f_n(.-c)}(X) =& \sum_{k\geq 0} k f_n(k-c) = \sum_{k\geq m} k g_n(\lceil k-c \rceil - (k-c), \lceil k-c \rceil) \\
=& \sum_{k\geq 0} (k+m) g_n(\lceil k-u \rceil - (k-u), \lceil k-u \rceil) \\
=& \sum_{k\geq 0} k g_n(u,k) + m \\
=& (1-u)\sum_{k\geq 0} k\frac{e^{-\lambda_n}\lambda_n^k}{k!} + u\sum_{k\geq 1} k\frac{e^{-\lambda_n}\lambda_n^{k-1}}{(k-1)!} + m \\
=& (1-u)\lambda_n + u\sum_{k\geq 1}(k-1)\frac{e^{-\lambda_n}\lambda_n^{k-1}}{(k-1)!} + u + m \\
=& \lambda_n + c
\end{aligned}$$

As explained in Remark 1.3, we assume that, for large $n$, the shapes of the distributions $\Theta(n)$ and $\mathcal{P}(\mathbb{V}_{\Theta(n)}(\theta_n))$ are simmilar. Hence, we need to shift the Poisson distribution so the mean are the same to better approximate $\Theta(n)$, yet the difference between the means is $|\mathbb{E}_{X\sim f_n(.-c)}(X) - \mathbb{E}_{\Theta(n)}(\theta_n)| = |c - \sum_k^n \frac{1}{k^2}|$, so the most appropriate shift is $c = \sum_k^n \frac{1}{k^2}$. However, we are interested in the asymptotic case. Fortunately, $\sum_k^n \frac{1}{k^2} \underset{n\to+\infty}{\to} \frac{\pi^2}{6}$, therefore $k \to f_n(k - \frac{\pi^2}{6})$ approximates well $\mathbb{P}_{\Theta(n)}$ for large $n$. Furthermore, $\lambda_n$ can be replaced by $\ln(n) + \gamma - \frac{\pi^2}{6}$ with $\gamma$ the Euler's constant for practicality. Then, with very large datasets, we have $\mathbb{P}_{\Theta(n)}(\theta_n \leq k) \approx \sum_{i=0}^k f_n(i-c) = F_{\mathcal{P}(\lambda_n)}(k - \lfloor c \rfloor) - \frac{\mathbb{1}_{k>\lfloor c \rfloor}(c - \lfloor c \rfloor)e^{-\lambda_n}\lambda_n^{k-\lfloor c \rfloor}}{(k-\lfloor c \rfloor)!}$, with $c = \frac{\pi^2}{6}$, and one can use this to ensure that $\mathbb{P}(\theta_n < \theta'_b) = \sum_{j=1}^b \mathbb{P}_{\Theta(n)}(\theta_n < j)\mathbb{P}_{\Theta(b)}(\theta'_b = j)$ is small with few computations as the Poisson CDF is known [5]. Although not completely characterized, such a result is similar to those found in the literature arising from [89] which stated what is now called Le Cam's theorem. Figure 1.1 shows the difference between $\Theta(n)$ and its approximation with a shifted Poisson for for $n = 10, 100, 1000$ and 10000.

**Lemma 1.4.** *Let $n \in \mathbb{N}$. Then, $\mathbb{P}_{\Theta(n)}$ increases and then decreases with at most two consecutive modes. The sequence of the last of mode $k_n^*$ of $\Theta(n)$ is non-decreasing.*

5. $F_{\mathcal{P}(\lambda)}(k) = \frac{\Gamma(\lfloor k+1 \rfloor, \lambda)}{\lfloor k \rfloor!}$, with $\Gamma(s,x) = \int_x^\infty t^{s-1}e^{-t}dt$.

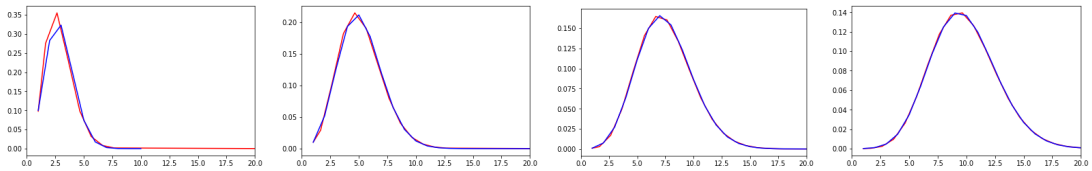


Figure 1.1 – Density functions of $\Theta(n)$ (blue) and $\mathcal{P}(\mathbb{V}_{\Theta(n)}(\theta_n))$ shifted to the right by $\frac{\pi^2}{6}$ (red) – that is with density $k \to f_n(k - \frac{\pi^2}{6})$ – for $n = 10, 100, 1000$ and $10000$ (cf. Remark 1.3).

**Remark 1.4.** *The first part of Lemma 1.4 is a direct consequence of the similar statement in Darroch's work [27] on the general case of a Poisson binomial distribution, but we present here the proof in the case of $\Theta(n)$, which is much simpler.*

*Proof of Lemma 1.4.* Using Lemma 1.2, one can do an induction on $n$. The base case is immediate, since $\Theta(1)$ is the dirac centered at 1. Let us assume that $\mathbb{P}_{\Theta(n-1)}$ increases and then decreases with at most two consecutive modes. Let $k$ be an integer smaller than the first mode or equal to the first mode of $\Theta(n-1)$. By the inductive hypothesis $\forall i \leq k, \mathbb{P}_{\Theta(n-1)}(i) > \mathbb{P}_{\Theta(n-1)}(i-1)$, so with Lemma 1.2:

$$\begin{aligned} \mathbb{P}(\theta_n = k) =& \frac{n-1}{n}\mathbb{P}(\theta_{n-1} = k) + \frac{1}{n}\mathbb{P}(\theta_{n-1} = k-1) \\ >& \frac{n-1}{n}\mathbb{P}(\theta_{n-1} = k-1) + \frac{1}{n}\mathbb{P}(\theta_{n-1} = k-2) = \mathbb{P}(\theta_n = k-1) \end{aligned} \tag{1.1}$$

Let $k$ be an integer greater than the last mode of $\Theta(n-1)$. By the inductive hypothesis $\forall i > k, \mathbb{P}_{\Theta(n-1)}(i) < \mathbb{P}_{\Theta(n-1)}(i-1)$, so with Lemma 1.2:

$$\begin{aligned} \mathbb{P}(\theta_n = k+1) =& \frac{n-1}{n}\mathbb{P}(\theta_{n-1} = k+1) + \frac{1}{n}\mathbb{P}(\theta_{n-1} = k) \\ <& \frac{n-1}{n}\mathbb{P}(\theta_{n-1} = k) + \frac{1}{n}\mathbb{P}(\theta_{n-1} = k-1) = \mathbb{P}(\theta_n = k) \end{aligned} \tag{1.2}$$

Now, let us consider the modes. If $\Theta(n-1)$ has only one mode $k^*$, (1.1) tells us that $\mathbb{P}(\theta_n = k^*) > \mathbb{P}(\theta_n = k^* - 1)$ and (1.2) tells us that $\mathbb{P}(\theta_n = k^* + 2) < \mathbb{P}(\theta_n = k^* + 1)$. Thus, $\Theta(n)$ has at most two modes. If $\Theta(n-1)$ has two modes $k^*$ and $k^* + 1$. By Lemma 1.2, we have $\mathbb{P}(\theta_n = k^* + 1) = \frac{n-1}{n}\mathbb{P}(\theta_{n-1} = k^* + 1) + \frac{1}{n}\mathbb{P}(\theta_{n-1} = k^*) = \mathbb{P}(\theta_{n-1} = k^*)$ and also $\mathbb{P}(\theta_n = k^*) = \frac{n-1}{n}\mathbb{P}(\theta_{n-1} = k^*) + \frac{1}{n}\mathbb{P}(\theta_{n-1} = k^* - 1) < \mathbb{P}(\theta_{n-1} = k^*)$. These two formulas give $\mathbb{P}(\theta_n = k^* + 1) > \mathbb{P}(\theta_n = k^*)$, yet equation (1.2) gives $\mathbb{P}(\theta_n = k^* + 3) < \mathbb{P}(\theta_n = k^* + 2)$. Hence $\Theta(n)$ has at most two consecutive modes. Furthermore, in both cases, if we note $k_n^*$ and $k_{n-1}^*$ the last modes of $\Theta(n)$ and $\Theta(n-1)$ respectively, we see that $k_n^* \geq k_{n-1}^*$. □

## 1.4 Conclusion

In this chapter, we have presented the main aspect of EVT from its historical context to its mathematical formalism and its plentiful applications. Then, we explained the link between EVT and anomaly detection. Notably, we have elucidated the whys and wherefores of this theory in the context of cybersecurity, and presented the most suitable methods from the EVT literature for anomaly detection in industrial systems. Finally, we introduced Fast Maxima Sampling, an efficient sampling method useful for estimators of the GEV distribution. We established its legitimacy by proving 1) that this procedure generates data points with the same law of probability than the canonical procedure that consists in generating a fixed number of sample points and retaining the one with the maximum value 2) that it is exponentially faster than the canonical procedure. Concomitantly, we presented and proved several properties so that one can make use of Fast Maxima Sampling in an efficient way.

Chapter 2

# Integrity of physical processes from industrial systems

As explained in the introduction of Chapter 1, the first step of many anomaly detection methods is to define an anomaly scoring function, i.e. a function that assigns a score to an input that indicates to which extent the input is normal. It is therefore necessary to agree on the definition of normality. One can borrow the definition of Grubbs [59]:« *An outlying observation, or "outlier", is one that appears to deviate markedly from other members of the sample in which it occurs.*». However, in the context of industrial systems security, one sometimes dreads some types of attack more than other types because of their different costs which varies from one system to another. For example, if an ICS has the capacity to threathen humans life, the detection of types of anomaly that could result in a deadly disaster is obviously prioritized against any other types. Furthermore, the cost-benefit ratio for the attacker is also an important aspect. It is indeed useful to better detect an attack the expert deems very feasible on its system than an attack that is very unlikely considering the potential malicious actors, their interest and their assets. The motivations of hostile actors range from the simple revenge for career disillusion to strategies in the context of international rivalry [66]. Hence, we need an anomaly scoring function flexible enough to incorporate the security expert definition of normality. A second aspect about industrial systems is that their operations tend to have a certain determinism. Indeed, their automata are programmed to follow a specific course of action and each of their devices are used for precise aims, usually only one per device. This is in contrast with other CPSs whose components can perform different tasks [112], for example in home automation [39]. This determinism aspect of ICS working has to be taken into account in solutions for integrity of physical processes from such systems. We will thus explore a model that can tackle both aspects, which is the autoencoder. In this chapter, we first explain the threats for ICSs in terms of data integrity. Secondly, we present an industrial system simulation model we developed for experimentation purposes, and then we present

our new concept of state in industrial systems that machine learning models can use to benefit from their near-deterministisc nature. The model used as the solution for the integrity is then introduced and finally the losses we defined to control the priority on the types of anomaly to detect are explained.

## 2.1 Data integrity threats

ICSs face various threats ranging from insider attacks to denial of service attacks. We focus on the attacks that have consequences on the integrity of the ICS data before the goal of the attack has been reached. For instance, we do not consider physical attacks whereby an adversary gains physical access to the system and damage one of its machine.

Integrity threats in an ICS physical process data can be classified in two classes: 1) data integrity attacks, that try to modify original data while preserving the semantics of the payload of the application layer packets (units of data handled by network communication protocols) 2) injection attacks, that introduce data into a program in order to facilitate the execution or the interpretation of malicious data in an unexpected manner. Data integrity attacks are a specific kind of attack that exploits the data themselves, while injection attacks are a broader class, which, when implemented on an ICS, can also involve the physical process.

We look at data integrity attacks in the form of replay attacks on values of registers (memory location of an automaton)—eavesdropped by the attacker in order to mislead the security supervisor or to hide another attack—or in the form of more severe attacks that direclty tamper with the sensors and actuators. As for injection attacks, we consider stealth attacks that change the code executed by the automata in order to adversely affect the ICS process; for example, to force manufacturing imperfections or to induce abnormal wear and tear on components. Introduced in 2003 [73], stealth attacks minimize the cost to, and visibility of, the attacker. Since Stuxnet in 2010, the risk of stealth attacks on ICSs has become real [29]. This kind of attack produces different anomalies from the more classical attacks such as replay attacks. We also consider a third kind of anomaly which, while unrealistic in practice, is useful for our analysis. Finally, SWaT dataset, which comprises 36 attacks involving physical data, is used for comparison with the literature.

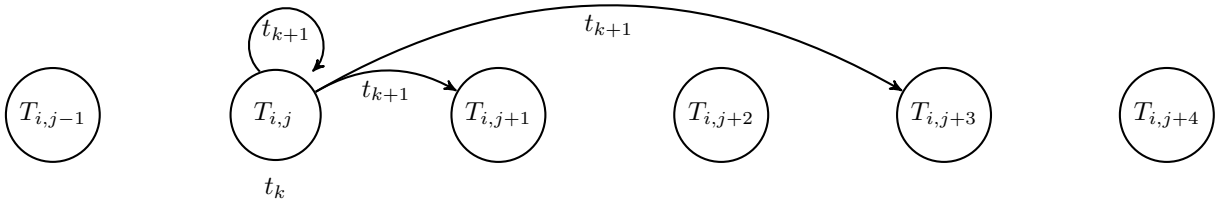



Figure 2.1 – Possible moves for a register $i$ at position $j$ within the template $T$ at time $t_k$. The simulation amounts to move forward the register $i$ in its line $T_i$ within the template $T$ and take the corresponding binary value or not and keep the same binary value $T_{i,j}$.

## 2.2 Simulation of industrial systems

In this section, we introduce Simulation of ICSs with Binary-valued Registers (Sibriz), that can be used to generate data from cyclic industrial systems with binary registers.

As already mentionned in the introduction, ICSs are networks of sensors, actuators and automata. An Automaton like Remote Terminal Unit (RTU), PLC or Distributed Control System (DCS), is a component that is in charge of the control of some parts of the industrial processes. It takes as input data issued from actuators and sensors so as to decide about the next actions to proceed, sending back control signals to these components. An automaton typically uses visual programming language, e.g. Grafcet (Graphe Fonctionnel de Commande des Étapes et Transitions), that specifies its behavior based on its inputs and its hardware configuration. An automaton interacts with sensors and actuator through registers that store their current status values.

Simulation models or ICS simulators that can be found in the literature are most often focused on one or more of the following aspects [105]: the operation of PLCs, the type of infrastructure (smart grid e.g. simulated by Mosaïk [127], manufacturing line, ...), the types of attacks or malfunctions that the ICS may face, the communication protocols used. These simulators mainly model the logical dependencies between the different components of the ICS but not the variability. To the best of our knowledge no work focus on a probabilistically realistic simulation of industrial systems in terms of physical data (data from sensors, actuators and PLC controls, all passing through the ICS registers). Yet, this is necessary to simulate attacks mainly involving physical data when one is interested in data integrity control. This is the purpose of the simulator we propose.

In order to simulate statistical dependencies and produce the register values of a binary

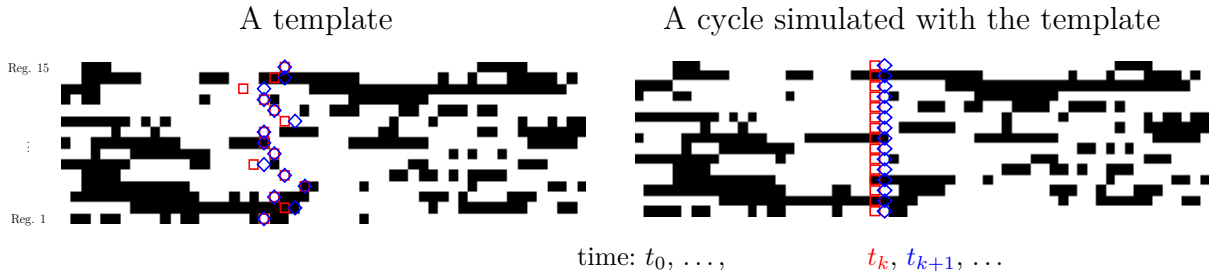



Figure 2.2 – Sibriz is based on a probabilistic progression in a template: the binary value of each register of the simulated cycle at a certain time $t_k$ is the value picked by a random variable that can either stay in the same position in the template as at time $t_{k+1}$ (and thus keep the same value) or move forward by a step possibly greater than one (and pick the corresponding binary value) depending on progressions of other random variables as depicted in Figure 2.1.

ICS at a certain point in a cycle, a Multivariate Markov Chain (MMC) model could be considered a good candidate. Its number of chains $r$ would be the number of registers in the ICS and its the order $n$ implies that the values of the registers at a given time only depends on their $n$ previous values. Thus, it is possible to associate the states of the MMC with the values of the registers (binary in our case) and to model the evolution of the ICS using the MMC transition probability matrix. However, according to the ICS, this transition matrix can be excessively large with a number of parameters of the order of $2^{2\times r\times n}$, for the conventional MMC model, or of $(r \times n)^2$ on the basis of the optimized model presented in [18]. Moreover, it is not easy to choose the right parameters to obtain a model that produces very similar cycles as it is the case for industrial systems. These are the reasons why we decided to follow another strategy with the objective of building an efficient simulator, i.e. with few parameters easy to choose.

To explain our approach, let's first consider the ideal version of an ICS, i.e. an ICS that does not suffer from any variability in its operation: it is deterministic. By observing the values of the registers of this ideal ICS over a duration equal to that of its complete cycle, we can construct a template of the cycle in the form of a binary matrix with as many rows as registers. Let us call these rows, temporal line of registers. In practice, due to its almost deterministic behavior, an ICS will produce cycles that are very close to that of the cycle of its ideal equivalent ICS, that is its template. The idea behind the use of the template is that it provides the behavior of each register in the cycle and that it is possible to simulate the behavior of a register such as its progression on its temporal line by making it speed up, slow down or stop over time, as shown in Figure 2.1. Of course, for the simulated cycle to be close to the template and to reflect a realistic ICS working,

the progression of one register must depend on other registers such that each register has about the same progression within the template as illustrated in Figure 2.2.

We can now describe the principle of Sibriz. When simulating a cycle thanks to a template $T$, the transition from a position $p$ (inducing the value $T_{i,p}$) to the position $p+1$ (inducing $T_{i,p+1}$) is random and depends on the position of other registers. In fact, to be more precise, register $i$ can go from a position $p$ (inducing $T_{i,p}$) to the position $p+q$ (inducing $T_{i,p+q}$) with $q > 0$, i.e. it can stay at the same position if q=0 or "jump in the template" by a step greater than 1, if $q > 1$.

The goal is then to find a way to generate the progression of a register on its temporal line in a probabilistic way and according to the progression of the other registers. Three principles seem to us necessary. They can be expressed by probabilities of progression of a register in the template conditioned by the position of the other registers. Note that these probabilities will not be formally calculated but will only serve as guidelines for defining the random variables of register positions in the template from intermediate random variables aptly dependent on each other.

It is important to remark that Sibriz simulates the cycle step by step in an efficient way, that is at time $t$ it generates the positions of every register in the template based on their previous positions at time $t-1$ and these positions induce the values of the registers. This way, we generate a Multivariate Markov Chain of a potentially high order, the values of the registers, thanks to a Multivariate Markov Chain of order 1, the positions of the registers within the template. However, this Multivariate Markov Chain of order 1 has $k \times r$ states, with $k$ the number of columns in the template (the length of the temporal lines), so it is not directly modeled (i.e. its very sparse transition matrix is not given) but rather generated thanks to a given set of rules detailed hereafter.

Let us note $p_i$ the random variable for the position of $i$ within the template. Let us note $p \xrightarrow{i} q$ the event "the register $i$ moves from position $p$ to position $q$". We can already note that we want $\mathbb{P}(p_i \xrightarrow{i} q | q < p_i) = 0$. Moreover:

1. We want $\mathbb{P}(p \xrightarrow{i} p | (p_j)_{j \neq i})$ to be lower the more $\frac{1}{r}\sum_{j=1, j\neq i}^{r} p_j - p$ is large. This simple rule already allows to obtain quite similar cycles by enforcing a register not to have a big delay on its temporal line. However, it is not sufficient to fully take into account the dependencies between registers.
2. In addition, registers can be linked by statistical relationships during the cycle of a ICS, expressing variability in the operation of the ICS. In other words, two registers $i$

and $j$ can be linked by a statistical relationship implying that one tends to accelerate or slow down the progress of the other in the template:

a) We say that $j$ tends to accelerate $i$ at position $p_j$ if, with $p$ the position of $i$ and $p_j$ that of $j$, $\forall q > 0, \mathbb{P}(p \xrightarrow{i} p+q | p_j > p) > \mathbb{P}(p \xrightarrow{i} p+q)$.

b) We say that $j$ tends to decelerate $i$ at position $p_j$ if, with $p$ the position of $i$ and $p_j$ that of $j$, $\forall q > 0, \mathbb{P}(p \xrightarrow{i} p+q | p_j < p) < \mathbb{P}(p \xrightarrow{i} p+q)$.

3. Registers can be linked by different logical relationships during the ICS cycle. This can be summarized as the relationship between two registers $i$ and $j$, one preventing, or forcing, the other to advance in the template:

a) The case where $i$ is blocked at a certain position $q$ until $j$, at position $p_j$, has reached $p$ can be expressed as: $\mathbb{P}(p_i \xrightarrow{i} p_i | p_i = q, p_j < q) = 1$.

b) The reverse, i.e. at position $q$, $j$ forces $i$ to move forward is described by $\forall a < q, \mathbb{P}(p_i \xrightarrow{i} a | p_j = q, \neg(p_j \xrightarrow{j} p_j)) = 0$.

This allows to model system actions that cannot start before other actions are completed or started, as in our example in Figure 1, where the robotic arm cannot pick up the object until it reaches the end of the conveyor belt.

In its principle, to simulate an ICS, Sibriz therefore relies on the definition of a template and random variables representing the positions of the registers in this template and whose distribution verifies the three cited properties.

As explained previously, we consider a template $T$ which is a matrix $r$ times $l$, with $r$ the number of registers and $k$ the size of the rows (temporal lines of the registers), i.e. the duration of the ideal and deterministic cycle that the template represents. We define for each register a variable $r_i^t$ which represents the position of the register in the template at time $t$: the value of the register $i$ at time $t$ is then $T_{t,r_i^t}$ (see Figure 2.2). All the registers start from the beginning of the template: $\forall i, r_i^0 = 1$. Let us start with the simple case where a register can either stay at the same position $p$, $p \xrightarrow{i} p$, or move forward by one step, $p \xrightarrow{i} p+1$.

Since we are in the simple case where only $p \xrightarrow{i} p$ or $p \xrightarrow{i} p+1$ are possible, we can note: $r_i^t = r_i^{t-1} + \mathbb{1}_{U_t < \mathbb{P}(r_i^{t-1} \xrightarrow{i} r_i^{t-1}+1 | (r_k)_k)} = r_i^{t-1} + B_i^t$, with $U_t$ a uniform random variable between 0 and 1, and $\mathbb{1}_{\cdot}$ the indicator function.

The problem here is that it is difficult to determine exactly $\mathbb{P}(r_i^{t-1} \xrightarrow{i} r_i^{t-1}+1 \mid (r_k)_k)$ so as to check the interdependencies between registers specified by 1, 3 and 2. We will

therefore proceed differently: we will generate, for each register, intermediate random variables, detailed below, and then use these intermediate random variables together to construct the random variables $(B_i^t)_i$ which, by construction, will be dependent on each other and will check 1, 3 and 2. The first intermediate variable is the so-called lag, $lag_i^t$, which represents for the register $i$ its lag relative to the other registers at time $t$. The greater the lag, $lag_i^t$, the greater $\mathbb{P}(r_i^{t-1} \xrightarrow{i} r_i^{t-1} + 1 | lag_i^t)$ must be. This allows to check the first property.

$$lag_i^t = \sum_{\substack{j=1,..,r \\ j \neq i}} F_1((r_k^{t-1})_k)^{r_i^{t-1} - r_j^{t-1}} / (r-1)$$

This term depends on another term called $F_1((r_k^{t-1})_k)$ which allows to give more or less importance to the delays of $i$ with respect to $j$, $r_i^{t-1} - r_j^{t-1}$, depending on the positions $(r_k^t)_k$ of all the registers in the template. This function $F_1$, detailed later on, must be positive and is a parameter in itself of our model and can be constant, meaning that the delay has as much importance at some point in the template.

The second intermediate variable is $fw_i^t = \mathbb{1}_{U_i^t < F_2(lag_i^t, (r_k^t)_k)}$ with $U_i^t$ a random variable following the uniform distribution on $[0, 1]$. This term depends on $F_2(lag_i^t, (r_k^{t-1})_k))$, where $F_2$ is an increasing function with respect to the first argument ($\forall u, x \to F_2(x, u)$ is increasing) bounded by 0 and 1. Thus, the greater the lag, the greater the probability that $fw_i^t$ equals 1. The fist property is verified.

We can then introduce the main parameters allowing to check the second desired property. These are two matrices $A$ and $B$ of sizes $r$ times $r$. Matrix $A$ will be used to link two registers so that one drags the other forward, and $B$ backward, that is one register slows down the other.

We define $PF_{ij}^t = A_{ij}.\mathbb{1}_{fw_j^t=1}/(A_{ij} + F_3((r_k^{t-1})_k, i, j))$. Similarly, for the backward relation, we define $PB_{ij}^t = B_{ij}.\mathbb{1}_{fw_j^t=0}/(B_{ij} + F_3((r_k^{t-1})_k, j, i))$. Note that we invert $i$ and $j$ in $F_3$ so as to have the opposite effect compared to the one in $PF_{ij}^t$.

$F_3$ is a function, detailed later, determining the influence of $j$ on $i$ according to the position of the registers in the template and such that $F_3(., i, j)$ is increasing with $r_i^{t-1} - r_j^{t-1}$. Before introducing logical dependancies, we formulate the progression of $i$ at time $t$ this way:

$$r_i^t = r_i^{t-1} + rs_i^t$$

The term $rs_i^t$ is determined by $pf_{ij}^t = \mathbb{1}_{o^t.F_j^t < PF_{ij}^t}$ and $pb_{ij}^t = \mathbb{1}_{(r-o^t).G_j^t < PB_{ij}^t}$ with $F_j^t$ and $G_j^t$ two random variables of uniform law $\mathcal{U}(0,1)$ and $o^t = \sum_{i=1}^r fw_i^t$:
$rs_i^t = \mathbb{1}_{\sum_j pf_{ij}^t > \sum_j pb_{ij}^t} + \mathbb{1}_{\sum_j pf_{ij}^t = \sum_j pb_{ij}^t}.fw_i^t$

The terms $o^t$ and $r - o^t$ allow us to rebalance the influence of each register to be able to compare the two sums $\sum_j pf_{ij}^t$ and $\sum_j pb_{ij}^t$. In other words, $i$ advances by one step ($rs_i^t = 1$) if the positive influence (matrix A) of some registers is sufficient ($\sum_j pf_{ij}^t > \sum_i pb_{ij}^t$) or if it has the negative influence (matrix B) of other registers ($\sum_j pf_{ij}^t = \sum_j pb_{ij}^t$) and $fw_i^t$ (determined by the lag) is 1. If, on the contrary, the registers drive $i$ backwards ($\sum_j pf_{ij}^t < \sum_j pb_{ij}^t$), then even if $fw_i^t = 1$, we have $rs_i^t = 0$.

Now let us move on to the more complicated case where $p \xrightarrow{i} p + q$ with $q > 1$ is possible. We will define new intermediate variables such that 1 and 2 already verified in the simple case will remain true and such that dependencies of type 3 will also be verified.

The idea is to add a term, noted $\sigma_i^t$, to the formula defining $rs_i^t$. This term represents a jump in the template for register $i$:

$$rs_i^t = \mathbb{1}_{\sum_j pf_{ij}^t > \sum_j pb_{ij}^t} + \mathbb{1}_{\sum_j pf_{ij}^t = \sum_j pb_{ij}^t}.fw_i^t + \sigma_i^t$$

This jump is determined by $L$ a list of $2^r$ pairs of a number and a subset of the set of $r$ registers, called subjection list. For an integer $c$ between 1 and $2^r$, $L_c = (d_c, e_c)$ where $L_{c,0} = d_c$ is a real between 0 and 1 and $L_{c,1} = e_c$ is the set of registers integers between 1 and $r$ for registers affected by the same phenomenon, represented by a jump in the template, unless it goes against a dependency between registers of the type defined in 3. This way, $\sigma_i^t = \min(a_i^t, b_i^t)$ with $a_i^t$ the sum of jumping phenomena within the template:

$$a_i^t = \sum_{0 < c \leq 2^r} \sum_{n \geq 1} \mathbb{1}_{H_c^t < {L_{c,0}}^n \wedge L_{c,1} \ni i}$$

with $H_c^t$ a uniform random variable between 0 and 1.

Since $\sigma_i^t$ is the minimum between $a_i^t$ and $b_i^t$ and since this second term $b_i^t$, explained in more detail later, is the maximum jump that register $i$ can make without going against a type 3 dependency minus 1, we are sure that making $i$ move by $1 + \sigma_i^t$ or $\sigma_i^t$ will respect these dependencies.

Finally, the dependencies of type 3 will be modeled by a matrix $W$, called retention matrix, of size $k$ times $r$, whose elements are lists of pairs composed of an integer rep-

resenting a register and a tag ('hard' or 'soft'). This matrix will be used to define a last intermediate binary variable (0 or 1), $\tilde{w}_i^t$, allowing to cancel, if necessary, the effect of $rs_i^t$ in order to respect a dependency of type 3, if $\tilde{w}_i^t = 0$ or not if $\tilde{w}_i^t = 1$. That is to say: $r_i^t = r_i^{t-1} + \tilde{w}_i^t.rs_i^t$. This matrix $W$ is of size $k$ times $r$ because for a register $i \in [\![1, r]\!]$ and its position in the template $r_i^{t-1} \in [\![1, k]\!]$, we will consider the list $W_{r_i^{t-1},i}$ which links the register $i$ to the registers of this list with the tag 'hard' or 'soft' at the position $r_i^{t-1}$ within the template.

We define the two following boolean values: $\mathrm{TAG}(i,j,t) = ((j, \mathrm{TAG}) \in W_{r_i^{t-1},i})$ and $\mathrm{TAG}^*(i,j,t) = (\mathrm{TAG}(i,j,t) \wedge r_j^{t-1} < r_i^{t-1})$, with the term TAG being either 'soft' or 'hard'. We say that $j$ is linked to $i$ by the tag TAG at time $t$ if $\mathrm{TAG}(i,j,t)$.

Here are the two desired properties of tags:

i) For both tags, TAG='soft' or TAG='hard':
$(\mathrm{TAG}^*(i,j,t) \vee (\mathrm{TAG}(i,j,t) \wedge r_j^{t-1} = r_i^{t-1} \wedge \exists(j', \mathrm{tag}), \mathrm{tag}^*(j,j',t))) \Rightarrow r_i^t = r_i^{t-1}$ i.e. when $j$ is linked to $i$ at time $t$, $i$ cannot advance if it is ahead of $j$ or if $i$ and $j$ are at the same position and $j$ is ahead of a third register linked to it

ii) $(\mathrm{hard}(i,j,t) \wedge r_i^t > r_i^{t-1}) \Rightarrow (r_j^t > r_i^{t-1} \vee \exists(j', \mathrm{tag}), (\mathrm{tag}(j,j',t) \wedge \neg\mathrm{tag}(i,j',t)))$ i.e. when $j$ is linked to $i$ by the tag 'hard' at time $t$, register $i$ cannot move if $j$ does not pass beyond the point $r_i^{t-1}$ unless there is a register that is linked to $j$ but not to $i$.

The only difference between tags 'soft' and 'hard' is that for the tag 'hard', $i$ must only pass a retention point after the register $j$ linked to it or at the same time when conditions are met, while for the tag 'soft', $i$ can pass the retention point before $j$. Incidentally, if one has both $(j, \mathrm{hard}) \in W_{p,i}$ and $(i, \mathrm{hard}) \in W_{p,j}$, that is $p$ is a hard retention point both for $i$ with respect to $j$ and for $j$ with respect to $i$, and if $W_{p,i} \setminus \{j\} = W_{p,j} \setminus \{i\}$, then $i$ and $j$ must pass $p$ exactly at the same time. To take into account the rules of the tags 'soft' and 'hard' described in i) and ii) so as to verify dependencies of type 3, we define four intermediate variables, $\tilde{rs}_i^t, w_{i,\mathrm{soft}}^t$, and $w_{i,\mathrm{hard}}^t$, and finally $\tilde{w}_i^t$, so that the rule to simulate the next positions will reduce to $r_i^t = r_i^{t-1} + \tilde{w}_i^t.rs_i^t$. With the convention $\prod_\emptyset = 1$:

$$w_{i,\mathrm{soft}}^t = \prod_{(j,.) \in W_{r_i^{t-1},i}} \mathbb{1}_{r_i^{t-1} \leq r_j^{t-1}}$$

$$\tilde{rs}_i^t = \mathbb{1}_{rs_i^t \geq 1}. \prod_{(j,.) \in W_{r_i^{t-1},i}} \mathbb{1}_{rs_j^t \geq 1}$$

$$w^t_{i,\text{hard}} = w^t_{i,\text{soft}} \cdot \prod_{(j,.)\in W_{r_i^{t-1},i}} \mathbb{1}_{r_i^{t-1} < r_j^{t-1} \vee \tilde{rs}^t_j = 1}$$

$$\tilde{w}^t_i = w^t_{i,\text{hard}} \cdot \prod_{(j,\text{tag})\in W_{r_i^{t-1},i}} \mathbb{1}_{w^t_{j,\text{tag}} = 1 \vee r_i^{t-1} < r_j^{t-1}}$$

We can now explain $b^t_i$ which is simply the number $m-1$ with $m$ such that $W_{r_i^{t-1}+m,i} \neq \emptyset$ and $\forall 0 < m' < m, W_{r_i^{t-1}+m',i} = \emptyset$. In other words, the term $b^t_i$ of $\sigma^t_i = \min(a^t_i, b^t_i)$ is the maximum jump that register $i$ can make without going against a dependency determined by $W$. This can be formalized by a simple sum :

$$b^t_i = \sum_{n=1}^{k-r_i^{t-1}} \mathbb{1}_{\substack{\forall 1\leq m\leq n \\ W_{r_i^{t-1}+m,i} = \emptyset}}$$

To conclude the definition of our Sibriz model, we must define a last set of parameters, $(S_t)_{0<t\leq k}$, that are $k$ positive numbers that will be used to define $F_1$ to determine the importance of the delay of a register in relation to the others at the $k$ positions of the template, $F_2$ used to formulate $fw^t_i$ – both used to verify dependencies of type 1 – and $F_3$ to determine the importance of the delay of a register in the definition of $PF^t_{i,j}$ and $PB^t_{i,j}$, used to verify dependencies of type 2.

With the previous notations, Sibriz$(T, A, B, W, L, (S_t)_{0<t\leq k}, r^{t-1})$ is defined as the vector $(T_{i,\min(r^t_i,k)})_{0<i\leq r}$ with $r^t_i$ the position of the register $i$ in the template $T$ at time $t > 1$ and $r^t_i = r^{t-1}_i + \tilde{w}^t_i.rs^t_i, \forall 0 < i \leq r, 0 < c \leq 2^r$.

Let us now detail the functions, $F_1$, $F_2$ and $F_3$ in order to fully define the algorithm. We first define $\tilde{S}_t = S_{\max_i(r_i^{t-1})}$. $F_1$ is then defined as follows: $F_1((r_k^{t-1})_k) = f(\tilde{S}_t)$, with $f$ defined in (2.1), useful to get an idea of the average length of the simulated cycles when, $A$ and $B$ are null matrices, $L$ is a null vector, and $W$ is an empty set list, i.e. when no parameters governing the dependencies between registers have been defined. Then, as an increasing function with respect to the first argument, bounded by 0 and 1 for $F_2$ we take: $F_2(lag, (r_k^{t-1})_k) = 1/(1 + 1/(\tilde{S}_t \times lag))$. Remember that $(S_t)_t$ are positive, so $F_2$ is increasing compared to the first argument. Then, $F_3$, which determines the influence of $j$ on $i$ according to the positions of the registers in the template, is defined as follows: $F_3((r^t_k)_k, i, j) = f(S_{min(r^t_i,k)})^{(r_i^{t-1} - r_j^{t-1})}$, with $f$ the same function as the one used for $F_1$.

Now that we have explained these equations, let us assemble them in a compact way.

**Definition of Sibriz**

With previous notations, let $\forall 0 < i \leq r, r_i^1 = 1$, given $r^{t-1}$ the positions at discrete time $t-1$ of the registers in $T$, the simulation at time $t$, $\text{Sibriz}(T, A, B, W, L, (S_t)_{0<t\leq k}, r^{t-1})$, is defined as a vector $(T_{i,\min(r_i^t,k)})_{0<i\leq r}$ with $r_i^t$ the position of register $i$ in template $T$ at time $t > 1$ computed as follows: $\forall (i,j) \in [\![1,r]\!]^2$,

$c \in ]0, 2^r], \tilde{S}_t = S_{\max_i(r_i^{t-1})}$

$(U_i^t, F_i^t, G_i^t, H_c^t) \overset{i.i.d.}{\sim} \mathcal{U}(0,1)$

$lag_i^t = \sum_{j\neq i} f(\tilde{S}_t)^{r_j^{t-1}-r_i^{t-1}}/(r-1)$

with $f$ a function such that its image $\text{Im}(f) \subset \mathbb{R}_+$ (typically (2.1)) [1]

$FW_i^t = \frac{1}{1+(\tilde{S}_t.lag_i^t)^{-1}}; \quad fw_i^t = \mathbb{1}_{U_i^t < FW_i^t}; \quad o^t = \sum_{i=1}^r fw_i^t$

$PF_{ij}^t = \frac{A_{ij}.\mathbb{1}_{fw_j^t=1}}{A_{ij}+f(\tilde{S}_t)^{r_i^{t-1}-r_j^{t-1}}}; \quad PB_{ij}^t = \frac{B_{ij}.\mathbb{1}_{fw_j^t=0}}{B_{ij}+f(\tilde{S}_t)^{r_j^{t-1}-r_i^{t-1}}}$

$pf_{ij}^t = \mathbb{1}_{o^t \times F_j^t < PF_{ij}^t}; \quad pb_{ij}^t = \mathbb{1}_{(r-o^t)\times G_j^t < PB_{ij}^t}$

$\sigma_i^t = \min(\sum_{n\geq 1}\sum_{0<c\leq 2^r} \mathbb{1}_{H_c^t < L_{c,0}{}^n \wedge L_{c,1} \ni i}, \sum_{n=1}^{k-r_i^{t-1}} \mathbb{1}_{\substack{\forall 1\leq m\leq n \\ W_{r_i^{t-1}+m,i}=\emptyset}})$

$rs_i^t = \mathbb{1}_{\sum_j pf_{ij}^t > \sum_j pb_{ij}^t} + \mathbb{1}_{\sum_j pf_{ij}^t = \sum_j pb_{ij}^t}.fw_i^t + \sigma_i^t$

$\tilde{rs}_i^t = \mathbb{1}_{rs_i^t \geq 1}.\prod_{(j,.)\in W_{r_i^{t-1},i}} \mathbb{1}_{rs_j^t \geq 1},$

$w_{i,\text{soft}}^t = \prod_{(j,.)\in W_{r_i^{t-1},i}} \mathbb{1}_{r_i^{t-1} \leq r_j^{t-1}},$

$w_{i,\text{hard}}^t = w_{i,\text{soft}}^t.\prod_{(j,.)\in W_{r_i^{t-1},i}} \mathbb{1}_{r_i^{t-1} < r_j^{t-1} \vee \tilde{rs}_j^t = 1}$

$\tilde{w}_i^t = w_{i,\text{hard}}^t.\prod_{(j,\text{tag})\in W_{r_i^{t-1},i}} \mathbb{1}_{w_{j,\text{tag}}^t = 1 \vee r_i^{t-1} < r_j^{t-1}}$

$r_i^t = r_i^{t-1} + \tilde{w}_i^t.rs_i^t.$

In order to have realistic cycles in terms of what an industrial system could produce, we generate a template thanks to $r$ random walks $((rw_i^t)_{0<t\leq\sigma k})_{0<i\leq l}$ of length $k$, of integers between 0 and 100, and we pick the following binary values $\mathbb{1}_{rw_i^t > a_i}$ every $\sigma$ steps, $T_{i,t} = (rw_i^{\sigma\times t} > a_i)$, with $a_i$ an integer between 0 and 100. Parameters $a_i$ represent the proportion of 1 and 0 produced by register $i$, and $\sigma$ the system activity intensity, since for high value of $\sigma$ the value of $rw_i^{\sigma\times t} > a_i$ will tend to change often. Then the parameters in $W$ are chosen randomly among strategic choices in term of positions in the template

1. The function $f : \mathbb{R} \to \mathbb{R}^+$ (cf. (2.1)) approximates $g : \mathbb{R}\times\mathbb{N}\times\mathbb{N} \to \mathbb{R}^+$ the function such that if $A = 0$ and $B = 0$, and $W = (\emptyset)_{0<u\leq r, 0<v\leq k}$ and $\forall 0 < c \leq 2^r, L_{c,0} = 0$, (and so $\forall i, t,\ \tilde{w}_i^t.rs_i^t = fw_i^t$) and $\forall t, S_t = c$ then $\mathbb{E}(\max_{t>0,\min_i(r_i^t)=k}(t)) = k.(1+\frac{1}{c})$, when $g$ replaces $f$ in the simulation model. We do not know the function $g$ but it turns out that it is nearly constant with respect to its last two variables $k$ and $r$ that are the width and length of the template $T$. Moreover, $g$ is regular enough with respect to its first variable so that we have been able to empirically approximate it by a function of class $C^1$ that equals 2nd and 3rd degree polynomial functions respectively on $[0.75, 1.5]$ and $]0, 0.75]$, and $x \to x^{1/\log(1+x)}$ on $[1.5, +\infty[$ (cf. (2.1)). $\mathcal{U}(0,1)$ is the uniform distribution on $[0,1]$ and $\mathbb{1}$ is the indicator function.

and registers. Indeed, $W$ represents positions in the template where registers have to wait for other registers to be able to move forward, hence a certain determinism. In our experiments, $\forall i, a_i = 0.55$ and $\sigma = 10$.

One can simply use the identity instead of $f$ but it is useful to get an idea of the cycles length, with no other parameter for a starting point. Let us note that whereas $r^t$ is a Multivariate Markov Chain of order 1, Sibriz allows simulating Markov Chain $(T_{i,r_i^t})_i$ of a much higher order. Here is the function $f$:

$$\begin{aligned} f(s) = & \mathbb{1}_{s \geq 1.5} \times s^{\frac{1}{\log(1+s)}} \\ & + \mathbb{1}_{0.75 \leq s \leq 1.5} \times (s \times a + c + \frac{(1.5-s)^2}{2}) \\ & + \mathbb{1}_{0 < s \leq 0.75} \times (s \times d + e - 0.75 \times d - \frac{(0.75-s)^3}{10}) \end{aligned} \tag{2.1}$$

$a = h'(1.5)$; $b = h(1.5)$; $c = b - 1.5 \times a$; $d = a - (1.5 - 0.75)$
$e = (0.75 \times a + b) + (1.5 - 0.75)^2 \times 0.5$ , with $h : x \to x^{\frac{1}{\log(1+x)}}$.

To validate our simulation model, we prove that it can produce two attacks, one brutal and another more subtle, by plotting Receiver Operating Characteristics (ROC) curves from the reconstruction error of autoencoders that turn out to be indeed of different magnitudes. As for attacks simulated with Sibriz, they can be built as follows: i) a Sibriz simulation with the same template $T$, the same delay matrices $A$ and $B$, but a different retention matrix $W''$; ii) a Sibriz simulation with the same $T$ and $W$ but different $A'$ or $B'$ or a different subjection list $L'$; iii) a Sibriz simulation with a spoiled version $T'$ of $T$; iv) a Sibriz simulation with the same $T$ but different $A'$, $B'$ or $L'$, and a different $W'$. These different simulations represent different kinds of attack: i) the first simulation represents the action of an attacker who took control of the system and changed more or less brutally the order of the system's operations; ii) the second simulation represents the action of an attacker who took control of the system remotely or directly and tamper with the system or changed its hardware, or its software configuration while keeping the same order of operation; iii) the third simulation represents the action of an attacker who wants to suddenly disrupt the system in order to damage it or to hurt people around; iv) the fourth simulation represents the action of an attacker who carries out a Man-In-The-Middle attack between the ICS and the monitoring station and mimics the system, e.g. thanks to a model trained on the system's cycle dataset.

## 2.3 A novel notion of state in industrial systems

Our solution relies on the same principle that digital forensics methods that detect an image forgery thanks to the characteristical noise pattern from the camera sensors [102], we want to characterize the slight variations of physical data readings to ensure data integrity. In our context, the main difference is that the system at issue does not execute only one action, like a camera whose only purpose is to take photos, but a variety of actions. To achieve ICS data integrity, we propose to take advantage of their near-deterministic working by introducing the concept of ICS state. This near-deterministic behavior, already mentionned in Section 2.2, appears in actions of an ICS, the latter being repeated with very little variations throughout the ICS working, which is reflected in the register values. Therefore, to characterize the intrinsic variability of an ICS and its near-deterministic working, we can consider its high-level actions. A high-level action is the result of the working of an ICS at the human level, like drilling an object or the manipulation of an object by a robotic arm, in other words what the system exists for. Our goal is thus to verify that data linked to high-level actions of the system have not been altered. On the contrary, actions, like sending a command to an actuator or converting an analog signal from a sensor to a digital signal, are referred to as low-level actions. In the following, an action will implicitly refer to a high-level action. Let us consider the simplest possible ICS operation: the one of what we call cyclical action system, that is a system for which roughly [2] the same sequence of actions appears at each cycle. An example of such a system is the testbed SWaT detailed later on. Let us focus on the registers values of a cyclical action system recorded every second, for instance. An action is linked to these values on time range. For instance, Figure 1 depicts a time window starting and ending with the beginning and the end of the action of the robotic arm transporting an object from one conveyor belt to another. This time window is thus related to this action, but also potentially other actions, happening in this time range, represented by the rest of the registers that do not appear in the scheme and whose values are grayed out in the time series. The length of a cycle refers to the number of registers data loggings from the start of the cycle until its end. Because of variations, the value of a register at certain instant in the cycle is not necessarilty the same at the same instant in another cycle, and it depends on other registers values. Therefore, the cycle has itself a varying length,

2. Because of the aforesaid variability, the exact order of actions might change a bit from a cycle to another for actions that usually start at approximately the same time, but modulo this changes the sequence of actions is the same.

denoted $L$. However, one can expect that time windows of registers values positioned at the same moment in the cycles relatively to their lengths will be very similar. Indeed, these time windows are linked to the same set of actions of a cyclical action system. To benefit from this near-deterministic working of ICSs, on can consider a set of the time windows linked to the same set of actions, which thus represents a certain state of the system. Data integrity is the fact to be able to distinguish normal representations of a set of actions to abnormal ones. Of course, every ICS is not a cyclical action system. For example, a manufacturing system can produce similar cars but with different features that require different actions, thus this is not a cyclical action system. However, one can generalize the concept of system state to such systems. To this purpose, let us have a formal definition of a system near-deterministic working that can be described this way: for a set of high-level actions of the system, there is a set of more or less likely time windows of registers values, whereas actions of the system are deterministic. In order to be as general as possible, let us define the near-deterministic working of a system for CPS, since ICS is just a special case of CPS.

**Definition 2.1.** *A Cyber-Physical System (CPS) $S$ is a set of elements, interacting with each other to perform high-level* ***actions in the real-world****, that consumes* ***inputs*** *and produces* ***outputs****, described by a function, with $\mathscr{P}(E)$ the power set of $E$:*
$f : \{\text{time windows of inputs and outputs of } S\} \to \mathscr{P}(\{\text{actions of } S\})$.

In the case of an ICS $S$, the descriptive function's domain is the set of time windows of S's registers values, that are substrings of the time series of the register values.

**Definition 2.2.**
1. *A CPS working with a describing function $f$ is said to be deterministic if $f$ is bijective. Otherwise, it is said to be* ***near-deterministic****.*
2. *A* ***state*** *of a CPS is the inverse image of a singleton under its describing function $f$, $f^{-1}(\{e\})$—the element $e$ is a set of system actions from $Im(f)$.*

As defined in Definition 2.2, a near-deterministic system state is a set of time windows of variable length containing values of registers linked to a set of actions of the system. One can define a portion of an action as an action per se so as to deal with time windows of similar lengths. In Figure 1, for example, one can consider the carriage of the object by the robotic arm half-way between the two conveyor belts as an action.

Extracting the time windows corresponding to the different states requires a precise knowledge of the functioning of the system, and thus requires a significant effort to adapt

the solution to a certain system. Moreover, using exact states at time of inference can delay the detection of attacks because one will only be able to evaluate a state when the latter is fully completed, whereas some evidence of an attack can be detected earlier. We therefore present in Section 2.5 a method to deal only with sets of fixed length time windows as states approximations for cyclical systems.

There are numerous states and, since two states can be linked to almost the same set of actions, one must choose which states are to be under scrutiny, now denoted $\{S_n\}_{1\leq n\leq m}$, with $m$ the number of states of interest. In the case of a cyclical action system, with $R_{ij}$ the value of the register $i$ at time $j$ within the cycle $(R_{ij})_{1\leq i\leq k, 1\leq j\leq L}$ of variable length $L$, with $l$ the length of the smallest cycle and for the $n$-th state $S_n$, there exists a random variable $R^n$ such that $\mathrm{Im}(R^n) = S_n$ and $R^n = ((R_{ij})_{1\leq i\leq k})_{\lfloor p_n\times L/l\rfloor \leq j\leq \lfloor p_n\times L/l+s_n\rfloor}$, with $p_n$, $s_n$ and $L$ random variables to define position $p_n \times L/l$—relative to the length of the cycle $L$—of a time window of size $s_n$ in a cycle, and $k$ the number of registers. This way, the anomaly score is $Err(R^n, \hat{R}^n) = \sum_{i,j}(R^n_{ij} - \hat{R}^n_{ij})^2/(k \times s_n)$. When one does not have precise knowledge about the system, one can instead consider time windows of the same size and fixed relative positions, as depicted in Figure 2.3. More specifically, $\tilde{R}^n = ((R_{ij})_{1\leq i\leq k})_{\lfloor p_n\times L/l\rfloor \leq j\leq \lfloor p_n\times L/l+s\rfloor}$, with $p_n$ and $s$ fixed integers, such that, for instance, $\exists c/\forall 1 < k \leq m, p_k - p_{k-1} = c$, and the anomaly score is defined based on $\tilde{R}^n$ instead of $R^n$. Positions $(p_n)_n$ may not be evenly spaced if this does not meet a constraint detailed in the next paragraph. In the case of non-cyclical system where the sequence of actions change over cycles, one cannot approximate states by positioning time windows relatively a cycle length. Indeed, since an action can be added or canceled from one cycle to another, the time lag between to actions can vary much more than if only the intrinsic variation of the system came into play, i.e. in the case of a cyclical action system. This way, the position in a cycle, of a time window, proportional to its length, does not necesarily match the same set of actions in two different cycles. In this case, knowledge on the system is needed to define states or their approximations.

Once the states of interest defined, one has two possibilities: either build a model that considers states independently from each other or a model that considers sequences of states. The first option is the simplest one and will be our choice. However, there is a constraint on states of interest for a model that considers states independently from each other. This constraint is that states of interest $(S_n)_n$ have to occur only once within a cycle. Imagine that the ICS executes a same set of actions at two different moments, 1 and

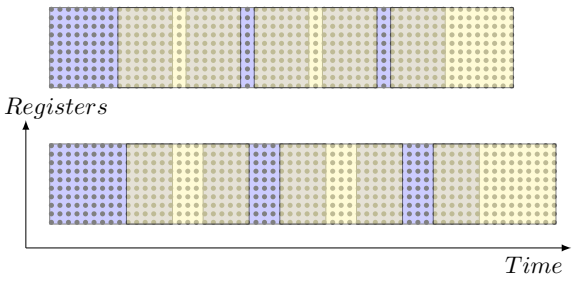



Figure 2.3 – A cycle is represented by a matrix $M$ of which $M_{ij}$ is the value of the register $i$ at the $j$-th position in the cycle. The variability of a cyclic near-deterministic system can be seen, among other things, in the length of its cycle. So the states, 6 in number in this example, are defined thanks to the relative positions of the time windows in the cycles.

2, in the cycle, then the corresponding state $S_*$ appears twice in the cycle. Imagine that there are some differences in the execution of its set of actions depending on whether it is moment 1 or at moment 2, so that only some time windows $S_{*1} \subset S_*$ appear at moment 1 and some time windows $S_{*2} \subset S_*$ appear at moment 2. Let us notice that $S_{*1} \cup S_{*2} = S_*$. Thus, if one of the states of interest $(S_n)_n$ is $S_*$, it can lead to a low anomaly score if a time window $w \in S_*$ appears at moment 1, for instance, even if $w \in S_{*2} \setminus S_{*1}$ which is an abnormal situation. To avoid this pitfall, one has to add context to $S_{*1}$ and $S_{*2}$, meaning that one has to consider two states of interest $S_{\#1}$ and $S_{\#2}$, instead of $S_*$, within a longer time range, i.e. such that for $i = 1, 2$, $\forall v \in S_{*i}, (\exists w \in S_{\#i}/v \sqsubset w) \wedge (\nexists w \in S_{\#i}/w \sqsubset v)$, with $v \sqsubset w$ meaning that $v$ is a substring of $w$. Moreover, for the model not to confuse time windows of $S_{\#1}$ and time windows of $S_{\#2}$ they have to be given enough context, that is their respective sets of actions have to be different enough. In our use case, such issue does not appear, so we just define positions $(p_n)_n$ of fixed length time windows such that $\exists c / \forall 1 < k \leq m, p_k - p_{k-1} = c$.

## 2.4 Setup of the experiment

**Datasets.** The dataset from the testbed SWaT for experiments on real data is detailed in the appendix A. As for evaluation on simulated data, datasets built consist of couples of sets of the same size, the first corresponding to cycles of an industrial system under normal conditions and the second comprises either attacks simulated with Sibriz, replay attacks or an artificial anomaly that we do not consider as an attack but only as a tool for our argumentation. This third anomaly is obtained by adding the zero vector in normal

time windows between two successive time steps and removing the last vector of the time windows. Let us call it a global on-off anomaly. We do not typecast this transformation as an attack as it would correspond to a global on-off attack at times when the system can resume as if it has never stopped, which is unlikely on large systems, yet interesting for the analysis. In brief, in this chapter, we consider attacks of type i) for long-term anomalies, global on-off anomalies, for punctual anomalies, and replay attacks and attacks on the SWaT dataset.

Replay attacks consists in replacing a part of the time series, captured by the registers, by a part of a past cycle of the same system matching the same operations. Depending on the system, this kind of online attack is more or less difficult, resulting in imprecision regarding the timing of the attack. In cyclic system, we model this imprecision by shifting the part of the time series replayed by a step between 0 and 4 in the genuine time series in regard to its relevant position. Indeed, the disruption is all the more important that the replayed window is shifted within the genuine cycle.

Attacks of type i) are simulated by Sibriz with the same parameters than for the normal regime apart from the retention matrices some retention points of which are removed or, for the ones with «hard» tags, changed to «soft». Retention matrix $W_a$ of the brutal attacks holds about 3/4 of the retention points of the normal regime's retention matrix $W_n$ and about 3/4 of these retention points have the same tags than in the normal regime, so about 1/2 of the parameters of $W_n$ are kept unchanged in $W_a$. For the tempered attack about 85% of retention points of $W_n$ are kept in $W_a$ and about 91% of their tags are the same, so $W_a$ is about 78% similar to $W_n$.

The training and validation sets are respectively composed of 4000 (except for the one the NNs of the first row in Figure 2.6, which contains 3000 cycles) and 1000 simulated cycles of the normal regime of the system with an average cycle length around 53. Another dataset is built with 1000 cycles from the normal regime and for each ROC curve estimation, 1000 abnormal cycles are simulated as previously described so that they have also an average length of 53, otherwise a simple statistical hypothesis test on the cycle length would tell us that the cycles distribution is abnormal.

**Neural Networks.** The NNs used are 1D-CNN autoencoders taking time windows of length $x$ as inputs on a cycle of 15 binary registers, with $x = 15, 23, 30$. NNs of type i) $(x)$ have this topology $[(x, 15)$, $(\lceil \frac{1}{2}x \rceil, 9)$, $(\lceil \frac{1}{2}x \rceil, 6)$, $(\lceil \frac{1}{2}x \rceil, 9)$, $(x, 15)]$ and the type ii) $(x)$, $[(x, 15)$, $(\lceil \frac{1}{2}x \rceil, 9)$, 45, $\lceil \frac{1}{2}x \rceil \times 9$, $(x, 15)]$ iii) $(x)$ $[(x, 15)$, $(\lceil \frac{2}{3}x \rceil, 9)$, 120, $\lceil \frac{2}{3}x \rceil \times 9$, $(x, 15)]$

and the type iv) $(x)$ $[(x, 15)$, $(\lceil \frac{2}{3}x \rceil, 9)$, $(\lceil \frac{1}{3}x \rceil, 6)$, 40, $\lceil \frac{1}{3}x \rceil \times 6$, $(\lceil \frac{2}{3}x \rceil, 9)$, $(x, 15)]$ v) $(x)$ have this topology $[(x, 15)$, $(\lceil \frac{2}{3}x \rceil, 9)$, $(\lceil \frac{2}{3}x \rceil, 6)$, $(\lceil \frac{2}{3}x \rceil, 9)$, $(x, 15)]$, the second type NNs vi) $(x)$ have $[(x, 15)$, $(\frac{3}{2}x, 9)$, $(\frac{3}{2}x, 6)$, $(\frac{3}{2}x, 9)$, $(x, 15)]$. In a list $[(a, b), c, (a, b)]$ representing an autoencoder, couples $(a, b)$ stand for convolutional layers with outputs of depth $b$ and spatial dimension $a$ (output shape obtained with omitted sampling and cropping layers), the singleton $c$ stands for FC layers of output size $c$. A reminder about autoencoders is given in Section 2.5. NNs convolutional layers have kernels of size 3, hidden Exponential linear units (ELUs), sigmoid output units and four convolutional layers on which a stride of 1 is applied, Adaptive Moment Estimation (Adam) [82] is used for the gradient descent in mini-batch of size 32 and the Log Loss on the output layer. Every NN is trained until convergence in 200 epochs. Each average ROC curve comes from ten instantiations of the model hyper-parameterization it features. The NN used on real data from the testbed SWaT is a 1D-CNN autoencoder has the following topology $[(600, 51)$,$(150, 25)$,$(38, 10)$,75,380,$(150, 25)$,$(600, 51)]$, taking time windows of size 600 and 51 dimensions.

**Performance criteria.** To visualize how a scoring based model orders data, two common methods are ROC curves and Precision Recall (PR) curves. For both, the Area Under the Curve (AUC) can be used as a measure of the model performance. ROC curves are preferred over PR curves when classes are equally represented in the dataset. Indeed, ROC curves are easier to estimate since linear interpolation is inappropriate for PR curves estimation [28]. But contrary to PR curves for which one of the rate, the so called precision, is a probability conditioned on the estimate of the class label, both rates of ROC curves are probabilities conditioned on the true class label. Therefore, ROC curves are the same for balanced or imbalanced datasets [131]. That is why PR curves are more suitable for imbalanced datasets.

In the case of simulated data, the goal is not so much to compare our method to the classical use of an autoencoder in anomaly detection for a specific industrial problem but rather to show that it can always improve the performance of the NN. The improvement significance, though, will depend on the peculiarity of the problem. So, we use ROC curves and ROC AUC. In the case of real data, we want to compare our method to the previous ones in the literature, so we will use the metric precision (TP/(TP+FP)), recall (also TPR) and F1 score (2.precision.recall/(precision+recall)). But since it only tells the performance of the model for one anomaly threshold, we will also give the PR AUC.

In [28], it is proven that when a ROC curve dominates another, the corresponding PR curves verify the same property, and thus both AUC promote the same model. The ROC curve estimation is more precise on balanced datasets than on imbalanced ones. Therefore, we use ROC curves with as much attacked cycles as normal cycles and plot separately the curves involving different attacks to prove the usefulness and the validity of our method. When two ROC curves cross each other at a certain FPR $r$, it simply means that the model the curve of which dominates the other before their intersection is better for FPRs lower than $r$ and worse for FPRs higher than $r$. In anomaly detection for industrial security, a low FPR is often preferred since a FPR too high would overload the ICS and compromise its availability. A security expert who can assess the specific constraints of the ICS and has an idea of the different attack type distributions occuring in the system can rather use one average PR curve for each model to evaluate or one average ROC curve, instead of separating attacks, in order to decide which model performs best. For example, if we suppose that each anomaly is as likely to occur and as dreadful as other anomalies, the graphs on Figure 2.6 indicates how the models perfom.

Since NNs are stochastic models, one needs to consider several ROC curves, ten in our cases, each from independent instantiations in order to compare different hyper-parameterization of the model. One should be careful not only about the number of instantiations to estimate ROC curves mean and variance, but also make sure that each instantiation have a different random seed than others from the very beginning and until the very end. For example, we did not notice right away that all instantiations in one parallel block had been mistakenly given a single seed during weights initializations because eventually, ROC curves were different due to a different seeding for shuffling data during training. So, the results was about the ROC curves for a model with particular starting weights. Fortunately, we detect this mistake because the results were not repeatable.

## 2.5 A baseline integrity score based on states and a 1D-CNN autoencoder

Before presenting the model, let us recall the structure of autoencoders and then CNNs. Autoencoders serve a whole class of DL-based methods. The original purpose of these NNs is to use this new representation for another task. This is why they are often used for dimension reduction, feature learning [54], image compression [113] or classification

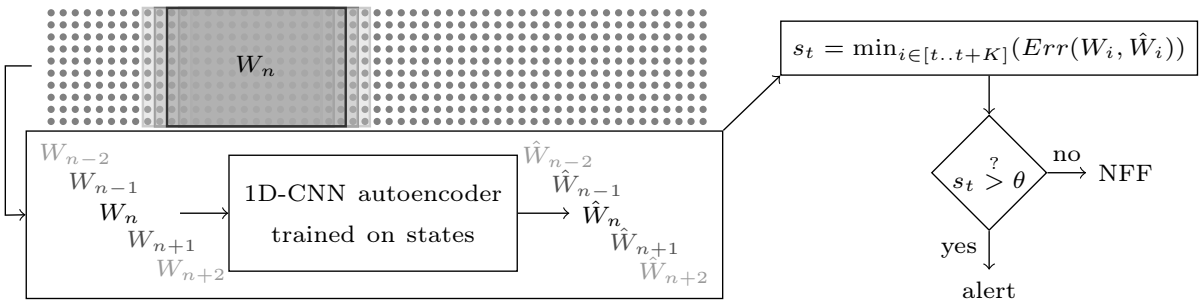



Figure 2.4 – Basic principles of our 1D-CNN autoencoder based integrity control. An 1D-CNN autoencoder trained on states reconstructs the sliding time window of width $K$ so that the smallest errors $s_t$ from time ranges of a fixed size $K$ constitute the scores used to raise an alert when higher than an anomaly threshold $\theta$.

[35]. Autoencoders are NNs whose input and output have the same dimension. They are composed of an encoder, the first layers, followed by a decoder, the last layers. The output of the encoder is called the code. Both encoder and decoder are trained so that the autoencoder output is the closest as possible of its input and the code can be considered as a new representation of the input. So the model does not need a label for the input. As a consequence, they are usually considered as unsupervised learning models. For this new representation to be useful, the code layer must face with constraints, like dimension reduction. Thus defined, an autoencoder will better reconstructs points from the training distribution than points far from it. That is why researchers saw in this model a way to define normality based on a dataset composed only of sample points from the genuine data distribution: reconstruction errors can be used as anomaly scores. In this context, constraints on the code determine the anomaly scoring function. CNNs, for their part, were originally derived from TDNNs to efficiently analyse images. They are NNs with significantly fewer parameters (also called weights) than a FC NN. A $n$D-CNN is composed of convolutional layers and pooling layers. The former consist of sets of filters that are sets of kernels (tensors of order $n$) operating the cross-correlation so as to capture spatial patterns along $n$ dimension. In the case $n = 1$, the model deals with temporal patterns. In layers after the first layer, the channel dimension corresponds to the number of filters that extract potentially different patterns, but the channel dimension of the input layer depends on the input shape. In a 1D-CNNs, for a $k$-dimensional time series, there are $k$ input channels. On their side, CNNs pooling layers change the spatial dimension without adding learnable parameters, further details in [2].

**Presentation.** We use autoencoders in order to learn what normal states are, either

one NN for every state or several NNs for different states, for example: two NNs, one for states represented in blue and the other for states in yellow in Figure 2.3. A 1D-CNN autoencoder is trained on time windows in the specified states from a near-deterministic system just as in the traditional setup. Inputs, randomly feeded to the NN, are time windows and only the time windows from normal regime is considered, so the 1D-CNN must have $k$ input channels with $k$ the number of registers. Therefore, instead of training the NN on every possible time windows, the training dataset contains only time windows related to some states of interest; all the states of interest if only one NN is used. This offers the opportunity to focus on well defined classes so as to tighten the definition of normality and allows the training time to be dramatically reduced. Indeed, in machine learning and especially in deep learning, the more complex is the task, the more examples the model needs for its learning. So, in our case, in order to better learn to reconstruct time windows from system cycles, one can add more cycles to the training set, or one can focus on specific part of the cycle, in other words, simplify the problem. An autoencoder will therefore more easily learn to reconstructs time windows in states of interest instead of all possible time windows from the cycles.

Of course, at testing time, one do not necessarily know which of the time windows of data from the physical process are in a specified state and which are not. Moreover, using the exact knowledge of states can delay the detection of attacks because one will only be able to compute $Err(R^n, \hat{R}^n)$ when $R^n$ is identified, whereas some evidence of an attack can be detected earlier. The same problem holds true with $Err(\tilde{R}^n, \tilde{\hat{R}}^n)$. For example, in a cyclical action system, because of the varying length of the cycle, one would have to wait until the end of a cycle to determine the fixed size time windows $(\tilde{R}^n)_n$ that correspond to the $n$ states, which is impractical. Hence, the average reconstruction error cannot be considered itself an anomaly score. However, it can be used to construct an anomaly score using fixed size time windows. In the case of a single autoencoder for every state, the anomaly score is simply the minimal value within a time windows, of its average reconstruction errors, of size the average stride between two states onsets. The idea is similar to the one from [35] where an autoencoder was used for speech segmentation, the sound corresponding to . Indeed, in Figure 2.5, the reconstruction errors of an autoencoder are lower around the moment of the cycle corresponding to states it has been trained on and the reconstruction errors between two states are not as high as on time windows from the end of the cycle where no states have been defined. So, the minimal value within a time windows of average reconstruction errors of an autoencoder (Figure 2.4) is a meaningful

anomaly score. In other words, we consider the reconstruction error of a sliding window with a stride of 1 and a duration $K$, thus the score $s_t$ to be compared to the threshold $T$ at time $t$ is the minimal value of $Err(W_i, \hat{W}_i)$ for $i \in [t, t+K]$, with $Err$ the Mean Squared Error (MSE). The parameter $K$ must be greater than the maximum distance between two consecutive states, so that at least one state is considered in this period. But it must also be less than the maximum distance between two consecutive states that do not overlap. Indeed, let us consider three consecutive state time windows $a = W_{i_1}, b = W_{i_2}$ and $c = W_{i_3}$ $(i_1 < i_2 < i_3)$, such that $a$ and $c$ do not overlap. If an anomaly appears in $b$ but not in $a$ and $c$, it is be missed: the high reconstruction error $err(b) = Err(b, \hat{b})$ is ignored (i.e. $\forall t \in [i_1, i_3]\,,\ s_t < err(b)$) since $err(b) > \max(err(a), err(c))$. In this way, the above-mentioned anomaly score is low under normal system conditions, and high in other cases. One way to verify both of these constraints is to take $K = max(L)/M$.

When several autoencoders are dedicated to different states, the anomaly score $S$ is the minimal value of the values of the aforesaid anomaly score, noted $S_i$ for autoencoder $i$, divided by a threshold $T_i$, so that $S = \min(S_i/T_i) > 1$ is the condition to raise an alert. This value is actually the one of the anomaly score of the autoencoder assigned to the current state, thus it is a meaningful anomaly score (cf. Figure 2.5).

The size $s$ of the time windows and the positions of these time windows in the cycles, entailing together the overlapping between states, have to be carefully considered. Indeed, the less overlapping between states an autoencoder is trained on, the more it can focus on these states. The reason is, if the strides between two states are small, then their corresponding time windows are similar according to temporal patterns. So, for an autoencoder with FC layers to benefit from the states, they have to be spaced out enough as shown by the first row of Figure 2.8. Yet, having little overlapping between states, implies straightforwardly a longer time detection, hence the usefulness for several NNs trained on disjoint subsets of states such that they do not overlap too much within each of the subset. Another practicality of having several NNs is that, the more NNs, the more efficient is the detection. Indeed, this is the generalization from the fact that, with enough examples, an autoencoder trained on states is more performant than an autoencoder trained on every time windows (cf. Figure 2.6), because in the first case, the NN has to focus on fewer patterns and thus better learn them. Parameter $s$ has to be greater than the average time between two states onsets so that there is always at least one time window representing a state and smaller than twice this average time so that there is at most one time window

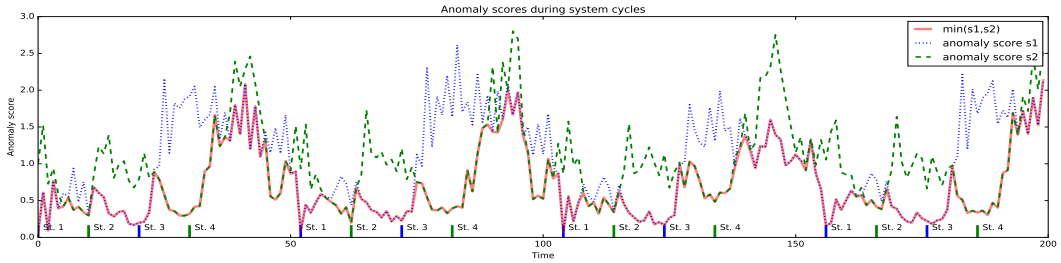



Figure 2.5 – Reconstruction errors during the system normal regime of two autoencoders trained on different states. The states have to be adequately spaced out according to the normality one wants to define. In order to shorten the detection time, different autoencoders can be trained on different states and the anomaly score would then be the minimum value of their reconstruction errors. The first NN is trained on states 1 and 3, and the second on 2 and 4. The reconstruction errors are normalized by the average reconstruction error over the validation set so that the scores of the two NNs are of the same magnitude and one can use the minimum score as a meaningful anomaly score. Normalizing according to the states, instead of the whole validation set, is also a solution assuming the knowledge of the current nearest state is known at testing time and even recommended when states have too different variabilities.

representing a state and eventually states have distinct representations.

**Analysis.** Data are simulated thanks to Sibriz. For brevity, we did not simulate cycles that would produce states with very different variabilities, i.e. we gave $\forall t, S_t = c$ (cf. Section 2.2). Hence, ROC curves from reconstruction error with and without the normalization according to states are alike. The idea of normalizing with respect to the state is straightforward as soon as the states are finite in number. In state space representation methods coming from safety engineering, the idea has already been suggested [46].

The length of time windows on which states are defined affects in opposite directions the performance of detection of global and local anomalies (Figure 2.7), also called long-term and punctual anomalies. Long-term anomalies are more easily detected with models trained on long time windows which possess more evidence of their presence and punctual anomalies are more easily detected with models trained on short time windows as these anomalies are less outweighed by the mass of information within. As for replay attacks, depending on similarities between the time series replayed after the genuine time series, the evidence of the anomaly can concentrate near the boundary between the concatenated time series or they can be scattered on both sides of the frontier. Replay attacks impact thus cannot be categorized into long-term or punctual anomalies. The conflict between learning long-term and punctual patterns leads us to define discriminative losses to be applied on an autoencoder code layer as in [21] to fine-tune the definition of integrity. In-

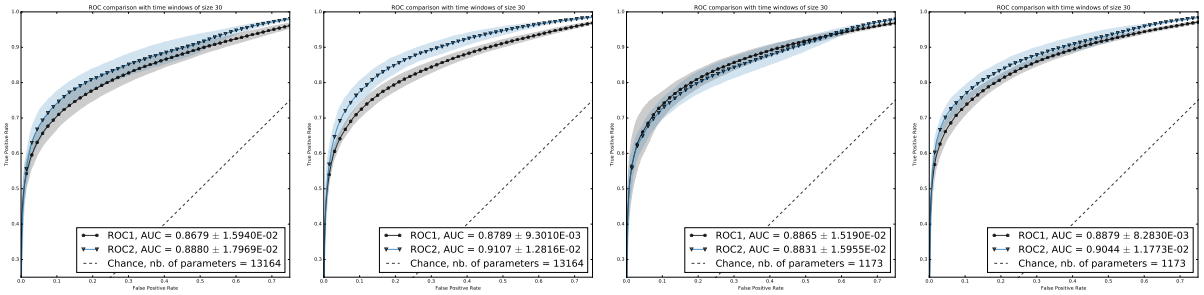



Figure 2.6 – Average ROC curves from reconstruction error of CNN autoencoders with FC layers (first two) and without FC layer (last two) from a dataset with all anomalies combined. Autoencoders are trained on states with relative strides around 20, apart from ROC1 for NNs trained on every time windows. The first and third figures NNs are trained on a dataset of three quarters the size of the one used to train the NNs of the second and fourth figures. Improvement due to additional examples is more important for NNs trained on states. The second and fourth figures are detailed in Figure 2.8.

deed, suitable constraints should be able to prioritize generalization of long-term patterns over punctual patterns so that the NN overfits more on the latter and thus better detect punctual anomalies. Conversely, other constraints could lead to better long-term anomaly detection at the expense of the detection of punctual anomalies.

We observe that 1D-CNN autoencoders with FC layers better detect replay attacks than NNs without FC layers (Figure 2.7). This can be explained by the fact that CNNs without FC layers are by their structure not suitable for detecting this kind of anomaly especially for time series with little variation like the one of near-deterministic systems. Indeed, if we imagine the counterpart of a replay attack on images, let us say that a 2D-CNN autoencoder was trained on images of a single house approximatively pictured from always the same point and with the same weather conditions at the same hour on different days. The counterpart anomaly of a replay attack would then be like giving as input to the NN an image of the house such that the left part of the image comes from a first photo and the right part from another. Because only convolutional and sampling layers are used, the reconstruction will be correct except that it would exhibit a blurry boundary between the two images where the abnormal input would have a sharp frontier. Thus, the anomaly score obtained from the reconstruction error will not reflect the degree of abnormality we could expect from this kind of input. On the contrary, with a FC layer the sharp abnormal boundary between two concatenated images can potentially have an impact of every pixel of the reconstructed image and then yield higher reconstruction error. So CNNs with at least one FC layer are preferable for replay attack detection. But

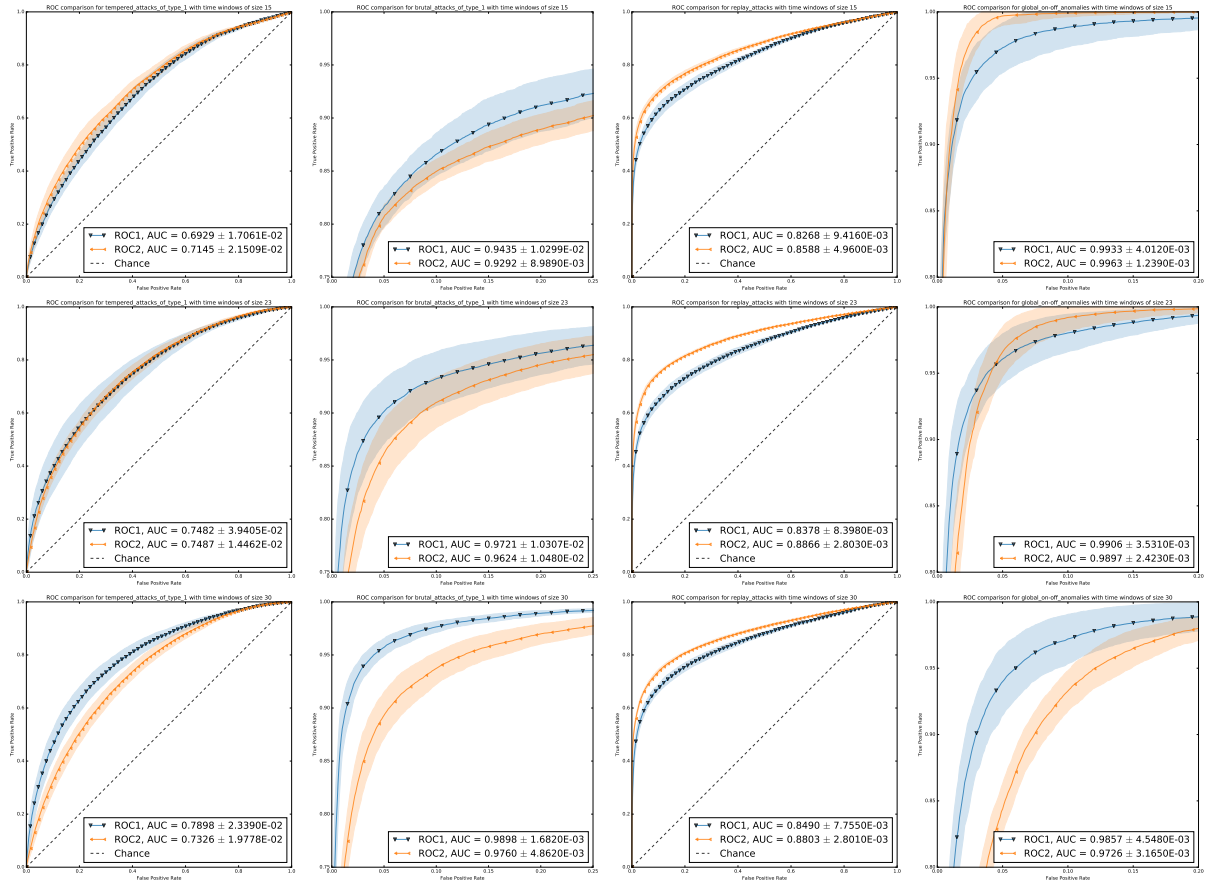



Figure 2.7 – Evolution of average ROC curves from reconstruction error of CNN autoencoders with FC layers (ROC2) or without (ROC1) against the time windows length. All autoencoders trained on 5 states with strides around 10. The first row correspond to time windows of length 15, the second, 23 and the third, 30. Each column corresponds to a kind of anomaly. From left to right: the temperered attacks of type i), the brutal attacks of type i), the replay attacks and the global on-off anomalies.

it comes with a lower performance to detect other types of anomalies compared to CNN without FC layers (Figure 2.7). Our experiments exhibit two options to mitigate these downsides, that is to improve the detection of long-term and puntual anomalies by CNNs with FC layers, and the detection of replay attacks by CNNs without FC layers.

In the case where there is no limitation for the capacity of its NN, one just has to train a CNN autoencoder with FC layers on states with large enough relative strides as depicted in the first row of Figure 2.8. Indeed, it is important to consider the length of time windows beside the number of states as it determines their overlapping. The incidence of the overlapping between states can be seen in Figure 2.5. This figure shows

the reconstruction errors of two autoencoders, each one trained on two states, on every time windows from the time series induced by the normal regime of a cyclic industrial system. We can see that, unsurprisingly, the reconstruction errors are lower around the moment of the cycle corresponding to states they have been trained on. But we can also notice that the reconstruction errors between two states are not as high as on time windows from the end of the cycle where no states have been defined. Indeed, if the strides between two states are small, their corresponding time windows will be similar according to time patterns. This explains why for an autoencoder with FC layers to benefit from the states, they have to be spaced out enough, otherwise it is better to train also on the time windows between the statesas shown in Figure 2.8 first row. However, if the states are too much spaced out we will have a longer detection time and potential false negatives if the anomalies lie right between two states. A small detection time can be necessary if one wants to apply counterattacks or resilience policies where a fraction of a second is crucial. To overcome these drawbacks, one can train several CNN autoencoders with FC layers on disjoint subsets of states such that they do not overlap too much within each of the subset. At test time the anomaly score would then be the minimal reconstruction errors among the ones of all the NNs as in Figure 2.5. This figure shows that the minimal reconstruction error of the autoencoders is actually the one of the autoencoder assigned to the current state, hence it is a meaningful anomaly score. The more subsets, the smaller the detection time. Finally, in order to better focus on physical characteristics of the system, states within a single subset of states – on which a single autoencoder is trained – can even have no overlapping at all as in Figure 2.3. In this figure, there are two subsets of states highlighted in blue and yellow.

In the case where there is indeed a limitation for the capacity of the NN, one can train a CNN autoencoder wihtout FC layers. Unlike CNNs with FC layers, too much overlapping between states does not worsen the performance compared to the baseline. It will only improve the detection of some anomalies rather than others. Specifically, a large overlap favors the detection of replay attacks while a little overlap favors the detection of long-term attacks. In any case, the performance is better than or equivalent to the one of the baseline and above all, it is even better than the performance of the CNNs with FC layers trained on every windows whereas the latter has around ten times more parameters.

Finally, to witness a significant improvement in the detection thanks to the states, one has also to be sure that the number of training data is sufficient as depicted in Figure

Table 2.1 – Comparison between our method (1D-CNN Autoencoder trained on states) and other anomaly detection methods for physical data from SWaT evaluated with the precision, the recall and the F1 score. Bold font for the highest number in a column, italic font for the second-highest number in a column.

| Method | Precision | Recall | F1 |
|---|---|---|---|
| DNN [71] | **0.983** | 0.678 | 0.803 |
| OC-SVM [71] | 0.925 | 0.699 | 0.796 |
| TABOR [98] | 0.862 | 0.788 | 0.823 |
| 1D CNN combined [85] | 0.958 | 0.639 | 0.767 |
| 1D CNN ensembled [85] | 0.912 | 0.861 | 0.886 |
| our method with FPR = 0.06 | **0.983** | *0.890* | *0.934* |
| our method with FPR = 0.14 | *0.964* | **0.999** | **0.981** |

2.6. In this situation, each anomaly is supposed to be as likely as other anomalies and for a CNN with FC layers the error, which can be summarized by the area over the curve, is reduced by one quarter. Indeed the AUC of the baseline is 0.8789 while with the states the AUC equals 0.9107, and $\frac{((1-0.8789)-(1-0.9107))}{(1-0.8789)} = 0.2625$ (Figure 2.6). But if we suppose that replay attacks are more likely to occur, then the error will be even more reduced compared to the baseline since for these anomalies the error is reduced by more than one third, $\frac{((1-0.8237)-(1-0.8859))}{(1-0.8237)} = 0.3528$ (Figure 2.8).

Another way to improve the detection is to apply a loss on the code layer. Thereafter, we will present the result on the well-known testbed SWaT. In the next section, we will present a loss useful to make a trade-off between long-term and punctual anomaly detections in addition to improve the overall detection rate.

**Results on the testbed SWaT.** The state-of-the-art method [85] for anomaly detection in physical data from SWaT [50] was able to detect 31 out of the 36 attacks. As we can see in Figure 2.9, our method is able to detect 35 out of the 36 attacks (anomaly threshold = 0.011). The only attack for which we cannot say whether our method detects it is the attack 10 that is directly followed by a similar but more severe attack as the first did not have the expected outcome. Hence, we cannot say whether the attack 10 alone would have been detected and we can only attribute the high anomaly scores following these attacks to the attack 11. Moreover, Table 2.1 shows the precision, the recall and the F1 score of previous methods gathered in [85] and an instance of our model for the dataset from SWaT. The performance of our model in this table is obtained with a threshold equal to 0.011 choosed to have a for relatively low FPR (empirical FPR: 0.06) and 0.0046 for

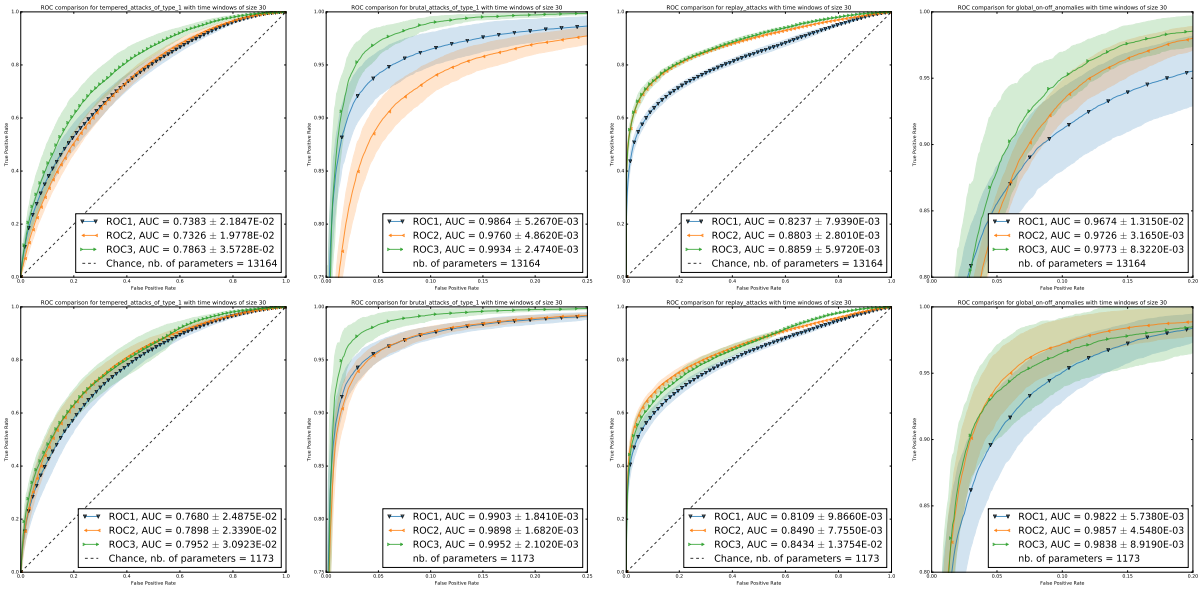



Figure 2.8 – Average ROC curves from reconstruction error of CNN autoencoders with FC layers (first row) and without (second row) trained on every time windows (ROC1, same NNs as those of ROC1 in second row of Figure 2.6) or on states with strides around 10 (ROC2) or 20 (ROC3, same NNs as those of ROC2 in second row of Figure 2.6). Same anomalies as in Figure 2.7.

a relatively high FPR (empirical FPR: 0.14). In the latter case, our method detects all attacks and still has a reasonable precision (0.964). Instead of coarsely estimating FPRs to choose the threshold corresponding to the highest acceptable FPR, this can be made using Extreme Value Theory. As our method relies on a stochastic model, we train 10 1D-CNN Autoencoders and compute the average performance metrics obtained with the same thresholds. For a threshold at 0.0046, the average precision is $0.95670 \pm 1.35 \times 10^{-4}$, the average recall is $0.9984691 \pm 7.89 \times 10^{-6}$ and the average F1 score is $0.977101 \pm 3.84 \times 10^{-6}$. For a threshold at 0.011, the average precision is $0.995632 \pm 2.26 \times 10^{-5}$, the average recall is $0.87056 \pm 6.04 \times 10^{-4}$ and the average F1 score is $0.92868 \pm 1.78 \times 10^{-4}$. In summary, the average PR AUC for our anomaly score is $0.99810143 \pm 8.51 \times 10^{-7}$ (the random baseline being 0.787 given that we kept about 122000 normal sample points for the test).

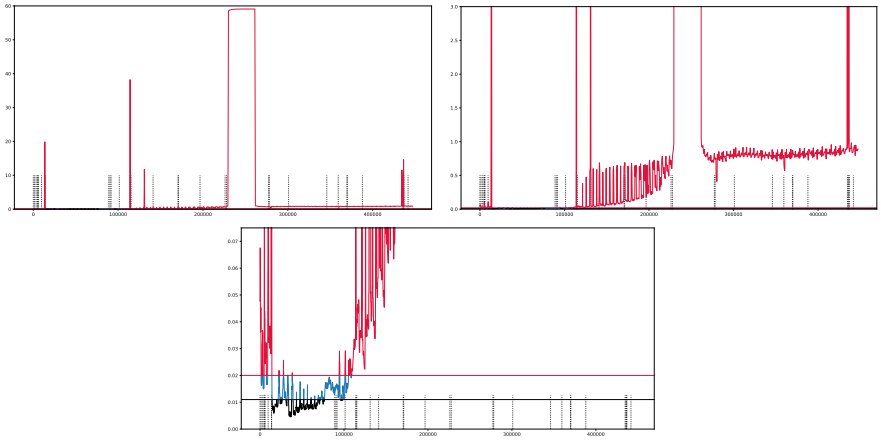


Figure 2.9 – Anomaly scores of our method at three different scales, that is the minimal value on a time window of size 366 of reconstruction errors of a 1D-CNN autoencoder trained on states of size 600 (i.e. on time windows covering 10 minutes) from physical data of the testbed SWaT. Dotted vertical lines represented starts of attacks. The black horizontal line representes a threshold at 0.011 for which precision recall and F1 score are reported in Table 2.1 (threshold of 0.456 from Table 2.1 not shown in these plots) and the red horizontal line representes a threshold at 0.02 (for which we achieve an almost perfect precision (0.999), we still have a reasonable recall of 0.790 and so a reasonable F1 score of 0.883).

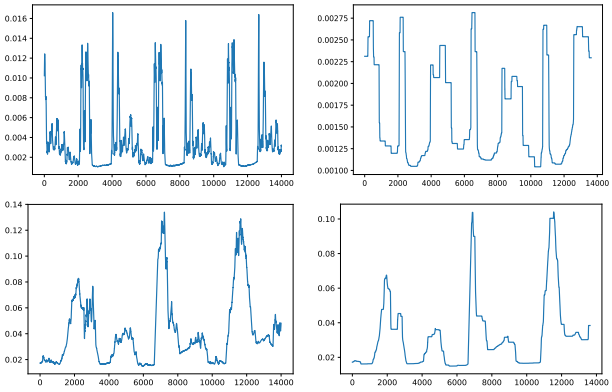


Figure 2.10 – Reconstruction errors from the autoencoder trained on states (first column) and anomaly scores (minimal values on time windows of size 366 of reconstruction errors; second column) during normal operation (first row) and abnormal operation (second row) of the CPS of SWaT. Note the change of scale in the normal case and the absence of change of scale in the abnormal case when taking the minimal values in time windows of reconstruction errors.

## 2.6 Losses and architectures of autoencoders to refine the integrity specification

In DL, a loss function is a function applied on a layer, other than the input layer, of a NN to determine the change in the parameters of the NN that will goad the learning algorithm to minimize, with enough epochs on a training dataset $D$, the objective function, that depends on $D$ and potentially several loss functions. A well-known loss function is the MSE, usually applied at the output of a NN. Regarding losses applied on a hidden layer, let us mention the example of the Center Loss [150] (2.2) that a version of which is applied on the code layer in [21] in addition to the MSE so as to take into account the class of the sample point $x$ by minimizing the distance of the encoded sample point $Enc(x)$ from the centroid $c_{s(x)}$ of its class $s(x)$. Thereafter, we will say that a NN is trained under a loss $L$ jointly with another loss $L'$ when one concerns the output layer, referred as the main loss, and the other concerns a hidden layer. The main loss (output layer) weight equals 1 and the additional loss weight will be denoted $\lambda$.

$$x \to \frac{1}{2} \times \|Enc(x) - c_{s(x)}\|_2^2 \tag{2.2}$$

### 2.6.1 Rational Sampling

The structure of CNNs has been described in section 2.5. In this section we present a new pooling layer that we found useful to control the CNN capacity independently from the FC code layer size. This is important because the representational capacity of the encoder, that is its «*family of functions* [it] *can choose from when varying the parameters*» [54], has to be in balance with the CNN autoencoder capacity. Yet, whereas modifying the spatial dimensions of a convolutional layer does change its number of parameters, it will drastically change the number of parameters of a FC layer coming right after. It also serves us to conduct our experiments on comparing CNNs with different capacity. To better control the information loss, one can use pooling layers, the most common ones being max-pooling and average-pooling. In [77], it is shown that changing the spatial dimension by a rational factor allows a CNN layer to learn more complex patterns since it has increased the classification performance of the NN.

There exist in the literature, two methods to reduce or increase spatial dimensions

from one layer to another by a multiplicative factor other than an integer. The first [56], Fractional Max-Pooling (FMP), uses max-pooling layers and hinges on a pseudorandom stride that can take the value 1, or 2: by choosing a stride that has a lower probability to take the value 2 than 1, it reduces the image by a factor less than 2. The second [77], Non-integer Strided Sampling (NSS), uses average-pooling layers with weights that are computed to match a visual interpretation of up/down-sampling. These two methods, which were designed to be used with images, are respectively unsuitable and unnecessary complicated for our use of 1D-CNN autoencoders. Thus, these methods are not suited to our problem. A sampling layer that changes layers spatial dimensions by a factor around $n/m$ with a couple of an upsampling layer $Upsamp(n)$ directly followed by a downsampling layer $Downsamp(m)$ is enough to ease the learning process of a NN faced with binary data. Let us call it Rational Sampling. Rational Sampling therefore operates a downsampling if $n < m$, and an upsampling if $n > m$. Moreover, one is not limited to a single sampling layer type, for example one can use max-pooling as well as average-pooling layers as $Downsamp(m)$. In the experiments, the $Upsamp(n)$ repeats data $n$ times (for example, $Upsamp(2)([1, 2, 3]) = [1, 1, 2, 2, 3, 3]$), and $Downsamp(m)$ operates a max-pooling with strides $m$ and pool size 2 (for example, $Downsamp(3)([1, 1, 2, 2, 3, 3]) = [1, 3]$).

Since an autoencoder needs to have the same dimensions at the output layer as at the input layer, it is interesting to have an automatic way to find back the dimension of the input layer thanks to a setup on the pooling layers. In the following, we assume to be in 1D, but the same applies in 2D. Depending on the spatial dimension $s$ of the layer to be sampled and integers $n$ and $m$ used in the Rational Sampling layer, one can have to choose between the ceil and the floor of $(s \times n)/m$ by executing an ultimate pooling or not. In the case of an autoencoder, we decide to take the ceil before the code and the floor after the code so that the symmetry of the autoencoder can be respected by using $RatioSamp(n, m)$ and $RatioSamp(m, n)$ respectively before and after the code and so the input dimension is systematically retrieved. Let us note that when $m \mid s \times n$, $RatioSamp(n, m)$, with an average-pooling layer as $Downsamp$, is equivalent to NSS up to a weighting coefficient [3]. Unlike NSS, our method is not confined to the average pooling but works with any downsampling.

3. In python with the library keras [19], it is straightforward: to deal with the floor$((s \times n)/m)$ one can add a cropping layer whenever needed in the decoder part after the Rational Sampling layers, because by default, keras executes an ultimate pooling when downsampling by factor that is not a multiple of the previous layer size.

### 2.6.2 The Tempered Center Loss on a Fully Connected layer

**Presentation.** The goal here is to propose a way to make a trade-off between long-term and punctual anomaly detections in order to specify data integrity. To make the most of the knowledge of the current state of the system to be monitored, one can train a 1D-CNN autoencoder under the main loss jointly with a loss applied on the code that gives a prior knowledge, either explicitly or implicitly, on the class of the data point. In this section, we propose the TCL which can be used to give an explicit knowledge on the state of a data point. As stated in [54], « *in practical deep learning scenarios, we almost always do* [find that] *the best fitting model (in the sense of minimizing generalization error) is a large model that has been regularized appropriately* ». In anomaly detection, one way to tighten the definition of normality is to simply give the NN more parameters, since the bigger the NN the more likely it is to overfit. This is another reason for seeking a loss suited to FC layers. In anomaly detection, to narrow the definition of normality one can give the NN more parameters, since the bigger the NN the more likely it is to overfit. Yet, with FC layers, changing the output dimension is a simple way to control the number of parameters. This is why we propose a loss to be applied on such layers.

To distinguish normal sample points from abnormal ones, one can tackle this objective by moving away normal sample points from abnormal sample points within the latent space of the code layer. Since the training dataset is supposed free or nearly free from anomalies, one way to achieve this is to concentrate data around a centroid so as to force the model to find, as far as possible, commonality between normal sample points. This can be done with the Center Loss (2.2), initially defined for separating clusters from each other in classification tasks [150]. From now on, a cluster refers to a set of time windows representing a particular state. So the first approach is to use the Center Loss with only one class, the normal regime. The idea of using this loss or a version of it on the code layer was proposed in [21] for a better dimension reduction. A second approach is to cluster encoded sample points according to their states, thanks again to the Center Loss this time refered as being in state class mode, in order to even more focus on state classes. In our context, the detection rates benefits from these methods, but prioritization of types anomaly detection is not tackled. To this purpose, a third approach is to concentrate the clusters formed by the Center Loss in relation to the states, near the origin (always in the code layer latent space), getting the best of two worlds. Based on the premise that there is an opposition in learning long-term and punctual patterns, one can think of a method

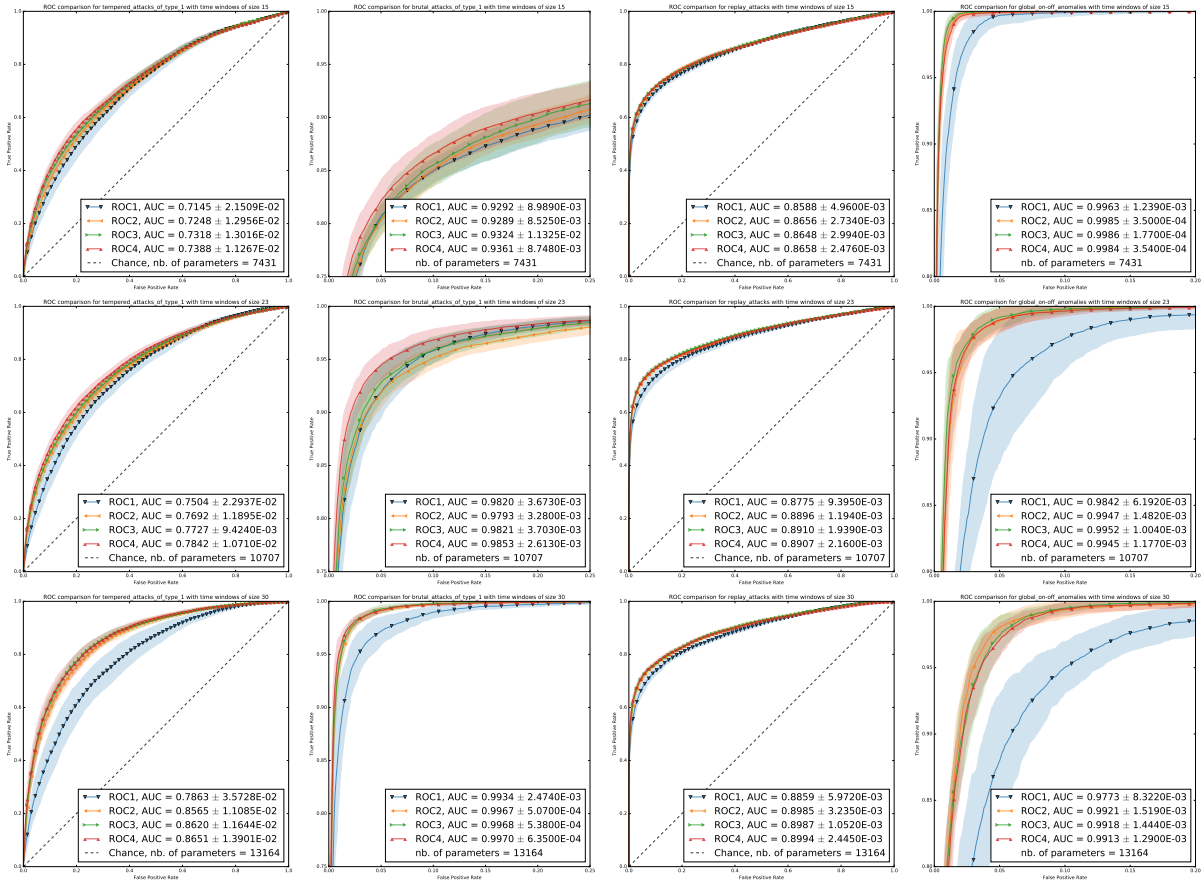



Figure 2.11 – Evolution of average ROC curves from reconstruction error of CNN autoencoders with FC code layers of 45 units trained with the Center Loss in one class mode (ROC2) or applied on state classes (ROC3), with the Tempered Center Loss (TCL) applied on state classes (ROC4) and with only the main loss (ROC1). The NNs from the first line of type ii)(15) and trained on 5 states with relative strides around 10; the second line: ii)(23), 3 states with relative strides around 15; and the third line: ii)(30), 2 states with relative strides around 20. From left to right: the temperered attacks of type 1), the brutal attacks of type 1), the replay attacks and the global on-off anomalies.

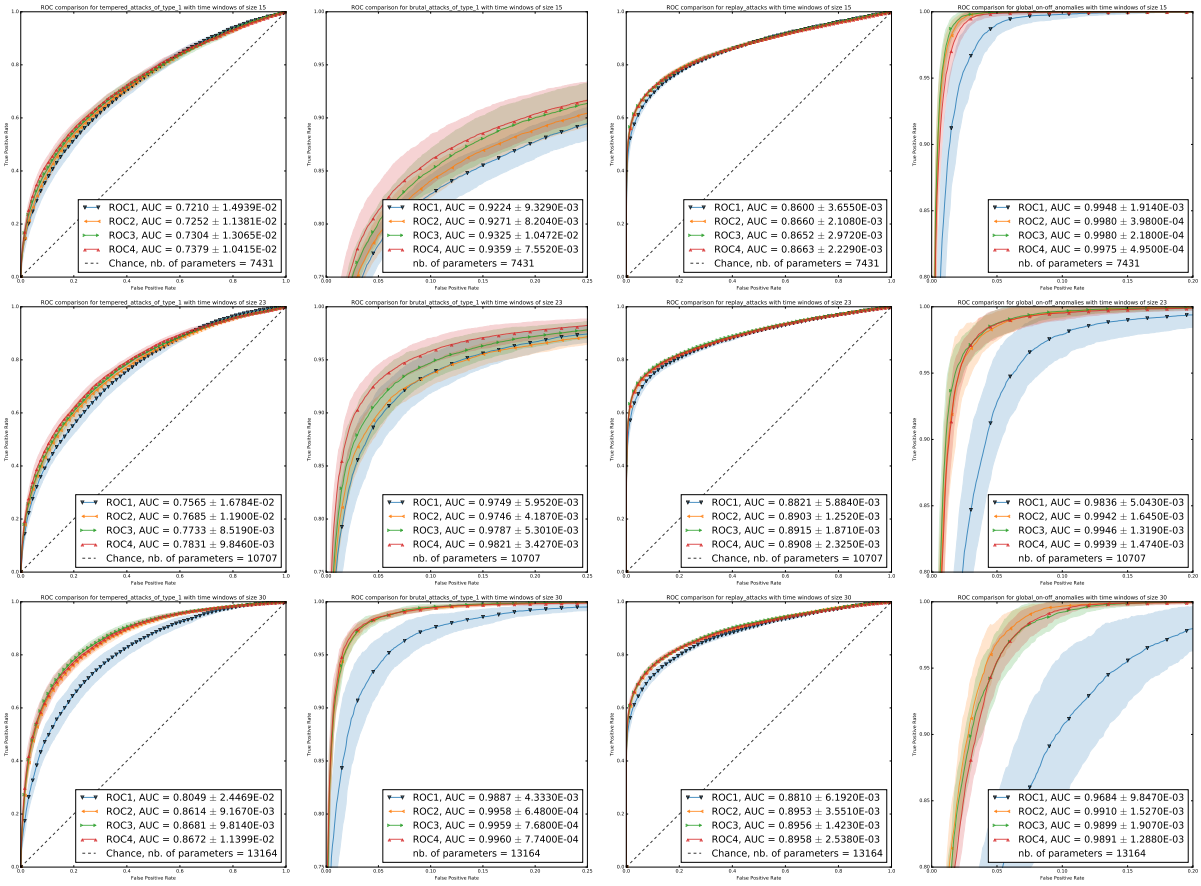



Figure 2.12 – Same as for Figure 2.11 except that anomaly scores are normalized according to the state

that tends to better generalize on punctual patterns and so overfit more on long-term patterns. This way, even slight deviations from normality of long-term patterns should lead to high anomaly scores. The idea is then twofold 1) employ a FC layer which captures punctual patterns more easily than long-term patterns 2) regulate the concentration of the sample points near the origin, in the aforesaid layer representation, according to their distance to the centroids of their class. Indeed, we consider that a sample point far from its class centroid should contribute to the migration of its cluster less than a closer sample point since the latter is a better representative of the class. We consider the origin to concentrate the clusters, because, it is known that good generalization comes with small activations in NN, which leads to plenty of work in activity regularization [54]. Admittedly, in anomaly detection, too much generalization of the autoencoder itself is not the goal since it would make the reconstruction error useless, still one can benefit from the conflict between learning long-term and punctual patterns. In our case, a FC layer, which has no structure describing the relation in time, will tend to generalize more efficiently on punctual than long-term patterns. Hence, an activity regularization on a FC layer of a CNN tends to increase the long-term anomaly detection rate, allowing a trade-off between the detection of punctual and long-term anomalies.

As the Center Loss is suited to classification, we thought of an additive term that is not restricted to our problem. The drawback of the Center Loss is that it does not consider inter-class variations. Our contribution is meant to be relevant not only as a regularization in the case of one class anomaly detection but also as an inter-class variability aware method for classification. Let us review the alternative to the Center Loss since its advent. Since the contribution of Wen et al. [150] where the Center Loss was suggested to build an objective function that helps a NN to find a discriminative representation of data labeled by class, many articles have focused on solving the drawback that only intra-class distance is considered by the Center Loss. All these articles, such as the first four [155, 79, 125, 21], have in common that they address the problem of inter-class variation by directly tackling the desired relationship between classes and add a lot of calculations. Authors of [155] propose to penalize the squared difference between the average class centroids norm and the norm of the centroid of the class of the point of interest to position cluster centroids near a sphere of radius the average class centroids norm in order to overcome the problem of imbalanced dataset. In [79], they force the representation to spread each feature uniformly from the global center of an hypercube by penalizing the positive difference between the furthest squared distance of the sample point feature – to which they add a

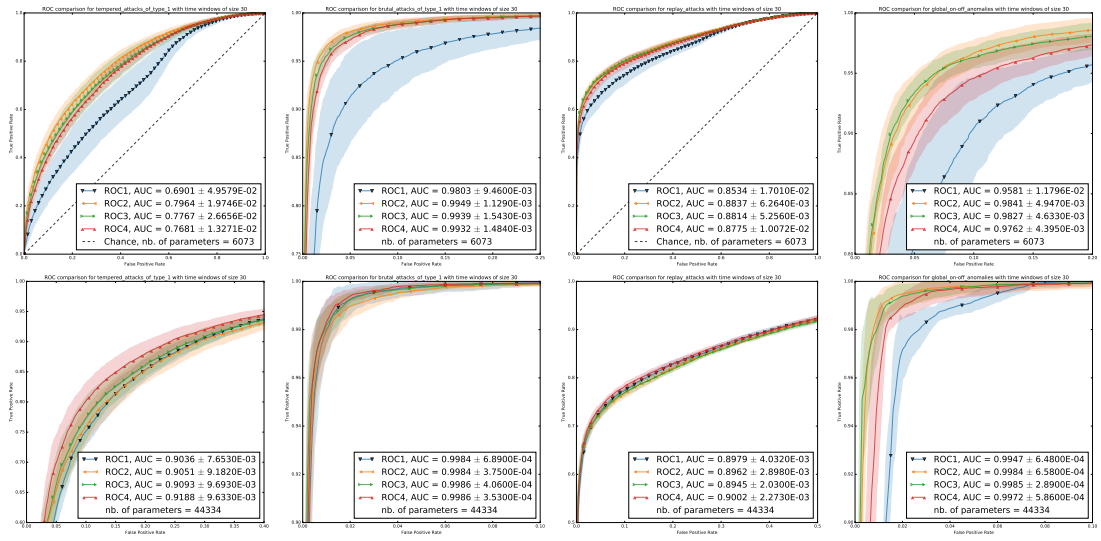



Figure 2.13 – Average ROC curves from reconstruction error of CNN autoencoders with FC code layers and low capacity (top row) or high capacity (bottom row) trained with the Center Loss in one class mode (ROC2) or applied on state classes (ROC3), with the TCL applied on state classes (ROC4) or with only the main loss (ROC1). High capacity NNs are of type iii)(30), low capacity: iv)(30), all trained on two states with relative strides around 20. Same cycles than in Figure 2.11.

margin – from the global center feature and the squared distance of the point from the global center in order to «*extract more meaningful distance on feature space*»[79]. In [21, 125], they directly exploit inter-class distances to penalize small ones, either by appending an additive term [21] or a multiplicative term [125] to the Center Loss.

Rather, we define the TCL which is the Center Loss to which a term is added in order to separate clusters at little cost, with $s(x)$ the class of $x$, that is, here, the state of the system:

$$\mathcal{L}_{TC} = \frac{1}{2} \times \|x - c_{s(x)}\|_2^2 + \frac{\epsilon \times (\tau - \|c_{s(x)}\|_2)^2}{1 + \log(1 + \|x - c_{s(x)}\|_2^2)} \tag{2.3}$$

This loss enables sending clusters away from each other with a reduced computational cost, because when the goal is only to separate classes, one can tolerate large clusters as long as they are far away from each other. The second term is inspired from the Center Invariant Loss [155] which puts clusters centroids near a sphere for two purposes, that are mitigating the imbalance of the multimodal distribution and indirectly increase inter-class distances. Only the second aspect interests us, hence the arbitrary radius $\tau$. Thus, the TCL mitigates the tendency of the Center Loss to make the clusters collapse towards the origin with hardly any additional calculations by indirectly increasing the inter-class

dispersion.

Going back to our problem, to carry out the third approach, we propose to use the TCL with $\epsilon > 0$ and $\tau = 0$, in order to gather the clusters near the origin all the more they are shrunk – instead of separating them thanks to a large $\tau$ – so as to regularize the activation of the features of a FC layer, since encoded data points will migrate near the origin in addition to explicitly being clustered according to their state.

Table 2.2 – Comparison between AUC in % of average ROC curves of anomaly scores from the same NNs that in Figures 2.11, 2.12 and 2.13, with normalization according to the state (white rows) and without (green rows). Largest AUCs between the ones with and without normalization are in bold font. The states used in the experiments have similar variabilities. In each box, the four AUCs relate to autoencoders trained, in order, 1) with only the main loss, 2) with the Center Loss in one class mode 3) with the Center Loss in state classes mode 4) or with the TCL.

| Capacity | Tempered attack | | | | Brutal attack | | | | Replay attack | | | | Global on-off anomaly | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6073 | 69.01 | 79.64 | 77.67 | 76.81 | **98.03** | **99.49** | **99.39** | **99.32** | 85.34 | **88.37** | 88.14 | **87.75** | **95.81** | **98.41** | **98.27** | **97.62** |
| | **75.46** | **81.90** | **80.88** | **80.40** | 97.44 | 99.39 | 99.29 | 98.99 | **85.71** | 88.23 | **88.20** | 87.63 | 94.59 | 98.18 | 98.04 | 96.85 |
| 7431 | 71.45 | 72.48 | **73.18** | **73.88** | **92.92** | **92.89** | 93.24 | **93.61** | 85.88 | 86.56 | 86.48 | 86.58 | **99.63** | **99.85** | **99.86** | **99.84** |
| | **72.10** | **72.52** | 73.04 | 73.79 | 92.24 | 92.71 | **93.25** | 93.59 | **86.00** | **86.60** | **86.52** | **86.63** | 99.48 | 99.80 | 99.80 | 99.75 |
| 10707 | 75.04 | **76.92** | 77.27 | **78.42** | **98.20** | **97.93** | **98.21** | **98.53** | 87.75 | 88.96 | 89.10 | 89.07 | **98.42** | **99.47** | **99.52** | **99.45** |
| | **75.65** | 76.85 | **77.33** | 78.31 | 97.49 | 97.46 | 97.87 | 98.21 | **88.21** | **89.03** | **89.15** | **89.08** | 98.36 | 99.42 | 99.46 | 99.39 |
| 13164 | 78.63 | 85.65 | 86.20 | 86.51 | **99.34** | **99.67** | **99.68** | **99.70** | **88.59** | **89.85** | **89.87** | **89.94** | **97.73** | **99.21** | **99.18** | **99.13** |
| | **80.49** | **86.14** | **86.81** | **86.72** | 98.87 | 99.58 | 99.59 | 99.60 | 88.10 | 89.53 | 89.56 | 89.58 | 96.84 | 99.10 | 98.99 | 98.91 |
| 44334 | 90.36 | 90.51 | 90.93 | 91.88 | **99.84** | **99.84** | **99.86** | **99.86** | **89.79** | **89.62** | **89.45** | **90.02** | **99.47** | 99.84 | 99.85 | 99.72 |
| | **90.41** | **90.77** | **91.13** | **92.05** | 99.78 | 99.80 | 99.84 | 99.85 | 89.48 | 89.48 | 89.36 | 89.95 | 99.24 | 99.84 | 99.85 | 99.72 |

**Analysis.** Using the TCL or the other losses improve the detection for every attack type. For example, in the last row of Figure 2.11, the error, which can be summarized by the area over the curve, is reduced by one third for tempered attack of type 1), one half for brutal attack of type 1), one tenth for replay attacks and nearly two thirds for global on-off anomalies, in comparison with the baseline. Under some assumptions, one can make a trade-off between the detection of long-term and punctual anomalies thanks to the TCL. For example, in the last row of Figure 2.13, by applying the TCL instead of the Center Loss in state mode, the error is reduced by one tenth (and one seventh against the baseline) for the detection of tempered attack of type 1) but it doubled for global on-off anomalies (yet it is still half the error of the baseline). In industrial security, such differences can be crucial.

The TCL has to be applied on a FC layers as convolutionnal layers have not enough representation capacity to allow this loss to improve the discrimination between classes. Moreover, like for the Center Loss, the states have to not overlap too much if one wants to benefit from this loss. Indeed, the Center Loss and the TCL are not suitable for states that overlap too much since these losses aim to discriminate between the different states

and the more they overlap the more similar neighbouring states are. However, it could be desirable to have time windows with length large enough to detect global anomalies. A first solution would consist in defining less but longer states. For example, in Figure 2.11, the states and the strides increase along the columns to keep the benefit of the losses. This solution can suffer from two problems that are a longer time to detection and potential false negatives if the anomaly lies right between two states. To overcome these drawbacks, one can train several CNN autoencoders with FC layers on different subset of states such that they do not overlap too much within each of them and the score would then be the minimal reconstruction errors among the ones of all the NNs described in section 2.5.

The choice of the parameter of the TCL follows two different rationales according to the problem one wants to solve, classification or anomaly detection. When used for classification, i.e. $\tau \gg 0$, we recommend to first try $\epsilon < 1$, since the main objective is finding a good centroid for each cluster and separating the cluster comes next. But for anomaly detection, i.e. $\tau = 0$, $\epsilon$ directly describes the trade-off between detecting long-term and punctual anomalies, that is high $\epsilon$ for detecting long-term anomalies and low $\epsilon$ for punctual ones, as long as the autoencoder is given enough parameters and a code large enough against the input dimension. Indeed, let us remark that in Figure 2.11, for each CNN autoencoder the code from the FC layer is kept equal to 45 while the time windows length increases along the columns of Figure 2.11. In this Figure, the TCL has the strongest effect beside the other losses in the second row, that is for the NN with capacity 10707. This remains true with the normalization according to the state (see rows for NNs with capacity 7431, 10707 and 13164 in Table 2.2). This teaches us that the constraints imposed by the TCL has the intended impact as long as the code size is congruent with the NN capacity. So, if one wishes to use an autoencoder with high capacity, that is with a large number of parameters, the representational capacity of the encoder has to be significant enough to witness the impact of the TCL as verified in the second row of Figure 2.13. If there is a strong constraint on the NN number of parameter, one should use the Center Loss in the one class mode as shown in the first row of Figure 2.13. Rational Sampling is therefore important since it allows controlling the spatial dimension size of the layer before the FC code layer. As a consequence one can completly decide on the number of parameters of the NN and the code size independently of one another. Finally, it is interesting to attest that the properties of the TCL and the Center Loss examined above can be retrieved when using Rational Sampling reducing the spatial dimension by two thirds or one third before the code instead of by half (see Figure 2.13) and thus our

method can benefit from this sampling layer.

We suggested that normalizing according to the state could be necessary in case of significant differences of variabilities among the states. The reasoning is that the more variable is the state, the more difficult will be to learn to reconstruct the input, so the larger will be the anomaly score on time windows of the aforesaid state. It turns out that even if the variabilities are not significantly different among the states, normalization can have an impact on the anomaly score. We see in Table 2.2 that when the states have similar variabilities, normalizing according to them will improve the detection of subtle anomalies (from tempered attacks of type 1)) at the expense of the detection of more obvious ones (from brutal attacks and global on-off anomalies), the replay attacks being in between subtle and brutal anomalies. The few exceptions in Table 2.2 for the Tempered Attack and the capacity 7431 and 10707, comes from the fact that the metric AUC of ROC curves is a useful summary of the performance but does not tell everything. Comparing the plots of the ROC curves with the score normalized according to the state in Figure 2.12 to their counterparts in Figure 2.11 teaches us that even when the metric AUC is not improved by the normalization according to the state for the Tempered attack, the area loss comes from the high values of FPR, but TPR are still increased for low values of FPR. All of this indicates that the errors of reconstruction of the inputs of the same state are better ordered for points near its normal representation than far from it, which is conceivable since NNs behavior far from the distribution underlying the training set is unpredictable [138]. Hence, when using the reconstruction error from an autoencoder with a FC layer as an anomaly score, the normalization according to the state is itself a tool for a security expert to be able to define data integrity based on his knowledge of dreadful types of attack or for a forensics scientist who wishes to focus on subtle attacks.

Finally, let us note that our results support the conflict between learning long-term and punctual patterns exposed in section 2.5. Indeed, the long-term anomalies, induced by tempered and brutal attack, involve AUCs in the row corresponding to NNs with 6073 parameters of Table 2.2 comparable to AUCs in the third row, corresponding to 10707 parameters, because NNs of the first row were trained on time windows of length twice the length of time windows NNs of the third row were trained on. And as a counterpart, for punctual anomalies, here global on-off anomalies, the performances of NNs of the first row are clearly worse than the ones of the NNs of the third row.

### 2.6.3 The Similarity Transfer Loss and the swelling topology

**Presentation.** In this section, we propose Similarity Transfer Loss (STL) which gives an implicit knowledge on the state of a sample point in order to better detect local anomalies in multidimensional binary time series. The capacity of an autoencoder is often used to decide the boundary between normal and abnormal points, but this is not the only constraint one can enforce on the NN. We claim that making the CNN autoencoder learn a higher data spatial representation with or without STL is also an option to control the definition of normality based on its reconstruction error and we show it for binary inputs CNNs without FC layers.

Dimensionality reduction can be a preprocessing for classification, data retrieval, data compression or other tasks, which explains the habit of decreasing the spatial dimensions of CNNs. Moreover, pooling layers, in most papers, only change the size of the spatial dimensions by an integer factor, which is not very convenient, especially for learning higher spatial dimensions representation. However, the increase of the spatial representation from the first layer, which we refer to as swelling topology, allows to have more feature units as well as to broaden the representational capacity of the layer while maintaining the number of parameters of the CNN. This way, a layer can have units with different sizes of cumulative receptive fields and a large spatial dimension to make the 1D-CNN learn temporal patterns of different scales.

Before going further, let us explain how the change of spatial dimensions with the Rational Sampling setup described in section 2.6.2 allows a CNN to have feature units with different size of receptive fields relative to the input, or cumulative receptive fields. Indeed, if the Rational Sampling layer is, for instance, a layer $Upsamp(3)$ followed by a layer $Donwsamp(2)$, a max pooling layer of pool size equal to 2, then the latter operates on the same value within a repetitive area of length 3 of $Upsamp(3)$, hence the receptive field of its output does not change as compared to its input, while it increases on the border separating two repetitive areas. Therefore, the number of receptive field sizes increases by one after each $RatioSamp(3, 2)$. This way, a 1D-CNN can focus on temporal patterns of different sizes just as a 2D-CNN on spatial patterns of different sizes. The more layers, the greater the size difference. This is not specific to Rational Sampling, it suffices to have an $Upsamp(n)$ followed by a layer with a kernel with size and stride such that the number of successive repetitive areas involved in the calculation of one output

unit varies from $k$ to $k+1$ while it slides [4].

Another reason why it has not been a widespread practice is that increasing spatial dimensions of hidden layers makes it hard for a CNN with a FC layer to learn useful patterns, most probably because of a greater number of the objective function local minima yielded by the distortion induced. Yet FC layers in CNNs are common. In our case we do not use FC layers and we increase the spatial dimension with Rational Sampling. The idea is similar to the one of OC-SVM where a higher dimension space [5] is seeked such that the boundary between normal points and abnormal points is easier to find thanks to the now called *blessing of dimensionality* from the paper [25] that proved that increasing the number of training features increases the probability to find linear separations between classes. Yet, an OC-SVM with a traditional kernel, does not take into account the long-term aspect of the data, which leads to a dramatically difficult parameters optimization, especially when the time series is composed of binary data. Indeed, in a multidimensional time series, two $m \times t$ time windows on $m$ variables can be considered similar if the first has the same variables than the second modulo some translations through time for some variables and yet appear very dissimilar according to the euclidean distance.

While an OC-SVM built on an euclidean metric will have trouble to find commonality within such a dataset, conversely, a CNN autoencoder will struggle to decide which point is abnormal facing «time similar» data points. Indeed, a CNN layer $L$ is by definition translation equivariant, that is, if the $m$ variables of a point $x_1$ are translated, by a same quantity and in a same direction, compared to another point $x_2$, so is the image of $x_1$ with respect to the image of $x_2$ under $L$. But the CNN itself is made partially translation invariant thanks to pooling layers and neither totally translation equivariant nor totally translation invariant is appropriate for time series anomaly detection, so one would like to be able to fix a balance, best for one's problem, between these two extremes. Of course, the variability of a multidimensional time series does not manifest only through

4. One could design a succession of convolutional and Rational Sampling layers with topological parameters so that the cumulative receptive field size on one layer ranges from the kernel size $ks$ of the first convolutional layer to the number of preceding layers plus $ks$. Hence one could build a CNN that focus on substantially different region sizes in the input space. This is interesting for CNNs with a spatial pyramid pooling [64] or without FC layers which can learn on sample points of different scales and save the model image resizing that alters its learning.

5. The code space has no higher dimensional representation if the number of channel is decreased enough before the code, and it is the case with the autoencoders in our experiments. Anyhow, our point is that the representation dimension of the latent space shall be far more larger than the intrinsic dimension of the input space when one increases the spatial dimensions in comparison to the case where one reduces it, hence the comparison.

translations but also through change of an individual variable in a particular point of time, and, especially for industrial systems, through correlated variable-wise translations and delays, let us call all of these, multivariate edits. If similar patterns, in term of multivariate edit, commonly appear on a sample of a state (Definition 2.2) during the normal regime, one will consider that the sample points are similar and normal, but if a sample point presents a pattern that can not appear during the normal regime, one would rather consider that it is an anomaly even if it is similar, in term of multivariate edit, to normal points. Roughly speaking, the generalization power will tend to make a traditional CNN autoencoder classify the aforesaid sample point as normal when it has too much capacity, inducing a low TPR or otherwise, hardly learn to reconstruct even normal points inducing a low True Negative Rate (TNR). The capacity alone is thus not sufficient to define normality of binary time series. Rather, we would like to control the generalization power of the CNN on the long-term and punctual patterns, keeping in mind that, for given capacity and topology, significantly improving one will lead to worsening the other as with TCL introduced in the previous section we provide a better detection rate for long-term anomalies. Can we do the opposite, that is improve the detection rate for punctual anomalies?

One could seek to formally define a computationally efficient multivariate edit distance on binary time series analogously to the Levenshtein distance in order to use it in an OC-SVM, for example, and therefore dispose of a model that takes into account the time relation between the variables. [6] Or one could let a 1D-CNN do the work for oneself with suited losses and architectures. The definition and the use of STL means to seize the best of two worlds: an euclidean measure, namely the cosine similarity, within STL, and a partially translation invariant method, that is the combination of convolutional and pooling layers, will compensate each other to find a happy medium, offering a trade-off between long-term anomaly and punctual anomaly detection. We choose the cosine similarity as the euclidean measure, because, above the fact that similarity is intrinsically tied with normality, the curse of dimensionality affects far less the variance in the angles between the difference vectors of a point to other points than the distance between two points as proven in [86]. Hence, it is a reasonable choice to transfer the cosine similarity levels between sample points from the input space to a latent space of the NN to facilitate

6. This option is not trivial since the number of comparison between two variables increases quadratically with the dimension of the time series and the comparisons themselves shall be expensive and grows with the length of the time windows.

higher spatial dimensions representation learning.

This loss, to be applied on a layer $l$ of a network $N$, depends on a function $Sim$ and the cosine similarity between inputs $x$ and $x'$ encoded by $N$ truncated at layer $l$, $N_l$:

$$\mathcal{L}_{ST}(Sim) = (\frac{\langle N_l(x), N_l(x')\rangle}{\|N_l(x)\|_2\|N_l(x')\|_2} - Sim(x, x'))^2 \tag{2.4}$$

In the following, we will only consider $Sim$ equal to the cosine similarity, i.e. STL is, here, only used to preserve the cosine similarity between sample points once encoded. Unlike the pairwise cosine loss [15], the image of $Sim$ is not necessarily reduce to $\{0, 1\}$, and hence the loss can be efficiently applied on convolutional layers. Another utilization of this loss would be to transfer the similarity of a hidden layer $l$ of a frozen network $FN$ – $Sim$ would then be the cosine similarity between data points encoded by $FN$ truncated at $l$, $FN_l$ – to a second NN trained on another dataset to ease its learning and improve its generalization. STL is applied on a hidden layer of the NN considered as the last layer of a siamese NN [11], so when applied on the code of an autoencoder, the encoder is duplicated into a siamese network.

It is worth repeating that STL alone is not enough to achieve the goal of taking into account punctual patterns within binary time series, it has to be helped by a topology that structurally disposes of the potential to do so. Yet, increasing spatial dimension provides redundancy and, when done with with the Rational Sampling, feature units with different cumulative receptive filed size within a single layer which proves to be appropriate topology properties to enable STL to make a trade-off between focusing on punctual or long-term patterns. Whereas the swelling topology itself can help to improve the detection of both long-term and punctual anomalies, there is one particular kind of anomaly for which this does not hold, that are anomalies coming from replay attacks for the reasons explained in the analysis of section 2.5. So, the swelling topology offers a trade-off between replay attacks detection and other attacks detection.

**Analysis.** The swelling topology can help detection of both global and local anomalies as showed in Figure 2.14, but fails to improve the performance of a CNN on replay attacks detection and even worsens it. We already discussed why CNNs without a FC layer are by their structure not suitable for detecting this kind of anomaly especially for time series with little variation like the one of near-deterministic systems in analysis of section 2.5.

We can see that the variance of ROC curves from the reconstruction error of CNN

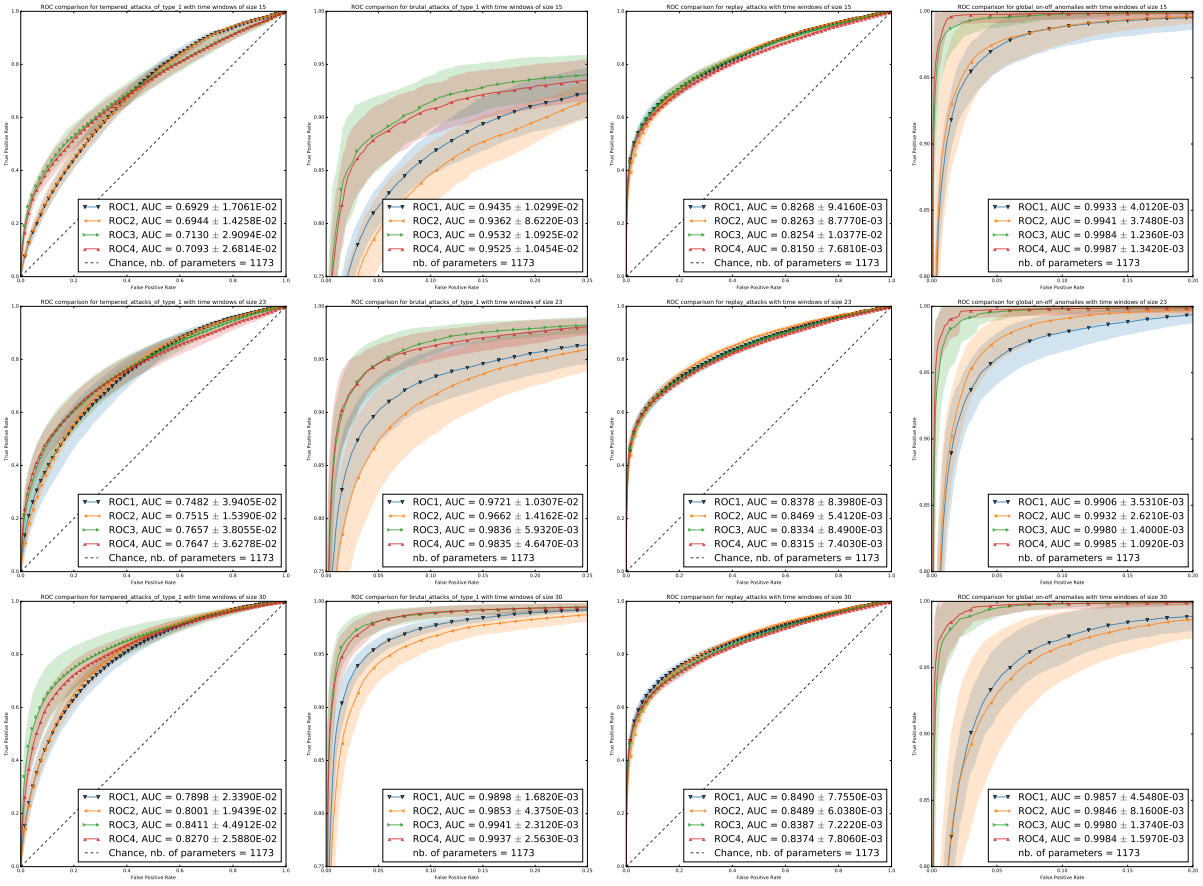



Figure 2.14 – Evolution of average ROC curves from reconstruction error of CNN autoencoders without FC layers with the same number of weights and with a code that has a spatial dimension of two-thirds the size of the input (ROC2, type v)($x$)), of half of the size of the input (ROC1, type i)($x$)), and of three halves of the size of the input (type vi)($x$)) trained with STL (ROC4) and without (ROC3), with $x = 15$ in the first row, $x = 23$ in the second row and $x = 30$ in the third. All autoencoders trained on 5 states with strides around 10. From left to right: the temperered attacks of type 1), the brutal attacks of type 1), the replay attacks and the global on-off anomalies.

autoencoders trained with the swelling topology, and more generally with Rational Sampling layers (ROC3, ROC2 in Figure 2.14) is often greater than the one of a classical autoencoder of the same capacity, which allows us to say that the increase of the spatial dimension itself is not the cause of a harder time learning but only the fact of having units of different cumulative receptive field size within a single layer. However, to be sure that the advantages of the swelling topology does not come from the side effect of having feature units with different size of cumulative receptive fields within a single layer, we consider a CNNs with a code layer of spatial dimension 2/3 the one of the input layer which is not enough to improve the detection of any anomaly (cf. Figure 2.14). [7]

Applying the STL on a CNN swelling topology helps to better detect local anomalies in binary time series by generalizing on global patterns, so it comes with the downside of a lower detection performance on global anomalies (see Figure 2.14; ROC4 versus ROC3). Hence the STL benefit from the inexorable conflict between learning long-term and punctual patterns, allowing a security expert working on time series to do a trade-off between the detection rate of long-term and punctual anomalies, like TCL but in the opposite direction. It is worth noticing the learning of a CNN with STL is, in the same vein as having single layer with feature units with receptive field of different sizes, about learning long-term patterns of different order of magnitude.

In the previous section, we showed that normalizing the anomaly score according to the state can itself be a tool to make a trade-off between the detection of subtle and brutal attacks, as long as the states have similar variabilities. The key idea is that NNs behavior far from the distribution underlying the training set is unpredictable [138], and thus the errors of reconstruction of the inputs of the same state are better ordered for sample points near its normal representation than far from it. However, we only experiment this idea in the context of CNN autoencoders with at least one FC layer. We found that this does not hold when we use the reconstruction error of a CNN autoencoder without FC layers. This suggests that, as expected, the unpredictable behavior of NNs on out-of-distribution inputs stems from FC layers much more than from convolutional layers.

7. It should be noted that deploying a residual topology implies taking into account units receptive fields of different sizes for the calculation of a single unit too, but unlike our setup, it does not produce units of different receptive field sizes within a same layer and above all, it requires more parameters except for applying an identity mapping as in [63].

## 2.7 Conclusion

In this chapter, we introduced a new concept of state in industrial systems that machine learnig model can use to benefit from their near-determinism. This concept is different from the one of state space representation models used in safety engineering. We showed its legitimacy thanks to our industrial system process simulator and also thanks to real data from SWaT. The model hinges on an autoencoder which gives an anomaly score from its reconstuction errors. The NN can be further tuned using new losses we introduced to improve the global detection performance and to prioritize the detection of some types of anomaly against other types. The first loss, TCL, is used on a FC layer so that the NN can better generalize on punctual anomalies and so better detect long-term anomalies. On the contrary, the second loss, STL, is applied on a convolutional layer to better detect punctual anomalies. In both cases, Rational Sampling plays an important part as it allows to better control the representational capacity of a convolutional layer.

Chapter 3

# Traceability of physical processes from industrial systems

The security of industrial systems implies of course online monitoring and therefore, solutions like the one we gave in the previous chapter, but another important component is digital forensics, which is the analysis of data storage devices for the purpose of proving or explaining crimes. In our context of physical processes, the questions are whether the raw data from the physical processes, or the transformed data, indeed originate from the ICS and, if it is transformed data, whether every transformation is the expected one, in brief we want to achieve traceability. The first question is important not only for digital forensics but also to ensure that monitoring solutions are constructed on data that are not compromised. Answering the second question allows identifying the transformation processes that have been compromised and better explain attacks. For example, if an insider attacker, that is a attacker who has a physical access to the system or to its security mechanism, manages to replace the monitoring device to hide a future long-term attack, we would like to be able to detect this action by means of the data that come from the monitoring device. We will pursue this example to expose our method for traceability and will refer to this by the question «Is it our IDS ?». Since we only have data, we need enough of them to achieve traceability, we propose to use hypothesis test. As we have multivariate data, the simplest way to do this is to use a summary of data points and a one-dimensional test to check that a new sample is from the same distribution than a sample securely stored. In our case, no assumptions can be made on the distribution, so we will use the well-known WMW test. It is worth noting that this distribution-free test has been extended to multivariate data by Paul R. Rosenbaum in 2005 [128]. To answer the question «Is it our IDS ?», we develop a DL model, the multipath NN, that is able to extract the characteristics of the monitoring system, though this model is not restricted to this case nor this use. Of course, traceability can also be useful for online monitoring to detect long-term attacks. In this chapter, we first present the WMW test and how to

use it for anomaly detection. Then, we present the multipath NN and show it provides a meaningful confidence measure along with its prediction. In our context, the multipath NN proves to be useful to characterize the monitoring system. Finally, we show how to use the WMW test and the multipath NN to achieve traceability.

## 3.1 Anomaly detection with the WMW test

The WMW test [153, 104] is a nonparametric distribution-free test that allows to check whether two samples $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_m$ are from the same distribution. The assumption of this test is that the observations from both samples are independent of each other. The null hypothesis $H_0$ is that the distributions are the same, and the alternative hypothesis $H_1$ considered herein is that the distributions are different. The statistic $U$ of this test is the minimum between the number of times a $Y$ precedes a $X$, $U_1$ and the number of times a $X$ precedes a $y$, $U_2$, in the ordered sequence composed of $X$'s and $Y$'s. That is $U = \min(U_1, U_2) = \min(U_1, nm - U_1)$ with

$$U_1 = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{1}_{X_i > Y_j} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{1}_{X_i = Y_j}$$

The test is consitent – i.e. the probability to reject $H_0$ when $H_1$ is true converges towards 1 when the $n, m \to +\infty$ – only if $\mathbb{P}(X > Y \mid H_1) \neq \mathbb{P}(Y > X \mid H_1)$. For small samples, one can see the critical values [1] for this test in Table B.4 from [24]. For large samples, i.e. samples of size greater than 20, the law of $U_1$ is known to be well approximated by a normal distribution [104] and a more practical formulation can be used: $U_1 = R_1 - \frac{n(n+1)}{2}$, where $R_1$ is the sum of the ranks of the $X$'s in the ordered sequence composed of $X$'s and $Y$'s. For equal values they are given the same rank which is the mean of their unadjusted ranks, for example the ranks of (1,6,6,10) are (1,2.5,2.5,4). Hence, one just has to check whether $\frac{U_1 - \mathbb{E}(U_1)}{\sqrt{\mathbb{V}(U_1)}}$ falls within the rejection region of a $\mathcal{N}(0,1)$ for a particular significance level, with $\mathbb{E}(U_1) = \frac{nm}{2}$ and $\mathbb{V}(U_1) = \frac{nm(n+m+1)}{12}$. For example, the rejection region for a significance level $\alpha = 0.05$ is $\mathbb{R} \setminus [-1.96, 1.96]$.

Now, let us describe how this test will be useful for anomaly detection in data coming

1. ... that are values of the statistic $U$ above which the null hypothesis is rejected for given significance level and sample sizes $n$ and $m$.

from physical processes of an industrial system. As mentioned above, we need a summary of the data on which this one-dimensional test can be performed. For example, the average error on the reconstruction of the input by an autoencoder as in the previous chapter is a natural choice. In the context of anomaly detection the assumption $\mathbb{P}(X > Y \mid H_1) \neq \mathbb{P}(Y > X \mid H_1)$ has to be watched closely since an attacker will try to produce $Y$ such that the previous equation does not hold so that its attack is undetectable.

One way to address this is to perform the test on a different summary obtained by secret transformation of the data. The principle would then be similar to the one of PUFs described at the end of the Problem statement. Instead of a CRP database kept secret to actively identify the physical object thanks to a challenge, one would use a secret transformation of the multi-dimensional data to passively identify the physical process of the system thanks to a WMW test on a summary of the transformed data.

Another simple solution to this problem is to perform the test both on the summary of the data and on non-monotonic transformations of it. The question is then which transformation? Since the idea is to detect whether two samples are from different distributions the natural transformation to consider is the probability density function itself. Indeed, let us say $X$ has a probability density function noted $f_X$, it is hard to find a random variable $Y$ with a different probability density function $f_Y$ such that both $\mathbb{P}(X > Y) = \mathbb{P}(Y > X)$ and $\mathbb{P}(f_X(X) > f_X(Y)) = \mathbb{P}(f_X(Y) > f_X(X))$ hold since the difference between $f_X$ and $f_Y$ will probably break the symmetry. And again, with $f_1$ the probability density function of $f_X(X)$, one can perform the WMW test on $X$, $f_X(X)$ and $f_1(f_X(X))\ldots$ Repeating this enough should lead to a consistent test even against malicious actors. Proposition 3.1 provides tools to compute the successive probability density functions in simple, yet most likely, cases. Indeed, if $(f_n)_{n>0}$ are recursively defined as being the probability density functions of $(f_{n-1}f_{n-2}((\ldots f_0(X))\ldots))_{n>0}$ with $f_0 = f_X$, then $(f_n)_n$ have most of the time a single mode and can be computed as in Proposition 3.1. The regularity condition ($\text{sign}(f'^2 - f.f'')$ is constant) is a reasonable properties for a probability density function since it prevents the function $f'$ to posses too abrupt changes. Yet, it is understandable that $(f_n)_n$ are more and more regular so that $\text{sign}(f_n'^2 - f_n.f_n'')$ tends to have less and less values while $n$ grows, except maybe for a particular initial function $f_X$. As a parenthesis, the exponential distributions, and consequently the Laplace distributions, are such that $f'^2 = f.f''$ except at their modes, hence Proposition 3.1 tells us that $f(X) \sim \mathcal{U}(0, \max(f))$, if $X$ is a random variable with probability density function $f$. If a distribution is deter-

mined by a probability density function verifying $f'^2 < f.f''$ that is a distribution with an extremely peak-shaped curve, $f(X)$ has a decreasing probability density function and it is increasing if $f'^2 > f.f''$. An example of distribution whose probability density function verifies $f'^2 > f.f''$ is the Gaussian distributions (cf. proof of Corollary 3.1).

**Proposition 3.1.** *Let $X$ be a real-valued random variable with a probability density function noted $f$ and $h$ be the probability density function of $f(X)$. We suppose that $f$ is twice differentiable, except maybe at the extremities of its support $S$ (i.e. $f(x) \neq 0 \Leftrightarrow x \in S$).*
*If $\forall x \in \overset{\circ}{S}, f'(x)^2 > f(x).f''(x)$, then $h$ is increasing on its support.*
*If $\forall x \in \overset{\circ}{S}, f'(x)^2 < f(x).f''(x)$, then $h$ is decreasing on its support.*
*If $\forall x \in \overset{\circ}{S}, f'(x)^2 = f(x).f''(x)$, then $h$ is constant on its support, so if the support is connected then $f(X) \sim \mathcal{U}(0, \max(f))$.*
*If $f$ has a single mode $m$ before which it increases and after which it decreases, then:*

$$a.e., h(x) = \left(\frac{x}{f'(f^{-1}_{|]-\infty,m[}(x))} - \frac{x}{f'(f^{-1}_{|]m,+\infty[}(x))}\right)\mathbb{1}_{x\in]\min(f),\max(f)[}$$

*with $f_{|I}$ the function $f$ restricted to the interval $I$ and a.e. meaning almost everywhere. If $f$ is increasing until $m$ after which it becomes null, then we have almost everywhere $h(x) = \frac{x}{f'(f^{-1}_{|]-\infty,m[}(x))}\mathbb{1}_{x\in]\min(f),\max(f)[}$ and if $f$ is increasing starting from $m$ before which it is null, then almost everywhere $h(x) = -\frac{x}{f'(f^{-1}_{|]m,+\infty[}(x))}\mathbb{1}_{x\in]\min(f),\max(f)[}$.*

*Proof of Proposition 3.1.* Let $X$ be a random variable with a twice differentiable probability density function noted $f$. Let us note $G$ the CDF of $f(X)$ and $h$ its probability density function. We have $G(z) = \mathbb{P}(f(X) < z) = \int \mathbb{1}_{f(x)<z} f(x)dx$.

Let us fix a $z$ in the interior $\overset{\circ}{I}$ of the support of $f$, $I$. If $f$ is monotonic on $I$, let us abusively note $f^{-1}$ instead of $f^{-1}_{|I}$, we can do the change of variable $u = f(x)$:

$$G(z) = \int \mathbb{1}_{f(x)<z} f(x)dx = \int \mathbb{1}_{u<z} u|f^{-1\prime}(u))|du = \pm \int \mathbb{1}_{u<z} \frac{u}{f'(f^{-1}(u))}du$$

If $z \in f(\overset{\circ}{I}), h(z) = G'(z) = \pm\frac{z}{f'(f^{-1}(z))}$, and of course, if $z \notin f(I)$, $h(z) = 0$. The sign depends on whether $f$ increases $(+)$ or deacreases $(-)$ on its support.

In the case where $f$ is not monotonic on its support, let us note $x_i$ the points such that $x_0 = -\infty$ and $\forall i > 0, f(x_i) = z, f'(x_i) \neq 0, x_i < x_{i+1}$. Then $\forall i, \exists I_i$ s.t. $f_{|I_i}$ is monotonic and $x_i \in \overset{\circ}{I_i}$ and then $G(z) = \sum_i \int_{x_{2i}}^{x_{2i+1}} f(x)dx$ and $G'(z) = \sum_{i>1} \text{sign}(f'(x_i))\frac{z}{f'(f^{-1}_{|I_i}(z))}$. Ac-

tually, if there is a $x$ such that $f(x) = z$ and $f'(x) = 0$ and increasing and decreasing sequences $(x_i)_i$ and $(x'_i)_i$ verifying $f(x_i) = z, f'(x_i) \neq 0, f(x'_i) = z, f'(x'_i) \neq 0$ and both converging towards $x$, one just has to do the same reasoning before and after $x$ so that $G(z) = \sum_i \int_{x_{2i}}^{x_{2i+1}} f(x)dx + \sum_i \int_{x'_{2i}}^{x'_{2i+1}} f(x)dx$. The same reasoning can be done with several $x$'s such that $f(x) = z$ and $f'(x) = 0$, but at the end, this is just a matter of notation, since every term of the sum is positive and thus the order has no importance. So, we can assume without loss of generality that there is no such $x$.

Yet, with $g_i : u \to \frac{u}{f'(f_{|I_i}^{-1}(u))}$, we have $g'_i(u) = \frac{1}{f'(f_{|I_i}^{-1}(u))} - \frac{uf''(f_{|I_i}^{-1}(u))}{f'(f_{|I_i}^{-1}(u))^3}$.

Finally, $g'_i(z) = g'_i(f(x_i)) = \frac{1}{f'(x_i)} - \frac{f(x_i)f''(x_i)}{f'(x_i)^3}$.

So, if $\forall i, \text{sign}(f'(x_i))g'_i(z) = \frac{1}{|f'(x_i)|} - \frac{f(x_i)f''(x_i)}{|f'(x_i)|^3}$ is greater than 0, we have

$$\sum_{i>1} \text{sign}(f'(x_i))g'_i(z) = G''(z) > 0 \text{ i.e. } h'(z) > 0$$

Therefore $f'^2 > f.f''$ is a sufficient condition for $f(X)$ to have an increasing probability density function on the support $]0, \max(f)]$, if the probability density function of $X$ is $f$. The other cases immediately follow. □

**Corollary 3.1** (of Proposition 3.1)**.** *Let $X \sim \mathcal{N}(\mu, \sigma^2)$ and note $\varphi_{\mu,\sigma^2}$ its probability density function: $\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2)$. Then, $\varphi_{\mu,\sigma^2}(X)$ has a probability density function that increases until its extremal point $\frac{1}{\sigma\sqrt{2\pi}}$ and that is null after this point.*

*Proof of Corollary 3.1.* We have $\varphi'_{\mu,\sigma^2}(x) = -\frac{x-\mu}{\sigma^2}\varphi_{\mu,\sigma^2}(x)$.

Therefore, $\varphi'_{\mu,\sigma^2}(x)^2 = \frac{1}{\sigma^2}(\frac{x-\mu}{\sigma})^2\varphi_{\mu,\sigma^2}(x)^2$ and $\varphi''_{\mu,\sigma^2}(x) = \varphi_{\mu,\sigma^2}(x)(\frac{1}{\sigma^2}(\frac{x-\mu}{\sigma})^2 - \frac{1}{\sigma^2})$.

Finally, $\varphi'_{\mu,\sigma^2}(x)^2 = \frac{1}{\sigma^2}(\frac{x-\mu}{\sigma})^2\varphi_{\mu,\sigma^2}(x)^2 > \varphi_{\mu,\sigma^2}(x)^2(\frac{1}{\sigma^2}(\frac{x-\mu}{\sigma})^2 - \frac{1}{\sigma^2}) = \varphi_{\mu,\sigma^2}(x)\varphi''_{\mu,\sigma^2}(x)$.

Then, thanks to Proposition 3.1, we conclude that $\varphi_{\mu,\sigma^2}(X)$ has a probability density function increasing until $\max(\varphi_{\mu,\sigma^2}) = \varphi_{\mu,\sigma^2}(\mu) = \frac{1}{\sigma\sqrt{2\pi}}$ and null after this point. □

This way, once the probability density function $f$ of $X$ has been approximated by $\hat{f}$, for example with a kernel density estimator, one should be able to derive analytically the probability density function of $\hat{f}(X)$, and so on, to perform the WMW tests. Further work needs to be done to know whether performing the WMW test on the probability density function applied on the random variables indeed lead to a series of tests that altogether make it impossible for an attacker to find a random variable $Y$ that is undetected.

In appendix B, we prove that the anti-function of $f$ is the fixed point of a functional under a simple condition on $f$ and provide an algorithm that converges towards $f^{-1}$ under two reasonable regularity conditions on $f$. It is important when the formulation of $f^{-1}$ is difficult to find, for example if $f$ is a monotonic part of a kernel density estimation.

## 3.2 Confidence in neural networks predictions

This section presents a model that is crucial to traceability of transformed information from a physical process. The use case will concern the authentication of the model monitoring the physical process, that is the autoencoder presented in the previous chapter. In order to verify that it is, in effect, the rightful autoeoncoder that is used to detect anomalies, we need a model that characterizes the autoencoder. The idea is to build a model that predicts the state of the system thanks to the reconstruction errors of the autoencoder and that provides a confidence measure along with he prediction. This way, if the autoencoder is maliciously replaced by another one from the attacker for data poisoning for instance [2], the distribution of the confidence measure is expected to change because the second autoencoder will have different kinds of error.

We propose herein an ad-hoc calibration method efficient even for some out-of-distribution samples, this method relies on a model we introduce, named multipath Neural Network (NN) and new strategy for DL models: Confidence Through Path Validation (CTPV), that is checking at the path of the information within a NN for estimating a measure of confidence in the trustworthiness of the input data.

Such a strategy is relevant for calibration on in-distribution data because activations of a group of neurons in hidden layers of a NN can be interpreted as the recognition of some patterns. So, if one forces these activations for given groups of neurons depending on the training input class, then for a new data point, one can assess the confidence in the prediction depending on whether the relevant neurons activate. This is equivalent to impose a path to the information within the NN, or to specialize a subpart, depending on the class.

2. Data poisoning is the fact to inject bad data into the trainig set. The attacker tries to inject bad data supposedly from the physical process, so that when a new monitoring model is built upon these data, it has some properties sought by the attacker to carry out an undetected very long term attack later on. To be sure that the current monitoring model does not detect these poisoned data and the attacker will temporarily replace it with another one that produces low anomaly scores for these poisoned data.

As for the issue of overestimation on out-of-distribution samples, the idea to specialize a subpart of the NN in a class is supported by the fact that the class with the most variance is likely to be the default choice for the prediction of a NN. Indeed, when a NN is trained to classify handwritten digit images as representing a 0 or a 2, random inputs from the uniform distribution within the tightest hypercube around the training set are most of the time classified as 2 by the NN. So, the NN has mostly learned patterns of 0 and predicts whether it is present.

We will see that the multipath Neural Network (NN) confidence measure is robust to changes of the discriminative features in the input and thus, when trained on the reconstruction error of an autoencoder to retrieve the state of the input given to the autoencoder, it allows a characterization of the autoencoder.

The architecture of our multipath NN is different from traditional NNs because of its penultimate layer composed of a new type of computational units, named Dissector Layer, as depicted in Figure 3.1. The definition of these units relies on the fact that, in a supervised classifier NN, «*the level of linear separability increases monotonically as we go to deeper layers*» [1]. These units aim at extracting features of interest and they must be able to separate linearly separable clusters to benefit from the aforesaid property. Let us consider the case where a NN layer achieves «perfect» linear separability of the clusters, i.e. each cluster is linearly separable from the union of the other clusters. A Dissector Layer can thus be supervisedly specialized in a class so that its outputs activate themselves all the more that the data point is far from a separating hyperplane in the half-space containing the class in question. This interpretation is supported by the fact that one can use Dissector Layers in an autoencoder to unsupervisedly seperate linearly seperable clusters. It acts as a filter that gives to the last layer, leading to the final decision, a revised version of the previous layer. The redundancy between the path of the information within the penultimate layer and the final decision enables a confidence in the prediction. However, the representation from a layer, even deep in the NN, might have a lower level of linear separability. If not all the clusters are separable from the union of the other clusters in the aforesaid representation, one could need more hyperplanes to isolate a cluster from the union of the other clusters and thus need more complex computational units. This is why we define Hyper-Neuron as a unit composed of Dissector Layer(s).

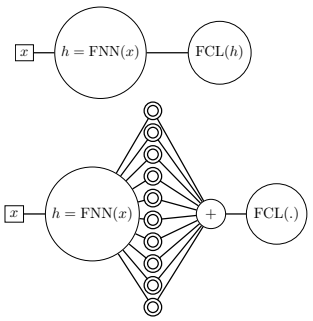



Figure 3.1 – Architecture difference between a traditional NN classifier (top) and a multipath NN classifier (down) with Diss-Layers (double circles) taking the same input $h$ and forwarding to the last layer the element-wise sum $\oplus$ of their outputs of the same size than $h$. FNN denotes a feed-forward neural network as the first part of the whole NN and FCL denotes a fully connected layer.

### 3.2.1 State of the art

The confidence one can have in the predictions of DL models is an important aspect in real-world environments, especially when critical decisions have to be made from these predictions. A good measure of confidence in a prediction can help to improve the decision in a subsequent action to balance risks. That is for example the case in [134] who aim at predicting failures of an autonomous vision based flying device. There exist different notions of what a good confidence measure should be. The most common is a calibrated probability estimate of the likelihood for the prediction to be true and it can be analyzed thanks to Expected Calibration Error (ECE). Basically, if a measure of confidence in a prediction equals $p$, then it is expected that this prediction is true with probability $p$. The confidence in a prediction issued from a deep Neural Network (NN) is known to suffer from miscalibration [62] and severe overestimation on out-of-distribution examples [119]. These issues are generally treated in the context of classification, but a recent work [80] demonstrates that one can calibrate NN Regressors confidence by shifting the problem onto the Regression-via-Classification framework. This work focuses on confidence issues in the case of classification.

Niculescu-Mizil et al. [120] studied Platt scaling as a post-hoc method with the aim of calibrating NNs. Then, Guo et al. [62] show how to effectively calibrate the confidence through temperature scaling, a special case of Platt scaling. In DL literature, temperature scaling first appeared as an efficient tool for knowledge distillation [13, 68]. This aims at imparting the ambiguity of the inputs to a small model thanks to the high temperature softmax predictions of a pre-trained large model viewed as soft labels. To address miscalibration, a temperature is found such that the Negative Log-Likelihood (NLL) is minimized on the validation set. To go further, Mozafari et al. [114] proposed an unsupervised version of temperature scaling. Seo et al. [137] calibrate the softmax prediction as

part of the learning with a loss that uses Bayesian uncertainty to decide to which extent the softmax prediction should be close to the uniform distribution. Lakshminarayanan et al. [87] use deep ensembles to estimate predictive uncertainty. Finally, Ovadia et al. [122] compare the robustness of the predictive uncertainty measure of many calibration methods under dataset shifts.

### 3.2.2 Dissector Layers and clustering

The key component of the multipath NN that enables us to characterize the errors of the monitoring model is the Dissector Layer thanks to which the confidence measure can be built.

Contrarily to miscalibration on in-distribution data, the issue of confidence overestimation on out-of-distribution data has been the subject of many methods. Only a few of these solutions participate in the learning phase. One can further differentiate them depending on whether the architecture of the NN is adapted to obtain a confidence measure. Gal et al. [43] estimate the Bayesian predictive uncertainty of a NN regularized with dropout, that is the variance of its prediction according to its learning parameters, by stochastic forward passes through the model. Papernotet al. [123] propose to inspect, in the layers of a Deep NNs, the nearest neighbors of a new input within an in-distribution set, to detect abnormal or ambiguous examples. Their method, DkNN, is inspired from the Conformal Prediction framework of Vovk et al. [148] and aims at answering similar questions to those we introduce to solve the aforesaid issues. Alternatively, a lot of researches focus on out-of-distribution detection thanks to the softmax prediction since the solution presented by Hendrycks et al. [67]. It relies on the fact that out-of-distribution examples produce lower softmax prediction max values than in-distribution examples. In order to make their confidence measure lower on out-of-distribution data Lee et al. [96] generate outliers thanks to a Generative Adversarial Network [55] and adapt the loss function so that the prediction for an adversarial example has a high entropy. In [97], Liang et al. add controlled perturbations to images in order to make in-distribution and out-of-distribution inputs more separable. At last, DeVries et al. [30] use the penultimate layer not only to feed the softmax layer but also to estimate the model confidence by the means of hints and penalties for out-of-distribution detection purposes.

As mentioned above, we want a computational unit that activates itself all the more

that the input is far from a separating hyperplane to benefit from the fact established by [1]. Therefore, a Dissector Layer (Diss-Layer) must extract features of interest so that a scoring function of its outputs can separate two linearly separable clusters, the simplest scoring function to consider is the sum. Hence, the sum of a Diss-Layer's activations have to be a substitute for the left-hand side of a hyperplane's equation: $\sum_i a_i.x_i = b$. It is known that in a supervised fashion this problem is solved by the perceptron [129, 110]. However, we want our model to learn a confidence in an implicit way, because our Diss-Layer has to «work» on its own within the multipath NN, so we need to tackle the unsupervised aspect of the separation of two linearly separable clusters. Thus, we relaxe the equation of a hyperplane: we define a substitute for the coefficient $a_i$. Of course, this substitute must not depend on $x_i$, but it can depend on $(x_j)_{j\neq i}$, the simplest and more computationally efficient way to do this is to define this substitute as a linear transformation of a weighted sum of $(x_j)_{j\neq i}$, that is $(\sum_{j\neq i} w_j.x_j).s_i + v_i$. So, a Diss-Layer is a computational unit that produces an output $y$ of the same size than the input $x$ with the following relation:

$$\begin{aligned} y_i &= g[((\sum_{j\neq i} w_j.x_j).s_i + v_i).x_i + b] \\ &= g[((P(x) - w_i.x_i).s_i + v_i).x_i + b] \end{aligned} \tag{3.1}$$

where $w$, $s$, $v$ and $b$, are learning parameters, $g$ the activation function and $P(x) = \sum_j w_j.x_j$. If one considers the identity as the activation function $g = Id$, then for each dimension $i$, the output $y_i$ is just a linear function of the input of the same dimension, $x_i$. In this case, the sum of the output of the Diss-Layer $\sum_i y_i$ is a quadratic substitute for the left-hand side of the equation of a hyperplane $\sum_i a_i x_i = b$, see appendix D for details about the regularization Diss-Layers must be subject to. The second formulation of (3.1) gives another intuition of the motivation for such a unit. Indeed $P(x)$ is the core element of our Diss-Layer concept. It acts as an action potential before the activation function, this is why we name it pre-potential. Let us recall that activation functions used in NNs, like the sigmoid function, try to mimic action potential of a neuron. The interest of such pre-potential function is that it can capture patterns in $x$ in an efficient way, since $w$ are shared across the outputs. Morerover, if several Diss-Layers are applied to the same input $x$, with distinct pre-potentials $(P_k(x))_k$, then the Diss-Layers can each capture different patterns in $x$. In the sequel, we express a Diss-Layer as a function of $x$ and denote it $l_k$ (or as a function of $x$ and $P_k(x)$: $y^k = f_k(x, P_k(x))$ in the appendix). In what follows, we show that Diss-Layers can be used in an autoencoder to unsupervisedly separate two

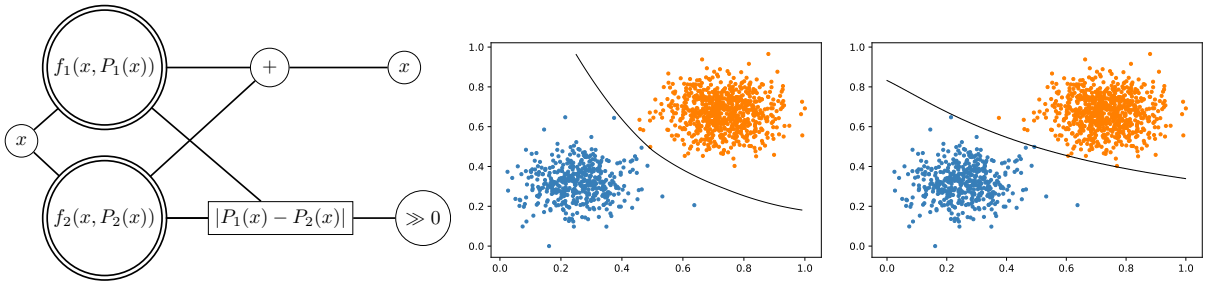



Figure 3.2 – Left: Scheme of the main objectives of a Hyper-Neuron autoencoder composed of two Diss-Layers to classify points from two linearly separable clusters in an unsupervised fashion thanks to $\sum_i f_1(x, P_1(x))_i = t_1$ and $\sum_i f_2(x, P_2(x))_i = t_2$, with $t_k$ fixed after the training phase such that the two boundaries agree on the classification. Middle and right: A toy example in two dimensions; the training set and the boundaries induced by the two Diss-Layers of a Hyper-Neuron autoencoder, with 14 learning parameters, unsupervisedly trained to separate two clusters. See appendix D for a complete description of the objectives.

clusters or to separate several clusters in a semi-supervised fashion as long as the clusters are linearly separable. This, with the increase of level of linear separability through the layers of a supervised NN classifier demonstrated by [1], supports their use within the multipath NN.

**Unsupervised classification of two linearly separable clusters.** Herein, the problem is to find, without any label, a separation between two clusters, that are supposed linearly separable. To do so, we consider two Diss-Layers (3.1) that take the same input $x$ and whose the element-wise sum of their activations $y^{(1)} + y^{(2)}$ is meant to reconstruct $x$, i.e. $y^{(1)} + y^{(2)} \approx x$. In other words, these two Diss-Layers alone constitute an autoencoder trained as described in Figure 3.2. Once the autoencoder has been trained, thresholds $t_1$ and $t_2$ relating to the sum of the two Diss-Layers are fixed so that the two boundaries defined by the equations $\sum_i y_i^{(1)} = t_1$ and $\sum_i y_i^{(2)} = t_2$ disagree as little as possible on the classification. However, one can argue that one Diss-Layer can have no threshold that allows a perfect separation. To be sure this does not happen, we have to apply proper activity regularizer on the Diss-Layers, so that each Diss-Layer is given the same importance (cf. Section D.1). Moreover, the simple fact of being linearly separable is not enough to ensure that a good substitute can be found. As a worst case scenario, let us consider two classes artificially defined as the two side parts of a multidimensional gaussian truncated along a hyperplane. In such situation, it is unlikely that the substitute found by our method will be close to the only hyperplane that separates these classes, in fact, in this case, it is impossible to perform the classification without a supervision. Some sort

of distance between two distributions has to be found so that, according to this distance, one can claim to use this method with a chance of success. The more distanced the two distributions will be, the more likely our method will be successful. To see how much the separations agree on the classification, we fix a threshold $t$ for one of the Diss-Layers (of ouput denoted $y^{(1)}$) and compute the area under the Receiver Operating Characteristic (ROC) curve from the sum of activations of the second Diss-Layer against the classification given by $\sum_i y_i^{(1)} = t$, we repeat this for several $t$ and keep the one inducing the highest area under the ROC curve, call it $t_1$ and find $t_2$ such that $\sum_i y_i^{(1)} > t_1$ and $\sum_i y_i^{(2)} > t_2$ disagree as little as possible. From our experience, in the case of well separated clusters as in Figure 3.2, having the two Diss-Layers of a same autoencoder agreed on the classification is enough to perfectly separate the clusters and this state is quickly attainable (usually one try).

Furthermore, a second experiment depicted in Figure 3.3 which, this time, brings into play two overlapping clusters, reveals the legitimacy of our use of pre-potentials. Indeed, we see in Figure 3.3 that the two boundaries are complementary. This is because the sum of the two Diss-Layers is supposed to be close to the input of the NN and their pre-potentials are repelled one from another. So, when the two clusters are well separated, the ratio of their size and goods separations by the two Diss-Layers are likely to be found following the procedure of the former paragraph. For example, in Figure 3.2 only few points are not classified the same way. As a matter of fact, the boundaries corresponding to a single tested ratio are quite different. So, when one of the boundaries from the aforesaid pairs crosses one of the clusters, they will produce different classifications. Hence two boundaries will agree only when they do not cross any of the two clusters, i.e. when they "correctly" classify the points, modulo some ambiguous points, or when they produce only one class. This is an additional evidence that when two distributions are "far" enough from each other, there exist thresholds such that corresponding boundaries will agree on the classification. In Figure 3.3, the similar ROC curves from the sums of the Diss-Layers activations, along with the boundaries' graph, show that the Diss-Layers capture different patterns in the input and are influenced by the centroids even when they are close.

Finally, to be sure that this method is scalable, we repeat the experiment of unsupervised clustering on two classes of 1000-dimensional data points. To do so, we generate examples from two isotropic gaussian distributions respectively centered in $0_{\mathbb{R}^{10^3}}$ and $1_{\mathbb{R}^{10^3}}$ both with standard deviation equal to $\sqrt{1000}/7$, then we apply a min-max normalization.

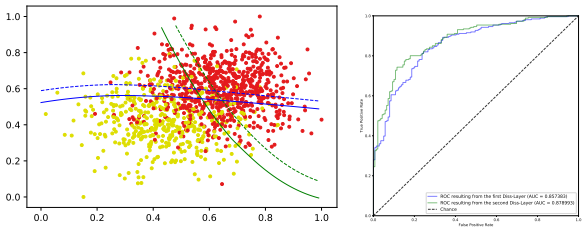


Figure 3.3 – Left: A toy example in two dimensions; the training set is composed of two overlapping clusters (the lower left contains 38% of the training set), the boundaries $\sum_i f_k(x, P_k(x))_i = t_k$ induced by the two Diss-Layers of a Hyper-Neuron autoencoder, with 14 learning parameters, with $t_k$ fixed so that the boundaries split the training set with the proportions 38-62% (dotted lines) and 50-50%. Right: The ROC curves derived from the sums $\sum_i f_k(x, P_k(x))_i$ $k = 1, 2$.

This way, the clusters are far enough as in Figure 3.2, but none of the pairs of dimension is enough to discriminate the examples. The autoencoder has 6002 learning parameters shared among the two Diss-Layers. We obtain an accuracy of 99.8% both in the training set – used for the learning phase of the autoencoder and the threshold setup – and the test set. Again, for data with 1000 dimensions, we consider a training set with one class that contains 17% of the whole set and through the same experiment we obtained accuracies equal to 98.4% and 98% respectively on the training set and the test set. So our method is also suitable for unbalanced dataset.

**Semi-supervised classification of linearly separable clusters.** In the previous section, a Hyper-Neuron with two Diss-Layers is used to separate two clusters in an unsupervised fashion. In the case where there is more than two classes, hyperplanes can still be helpful if clusters are linearly separated from each other. However, the minimal number of hyperplanes to separate several clusters can be much higher than the number of clusters. Moreover, it is not clear what this number is and how to create, thanks to Diss-Layers and appropriate thresholds, substitutes of hyperplane that would help discriminating data points. Again some sort of measure is needed to determine the difficulty to find hyperplanes useful to the classification. A first step toward such a measure is presented in appendix C. This allows us to generalize the definition of linear separablility of two sets to any number of sets. However, we show in Figure 3.4, a simple example with four clusters where our method is useful. In this experiment, the Hyper-Neuron autoencoder output is the sum of four Diss-Layers, instead of two like in Figure 3.2. Moreover, the data have two dimensions so that we can search on the plot for proper thresholds to partition the space with the aim of separating the clusters thanks to the sum of the Diss-Layers activations. If there is too many Diss-Layers as in Figure 3.4, some separations are

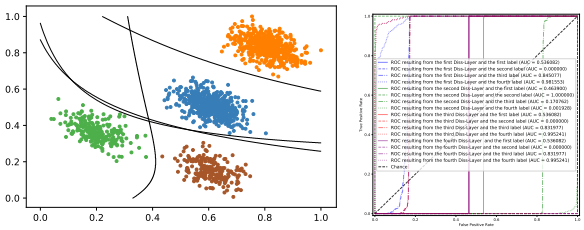


Figure 3.4 – Left: A toy example in two dimensions; the training set is composed of four clusters (with 530, 520, 231 and 219 examples), the boundaries $\sum_i f_k(x, P_k(x))_i = t_k$ induced by the four Diss-Layers of a Hyper-Neuron autoencoder, with 28 learning parameters, with $t_k$ manually fixed to separate the clusters. Right: The ROC curves derived from the sums $\sum_i f_k(x, P_k(x))_i$; $k$ = 1,2,3,4 against the labels in the basis one-vs.-all.

redundant. This allows us to claim that one can define an algorithm to search for the best boundaries from the Diss-Layers by means of a partially labeled dataset.

### 3.2.3 Hyper-Neuron and Confidence Through Path Validation

**The multipath NN classifier.** The goal of the multipath NN is to provide a confidence measure along with its prediction that was implicilty learned during the training phase thanks to strategy CTPV. We propose to fullfill this strategy through two properties that our measure will have to verify. The first is the ability to assess how much the information drawn from the input contribute to the prediction of a class. And the second is the ability to quantify to which extent it is similar to the patterns from the training examples. We define the multipath NN as a NN with at least one of its layers composed of Diss-Layers that receive the same input. In the rest of the paper, multipath NNs will implicitly refer to multipath NN classifiers. A layer is a set of real-valued functions with common inputs.

Let us consider the simplest case where it is possible that, in the representation from a layer deep enough in a NN classifier, every clusters are linearly separated from the union of the other clusters. Then one can deploy a multipath NN with as many Diss-Layers as classes so that each Diss-Layer is specialized in a class as depicted in Figure 3.5. Every Diss-Layer takes the same input that is the ouput from the previous layer as in the left scheme of Figure 3.6 and, at testing time, their element-wise sum is given to the next layer as in the right scheme of Figure 3.1, but at training time this is different. The specialization is done thanks to two mechanisms depending on whether the input of the network is a meaningful labeled example or a random input. The idea behind the use

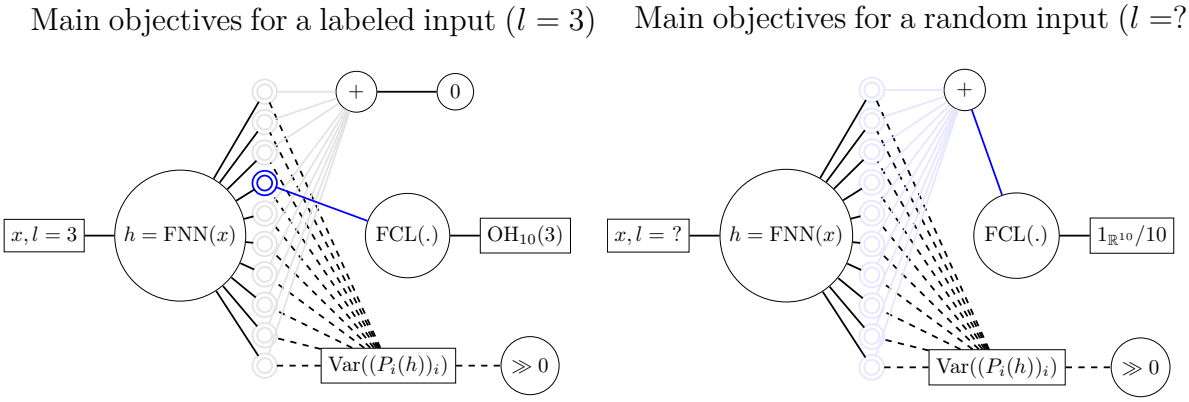



Figure 3.5 – Multipath NN during training. Scheme of the main objectives of a multipath NN classifier with 10 Diss-Layers (double circles) each for one class which receive the same input $h$; $(x, l =?)$ stands for random input (i.e. input from the uniform distribution in the tightest hypercube surrounding the training set, thus no label $l$), FNN denotes a feedforward neural network as the first part of the whole NN, FCL denotes a fully connected layer $\oplus$ denotes the element-wise sum, $\text{OH}_{10}$ denotes the one-hot encoding for 10 classes and $\text{Var}((P_i(h))_i)$ denotes the variance of the pre-potentials of the Diss-Layers. Find the confidence for this NN in (3.2) (general framework: (3.3)).

of random inputs is that we want no class to be a default class like classical NNs that ususally have a default class which is the class with the most variance (cf. section 3.2 third §). When the input has a label $j$, only the Diss-Layer specialized in the class $j$ will participate to the element-wise sum given to the last layer(s) (cf. FCL in Figure 3.5) and the other Diss-Layers are regularized by making their element-wise sum close to the null vector. This mechanism provides redundancy that enables the construction of a confidence measure. When the input is random (generated by the uniform distribution in the tightest hypercube of the training set in our case), the element-wise sum of all the Diss-Layers are forwarded to the last layer(s). In any case, at training time, the variance of the pre-potentials of every Diss-Layer is moved away from zero so that the Diss-Layers extract different patterns from the examples and none are redundant with another (cf. $R$ in (3.5)).

The confidence of a multipath NN classifier $N$ must ascertain the role of the Diss-Layer specialized in the prediction of the class $\hat{y} = \text{argmax}(L(\sum_k l_k(F(x))))$ for an input $x$, with one layer composed of $K$ Diss-Layers $(l_k)_k$ taking the same input $h = F(x)$, with $F$ the first part of $N$, and feeding its output to the last layer(s) $L$. The confidence $\mathcal{C}$ in the prediction $\hat{y}$ from $N$ for the input $x$ is described in (3.2). This confidence measure follows the strategy CTPV thanks to the factor on the right of the multiplication. Indeed, a multipath NN classifier trained as described in Figure 3.5 will produce $L(l_{\hat{y}}(F(x)))$ and $L(\sum_i l_i(F(x)))$

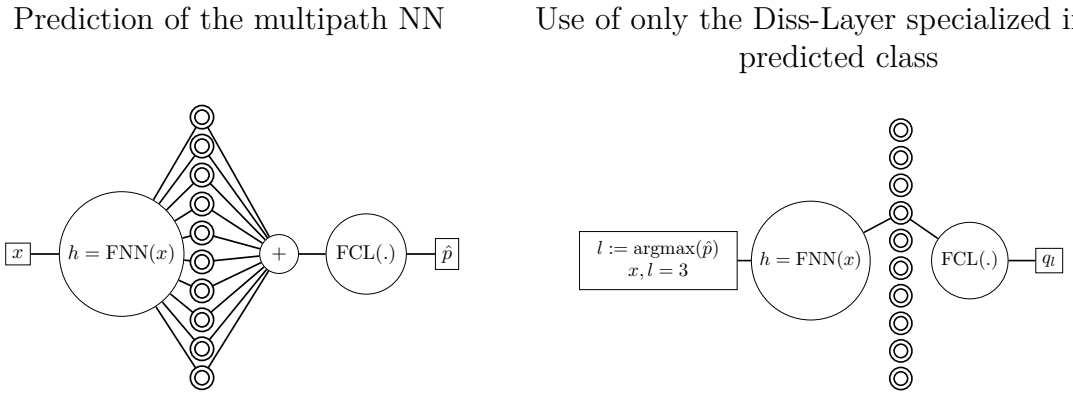



Figure 3.6 – Multipath NN during testing. Scheme of the prediction and the confidence measure computation of a multipath NN classifier with 10 Diss-Layers (double circles) each for one class which receive the same input $h$; FNN denotes a feedforward neural network as the first part of the whole NN, FCL denotes a fully connected layer, $\oplus$ denotes the element-wise sum. Find the confidence for this NN in (3.2) (general framework: (3.3))

that are likely to be close from each other for an input $x$ drawn from the distribution underlying the labeled training set. In other words, comparing these terms reveals the divergence between the ideal path and the actual one within the multipath NN. The first factor in (3.2) aims at verifying how much the information drawn from the input contribute to the prediction of a class. The second factor aims at quantifying to which extent this information is similar to patterns, from the training sample, used to predict the same class. Both factors naturally involve the aleatoric uncertainty. The confidence measure of the multipath NN is implicitly learned, so it is never computed during the training phase. Figure 3.6 shows how the confidence measure of the multipath NN is computed during the testing phase according to the following equation, with $\|.\|_\infty : x \to \max_i(|x_i|)$:

$$\mathcal{C}(N)(x) = L(\sum_k l_k(h))_{\hat{y}} \times (1 - \|L(l_{\hat{y}}(h)) - L(\sum_k l_k(h))\|_\infty) \tag{3.2}$$

In more complex cases, some clusters, in the representation from a layer deep enough in a NN classifier, cannot be separated from the union of the other clusters. Therefore, a more complex computational unit is needed: we define Hyper-Neurons as the computational units resulting from the element-wise sum of Diss-Layer(s) on which is applied an activation function. As for the multipath NN, the mechanisms are the same as before,

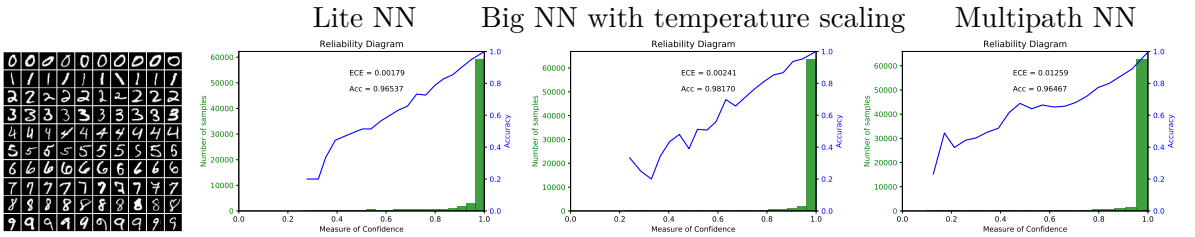



Figure 3.7 – Reliability diagrams for measures of confidence on an augmented dataset from the MNIST test set. From left to right: 1) Subset of the MNIST dataset 2) Confidence measure obtained by taking the maximum value from the softmax output of a lite NN classifier 3) Confidence measure obtained by taking the maximum value from the softmax output of a big NN classifier calibrated with temperature scaling 4) Confidence measure from (3.2). NNs trained on the MNIST training set.

but Hyper-Neurons are specialized in a class instead of just one Diss-Layer per class. Let us examine in more detail how they can be useful.

**Multipath NN based on specialized hyper-neurons with several Diss-Layers.** Herein, we introduce the role of Hyper-Neurons in the multipath NN. Let us recall that a Hyper-Neuron is a computational unit resulting from the element-wise sum of Diss-Layer(s) on which is applied an activation function. From this, one can consider in a NN, a layer that is composed of $K$ Diss-Layers taking the same input. In this layer, there exist $2^K$ Hyper-Neurons, that is the number of subsets from the set of Diss-Layers. The path of the information depends on the Hyper-Neuron used to process the input. Here, the idea is the same as in the previous section, but Hyper-Neuron which is more general and complex than Diss-Layer takes its place as a computational unit assigned to a class. That is to say some Hyper-Neurons specialize in a class by giving to the first layer after the Diss-Layers the output of one Hyper-Neuron depending on the label of the input or absence of label. If the input has a label, the Hyper-Neuron we use is the one assigned to the input class. If the input is randomly generated, and thus has no label, the Hyper-Neuron at issue is the one with every Diss-Layer. When the input has a label, the Hyper-Neuron composed of each Diss-Layer that is not part of the Hyper-Neuron specialized in the input class is silenced by making its output close to the null vector. All this follows the same reasoning than in the previous section but with specialized Hyper-Neurons that have more than one Diss-Layer, more complex classification problems can be handled. This claim is supported by the fact that the Hyper-Neuron autoencoder in Section 3.2.2 can serve to separate several clusters in a semi-supervised fashion. Section D gives more details on the multipath NN implementation. In our experiment, a specialized Hyper-Neuron has only

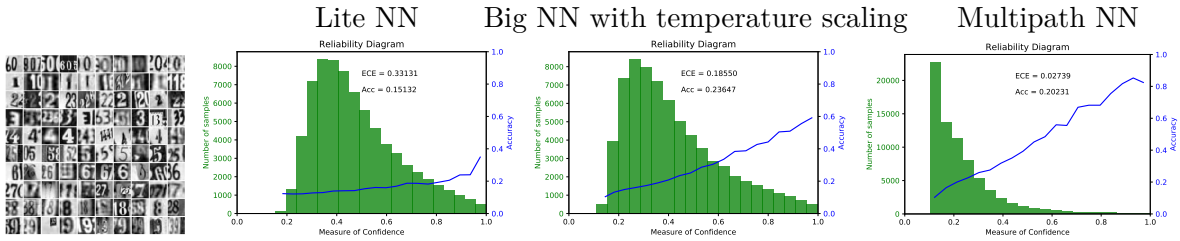



Figure 3.8 – Reliability diagrams for measures of confidence on the black and white SVHN set. From left to right: 1) Subset of the SVHN dataset 2) Confidence measure obtained by taking the maximum value from the softmax output of a lite NN classifier 3) Confidence measure obtained by taking the maximum value from the softmax output of a big NN classifier calibrated with temperature scaling 4) Confidence measure from (3.2). NNs trained on the MNIST training set.

one Diss-Layer, so a Diss-Layer is involved in a single class, but the use of a multipath NN is not restricted to this scheme. Therefore, one should keep in mind that it is preferable that Diss-Layers involved in several classes are part of Hyper-Neurons that specialize in classes with commonalities. Moreover, specialized Hyper-Neurons composed of a lot of Diss-Layers should concern complex classes while the ones with few Diss-Layers should concern simple classes.

The confidence of a multipath NN classifier $N$ must ascertain the role of the Hyper-Neuron specialized in the prediction of the class $\hat{y} = \text{argmax}(L(\sum_k l_k(h)))$ for an input $x$ with one layer composed of $K$ Diss-Layers $(l_k)_k$. The confidence $\mathcal{C}$ in the prediction $\hat{y}$ is:

$$\mathcal{C}(N)(x) = L(\sum_k l_k(h))_{\hat{y}} \times (1 - \|L(\sum_{j \in S_{\hat{y}}} l_j(h)) - L(\sum_k l_k(h))\|_\infty) \tag{3.3}$$

with $S_j \subset \{1, ..., K\}$ the indices of the Diss-Layer(s) of the Hyper-Neuron assigned to the class $j$ and the same notations as in the beginning of this section. The confidence defined in (3.2) is a special case of (3.3) and the latter also follows the strategy CTPV. We present the Hyper-Neuron for CTPV, although its application scope is surely much broader. One can think of achieving hierarchical structures, thanks to an adequate layout of such units, needed in Natural Language Processing (NLP), for example.

**Training objectives of a multipath NN classifier.** Now, let us describe the training objectives. The goal is to enforce the information to go through the right Diss-Layer(s) thanks to the two aforesaid mechanisms for random and labeled inputs. Let $C$ be the number of classes. Let $K$ be the number of Diss-Layers in the penultimate layer. Let $(S_l)_{1 \leq l \leq C+1} \in \mathscr{P}(\{1, ..., K\})^{C+1}$ the sets of indices of the Diss-Layers composing $C + 1$

Hyper-Neurons: the $C$ specialized Hyper-Neurons plus the Hyper-Neuron for random input, i.e. the Hyper-Neuron composed of each of the $K$ Diss-Layers. In the simplest case described in the previous section, we have $K = C$ and $S_l = \{l\}$. Actually, $2.C + 1$ Hyper-Neurons are employed during the learning phase but the $C$ silenced Hyper-Neurons are complementary to those specialized in a class, so $(S_l)_{1 \leq l \leq C+1}$ is enough for the equations that follow. Let $(x, y)$ be the couple of the input $x$ and the target $y$. For random input, the target will be $y = (\frac{1}{C})_{l \in \{1,...,C\}}$, for labeled input it is the regular one hot vector. $L$ is the last layer(s), let us call $Dl$ the layer, composed of Diss-Layers $l_k$, that feeds $L$. Each Diss-Layer $l_k$ takes the same input $h$, which is the output of the previous layer. The ouput of $Dl$ is computed as follows: $Dl(h) = \sum_{k \in S_y} l_k(h)$. We make a misuse of langage: the indice $y$ is not an integer but a $C$-uplet, for the sake of clarity we elude the correspondance between the indice of the Hyper-Neurons and the target of the classes they specialize in. We use the NLL loss, the last layer of $L$ has the softmax function and the main loss is:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c . \log(L(Dl(h))_c) \tag{3.4}$$

This is related to the loss proposed in equation (1) of [96]. However, we are not in their adversarial setup, so we only use the NLL loss for both labeled and random inputs. It is worth noticing that generating random inputs from the uniform distribution in the tightest hypercube surrounding the training set supposes that the dimension of the input is sufficiently large so that the probability of generating a random input close to the distribution underlying the training set is very close to zero. Finally, considering a mini-batch $(x_i)_{i \in I}$, and with $h_i$ the output of the first part of the NN that takes $x_i$ and feeds the Diss-Layers $(l_k)_k$, the total loss of the classifier multipath NN is:

$$\frac{1}{I} \sum_{i \in I} [\mathcal{L}(h_i) + \lambda_1 . \sum_j (\sum_{k \notin S_y} l_k(h_i)_j)^2] + \lambda_2 . R((h_i)_{i \in I}) \tag{3.5}$$

with $R$ defined in appendix D. The sum of the terms $(\sum_{k \notin S_y} l_k(h_i)_j)^2$ serves to annihilate the information from the Diss-Layers that are not part of the Hyper-Neuron specializing in the class of label $y$: the element-wise sum of the outputs of Diss-Layers not specialized in the class $y$ draws near to the null vector. As for $R$, it ensures that the Diss-Layers capture different patterns from the input. In our experiment, $\lambda_1 = \lambda_2 = 0.01$, and $I = 64$ ($I$ is of concern for $R$ cf. appendix D), the number of epochs is 5 (same batch size and number of epochs as for the classical NN) and the activation function of the Diss-Layers

is the rectified linear unit (ReLU) (and the identity for Hyper-Neurons). As well as for autoencoders with Diss-Layers, we use the Adam optimizer [82] of keras with the gradient descent's default parameters. In our experiment on the balanced training set MNIST, we generate as many random examples than the number of examples in each class.

**Comparison.** We compare the calibration of our models confidence measure against the measure of a big NN calibrated through Temperature scaling and the one of a lite NN naturally calibrated thanks to its small size. We also compare their ability to detect outliers. Details for reproducibility are in appendix D. More specifically, details on the datasets are in the appendix section D.2. Histograms in Figure 3.8 show that the multipath NN is more appropriate for outlier detection than a lite NN and a big NN calibrated through temperature scaling. Indeed, for each of the out-of-distribution samples, the histogram of our confidence measure shows it gives lower scores in average to out-of-distributions points than the other measures. To evaluate in- and out-of-distribution calibration, we use a new metric we call EoA (ECE over Accuracy): $ECE/Acc$, which aims to give more importance to calibration than to accuracy. Indeed, a model that is inaccurate on a new distribution but that has systematically a low confidence value is better than a model maybe two times more accurate but that fails to detect a great deal of the inputs that induce incorrect predictions and thus inappropriate decisions. Conversely, having a high accuracy is not useful if low confidence values prevent us to use the predictions. Not to mention that calibration is more difficult than accuracy: predicting a point in a continuous space, the confidence measure, versus in a discrete space, the class. Since we deal with stochastic models, we need to repeat the experiment several times to support results shown in Figures 3.7, 3.8 and 3.9. ECEs, accuracies and EoAs of 10 NNs of each of the three types are reported in Table 3.1. In brief, the average EoAs for the lite NN on the datasets MNIST, SVHN and Semeion equal 0.004, 2.56 and 0.18, respectively. For the big NN with TS they equal 0.002, 1.21 and 0.19 and for the multipath NN, they equal 0.02, 0.14 and 0.30. Further experiment detailed and analysed in appendix E show that, while temperature scaling on a big NN is the best method when facing no other changes than dataset mild shifts, when the dataset is not shifted but utterly changed, our measure, thanks to the redundancy made possible by Diss-Layers that extract meaningful patterns, is the more robust provided that the discriminative features are similar to

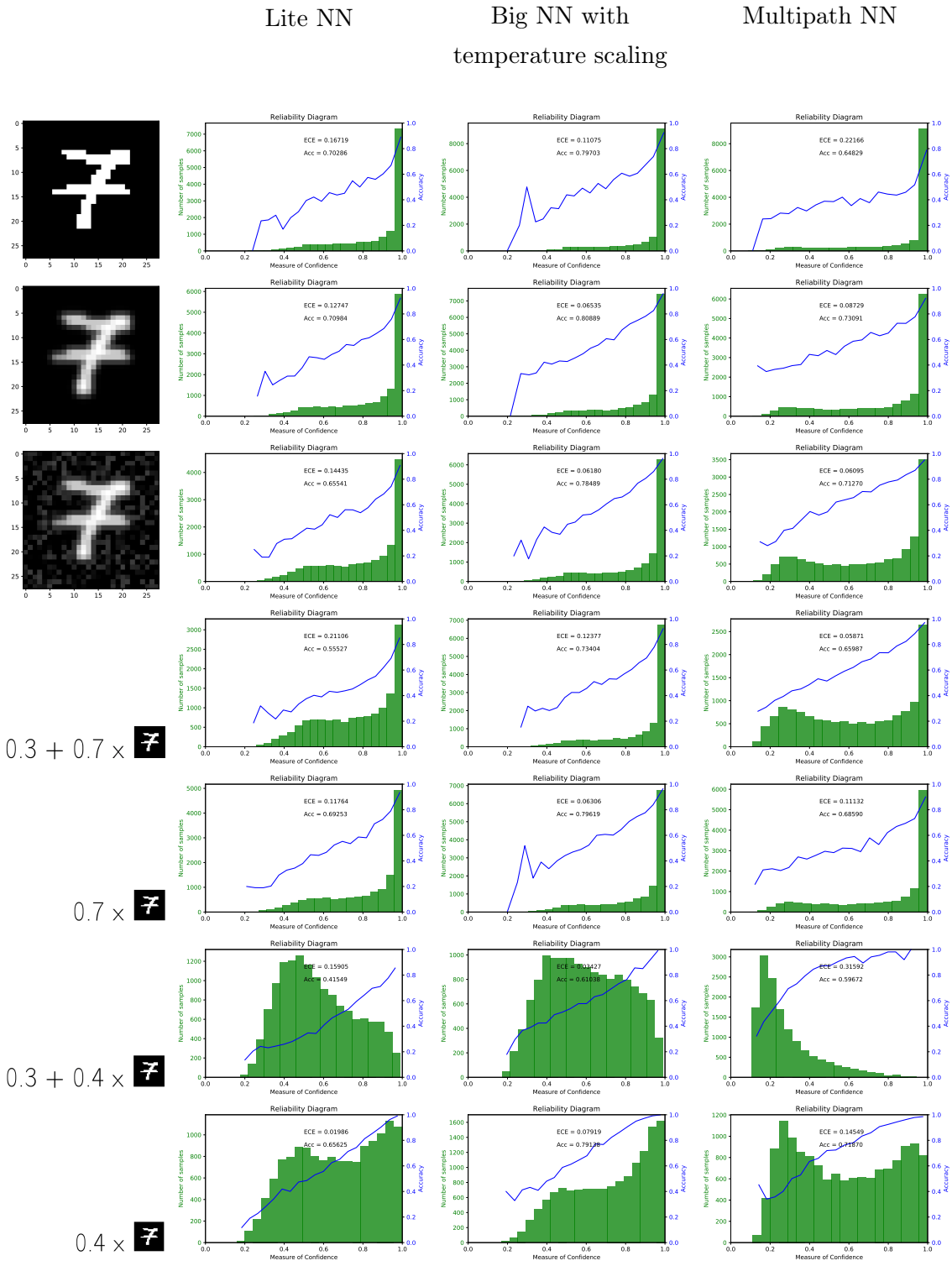



Figure 3.9 – Reliability diagrams for measures of confidence on data adapted (details in the appendix section D.2) from the Semeion Handwritten Digit Data Set. NNs trained on the MNIST training set.

Table 3.1 – Percentage of ECE (1 ↓), percentage of Accuracy (2 ↑) and EoA (3 ↓) for the 10 small NNs, the 10 big NNs with TS and the 10 multipath NNs (one per row), all trained on MNIST (averages in a separated row below, ↓: the lower the better, ↑: the higher the better).

| Small NNs trained on MNIST | | | | | | | | | Multipath NNs trained on MNIST | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MNIST | | | SVHN | | | Semeion | | | MNIST | | | SVHN | | | Semeion | | |
| 1 ↓ | 2 ↑ | 3 ↓ | 1 ↓ | 2 ↑ | 3 ↓ | 1 ↓ | 2 ↑ | 3 ↓ | 1 ↓ | 2 ↑ | 3 ↓ | 1 ↓ | 2 ↑ | 3 ↓ | 1 ↓ | 2 ↑ | 3 ↓ |
| 0.2 | 96.8 | 0.002 | 40.8 | 20.8 | 1.97 | 14.9 | 72.4 | 0.21 | 2.5 | 94.5 | 0.03 | 0.8 | 11.8 | 0.07 | 20.3 | 68.0 | 0.30 |
| 0.5 | 96.8 | 0.005 | 54.3 | 15.4 | 3.52 | 9.7 | 78.5 | 0.12 | 1.4 | 95.8 | 0.02 | 2.6 | 15.1 | 0.17 | 24.6 | 62.8 | 0.39 |
| 0.4 | 96.8 | 0.004 | 46.4 | 16.0 | 2.91 | 12.4 | 76.0 | 0.16 | 1.5 | 96.7 | 0.02 | 2.4 | 12.3 | 0.19 | 22.5 | 67.0 | 0.34 |
| 0.3 | 95.8 | 0.003 | 33.6 | 21.7 | 1.55 | 15.6 | 72.8 | 0.21 | 1.5 | 96.4 | 0.02 | 1.1 | 12.0 | 0.09 | 20.7 | 68.6 | 0.30 |
| 0.5 | 97.0 | 0.005 | 42.0 | 21.4 | 1.96 | 12.6 | 72.0 | 0.18 | 1.6 | 96.2 | 0.02 | 3.5 | 16.7 | 0.21 | 19.5 | 71.8 | 0.27 |
| 0.2 | 96.8 | 0.002 | 63.3 | 20.6 | 3.07 | 11.7 | 77.0 | 0.15 | 2.1 | 94.7 | 0.02 | 0.9 | 11.3 | 0.08 | 18.6 | 71.0 | 0.26 |
| 0.8 | 96.9 | 0.008 | 30.2 | 22.9 | 1.32 | 9.9 | 76.2 | 0.13 | 1.5 | 96.3 | 0.02 | 4.6 | 15.5 | 0.30 | 18.1 | 70.7 | 0.26 |
| 0.5 | 96.8 | 0.005 | 28.6 | 19.2 | 1.48 | 14.9 | 73.3 | 0.20 | 1.4 | 96.8 | 0.01 | 1.7 | 14.3 | 0.12 | 23.5 | 65.9 | 0.36 |
| 0.4 | 95.8 | 0.004 | 52.7 | 12.6 | 4.18 | 20.2 | 66.5 | 0.30 | 1.6 | 96.3 | 0.02 | 0.7 | 10.8 | 0.07 | 18.1 | 70.2 | 0.26 |
| 0.4 | 96.2 | 0.005 | 53.9 | 14.6 | 3.68 | 10.7 | 76.3 | 0.14 | 1.7 | 95.9 | 0.02 | 1.0 | 18.2 | 0.06 | 21.3 | 68.2 | 0.31 |
| 0.4 | 96.6 | 0.004 | 44.6 | 18.5 | 2.56 | 13.3 | 74.1 | 0.18 | 1.7 | 96.0 | 0.02 | 1.9 | 13.8 | 0.14 | 20.7 | 68.4 | 0.30 |

| Big NNs trained on MNIST | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| MNIST | | | SVHN | | | Semeion | | |
| 1 ↓ | 2 ↑ | 3 ↓ | 1 ↓ | 2 ↑ | 3 ↓ | 1 ↓ | 2 ↑ | 3 ↓ |
| 0.2 | 97.9 | 0.002 | 13.6 | 22.9 | 0.59 | 12.8 | 76.3 | 0.17 |
| 0.1 | 98.0 | 0.001 | 14.1 | 24.3 | 0.58 | 15.0 | 75.0 | 0.20 |
| 0.2 | 97.5 | 0.002 | 21.0 | 20.6 | 1.02 | 14.4 | 74.9 | 0.19 |
| 0.2 | 97.9 | 0.002 | 25.6 | 20.4 | 1.26 | 15.3 | 74.3 | 0.21 |
| 0.2 | 98.0 | 0.002 | 24.7 | 17.6 | 1.40 | 13.5 | 76.2 | 0.18 |
| 0.2 | 98.0 | 0.002 | 32.0 | 21.0 | 1.52 | 14.2 | 75.3 | 0.19 |
| 0.2 | 97.9 | 0.002 | 27.9 | 18.6 | 1.50 | 15.2 | 74.6 | 0.20 |
| 0.3 | 97.6 | 0.003 | 12.9 | 24.0 | 0.54 | 13.5 | 74.6 | 0.18 |
| 0.5 | 97.3 | 0.006 | 23.7 | 21.0 | 1.12 | 16.3 | 72.8 | 0.22 |
| 0.1 | 97.4 | 0.001 | 32.0 | 12.7 | 2.52 | 13.3 | 75.2 | 0.18 |
| 0.2 | 97.8 | 0.002 | 22.7 | 20.3 | 1.21 | 14.3 | 74.9 | 0.19 |

those of the training set (Figure 3.8). Based on this property, we propose in the appendix, Section E.2, a line of research involving Topological Data Analysis (TDA) tools to arrive at a model that is calibrated for any input. Details for reproducibility are in appendix D.

## 3.3 Setup of the experiments

In order to show that the WMW test is a useful method to achieve traceability for data from a physical process, we conduct two experiments.

The first experiment consists in verifying that the data supposedly from the industrial system indeed originate from it. In this experiment we compare two methods thanks to ROC curves: 1) we plot the ROC curves from the opposite of the p-values [3] from the WMW test on the average reconstruction errors of the autoencoder, or 1 minus the p-value if one wants to consider probabilities 2) we plot the ROC curves from the maximum anomly score in the sample, that is the maximum average reconstruction errors. The second method considers that a sample is abnormal if there is at least one abnormal sample point with respect to the anomaly score of the previous chapter, while the first method examine the distribution of this score. Samples are labeled 0 if they are from the normal regime and 1 otherwise. We test the detection of two different attacks. The first attack is the tempered attack of type i) already used in Chapter 2. The second attack is the attack of type ii), which, we recall, is an attack that concerns only the delay matrices and the subjection vector used for the creation of the time series. More precisely, for the experiment, the delay matrices for which the order of magnitude of the values is 10, are transformed by adding a matrice of values randomly chosen between -1 and 1. Hence, by the way delay matrices affect the cycles ($A$ and $B$ in Definition of Sibriz), differences are very subtle, so this is a very long term attack. This experiment's results are shown in Figure 3.10.

The second experiment consists in veryfying that the monitoring model from the IDS is the rightful autoencoder. The attack consist in replacing the autoencoder that provides reconstruction errors matrices and MSEs, so samples are labeled 0 if they come from the rightful autoencoder and 1 if they come from the wrong autoencoder. ROC curves from WMW on three different variables are plotted, the first variable is the average reconstruction error, the second is the confidence measure of a classical NN (softmax value), and the third is the confidence measure of the multipath NN. The confidence measures follow the predictions from the NNs of the states of the system from the matrices of reconstruction errors and the NNs both achieve almost a perfect test accuracy.

3. The p-value of a test is the probability of observing points at least extreme as the tested sample under the null hypothesis: $\mathbb{P}(|T(x_1, x_2, ..., x_n)| \geq t \mid H_0)$, with $T$ the statistic of the test and $t$ the threshold.

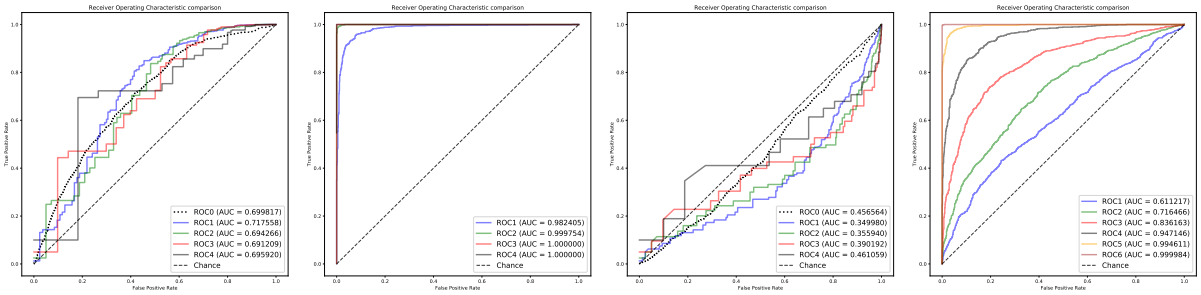



Figure 3.10 – ROC curves from maximum values in samples of MSEs of an autoencoder (the main loss is not necessarily the MSE, in our case it is the Log Loss) (first and third graphs) and from 1 - p-values of the WMW test on the same samples (second and fourth graphs) for samples of size 25 (ROC1), 50 (ROC2), 100 (ROC3), 200 (ROC4), 400 (ROC5) and 800 (ROC6). ROC0 comes directly from the MSEs of the autoencoder used. Each sample is either generated by the normal regime or comes from a time series resulting from an attack. The first two graphs concern tempered attack of type i) explained in chapter 2 and the second two concern a very long term attack of type ii). For the WMW test, samples are tested against a sample of MSEs of size 1000 from the normal regime. The autoencoder used is one of the autoencoders of type ii) trained on time windows of length 15 from states with an average step equal to 10 in the cycles for experiments in chapter 2.

For the WMW tests, samples, of sizes 25 in Figure 3.11 and 200 in Figure 3.12) are tested against a sample of reconstruction errors of size 1000 from the rightful autoencoder. The architectures of the classical NN and of the multipath NN are respectively:

$$
\begin{aligned}
&InputLayer(15,15) \to Conv(15,19)\\
&\to Up(4)(60,19) \to Down(5)(12,19)\\
&\to Conv(12,22) \to Up(4)(48,22)\\
&\to Down(5)(10,2) \to Conv(10,24)\\
&\to Up(4)(40,24) \to Down(5)(8,24)\\
&\to Conv(8,25) \to Flat(200)\\
&\to Dense(5)
\end{aligned}
$$

$$
\begin{aligned}
&InputLayer(15,15) \to Conv(15,19)\\
&\to Up(4)(60,19) \to Down(5)(12,19)\\
&\to Conv(12,22) \to Up(4)(48,22)\\
&\to Down(5)(10,2) \to Conv(10,24)\\
&\to Up(4)(40,24) \to Down(5)(8,24)\\
&\to Conv(8,25) \to Flat(200)\\
&\to Dense(100) \to 5DissLayers(100)\\
&\to Dense(5)
\end{aligned}
$$

with the same notation as in section 2.6.1 for layers $Down$ and $Up$, $Flat$ being a layer without parameters used to transform the shape of the output of the previous layer so that it can be given to a $Dense$ layer which is a traditional FC layer and $5DissLayers$ is a layer composed of 5 Diss-Layers as explained in section 3.2.3 to get a confidence measure.

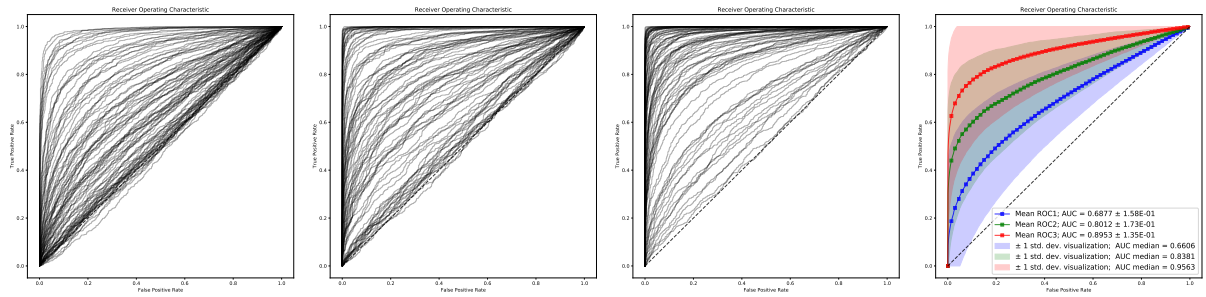



Figure 3.11 – ROC curves from 1 - p-values of the WMW test on an autoencoder MSEs (first graph), from 1 - p-values of the WMW test on the confidence on the state prediction from a traditional NN (second graph) and from a Multipath NN which have been given reconstruction errors matrices (third graph) for samples of size 25. The fourth graph is the comparison between the mean ROC curves of the previous graphs. Each of the first three graphs is composed of 90 ROC curves (10 genuine autoencoders times 9 spoofing autoencoders) detailed in Figures 3.13 and 3.14.

## 3.4 Traceability thanks to the WMW test and the state recovery

### 3.4.1 Is it our Industrial Control System (ICS) ?

As explained at the beginning of this chapter, in order to verify that data describing the physical process of an industrial system is indeed originating from the system at issue, we make use of the WMW test on a summary of these pieces of data. Thus, for the test to be useful, the summary used has to be sensitive to variation of the physical process from its normal regime. That is why a natural summary is the anomaly score presented in the previous chapter. The anomaly score of the previous chapter is the average reconstruction error from an autoencoder of a piece of data, a time window to be more specific. This score is meant to be sensitive to deviation from the normal regime so that an abnormal time window leads to a high score, then a threshold is used to decide whether the time window is normal. While we have seen that this anomaly score can be refined so that long-term attacks can be better detected thanks to a threshold rule, there is a inherent limitation of this technic for detecting very long term attack. Indeed, the evidence of an anomaly can be spread across time windows distant in time, while these time windows alone are not sufficient to claim that an anomaly occured in the time series. Therefore, taking the maximum of the anomaly scores of time windows from a time series to decide whether the time series is normal is not effective for subtle anomaly detection as depicted in Figure

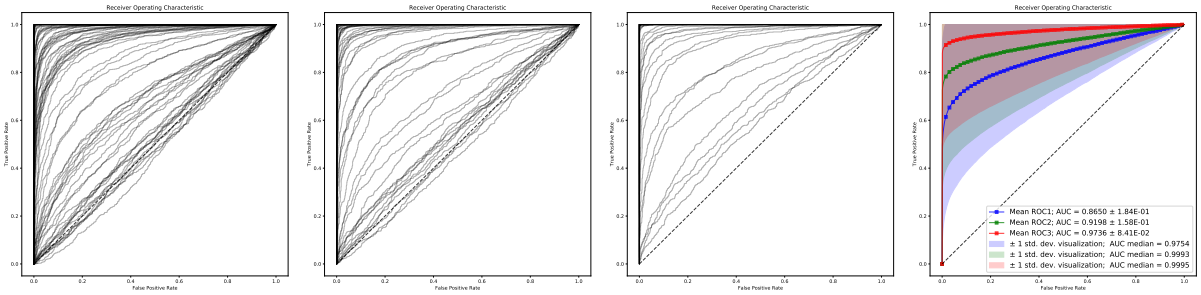



Figure 3.12 – Same as in Figure 3.11 except that the samples size is 200.

3.10. A first solution is to take longer time windows, but there is of course a computational limit to this solution. Hence the need for hypothesis test. In effect, if the evidences of the anomaly are spread across several time windows, the distribution of the anomaly score of the time windows is likely to be changed in comparison with the distribution of the anomaly score for the normal regime and hopefully the same holds for the distributions of the anomaly score under the different regimes. One can appreciate the consistency of the WMW test thanks to Figure 3.10 where the second and fourth graphs show that if the size of the sample to be tested increases the p-value is more relevant. Indeed, even for the second attack, which is much more subtle, having enough sample points allows deciding the normality of the time series.

### 3.4.2 Is it our Intrusion Detection System (IDS) ?

In the previous section, we saw how to authenticate an ICS physical data. The solution relies on an autoencoder from an IDS used for online monitoring by veryfing that its average reconstruction error follows the same distribution than usual. Therefore, one have to assume that the IDS is well protected itself. Yet, the author of [3] reported that security of most of IDSs is low. While this survey dates from 1998, a decade later, in 2009, [44] stated that «*more significant efforts should be done to improve intrusion detection technology in this aspect* [security]». Since then little research focused on this subject. Security issues might be even worse for distributed Intrusion Detection Systems (dIDSs). In our case, the question is then, how to authenticate the autoencoder data? One solution is to save some input-output pairs of the autoencoder so that one can verify that they match those of the supposed authentic autoencoder currently used by the IDS. For digital forensics purpose, one will want to store these input-output pairs for a long

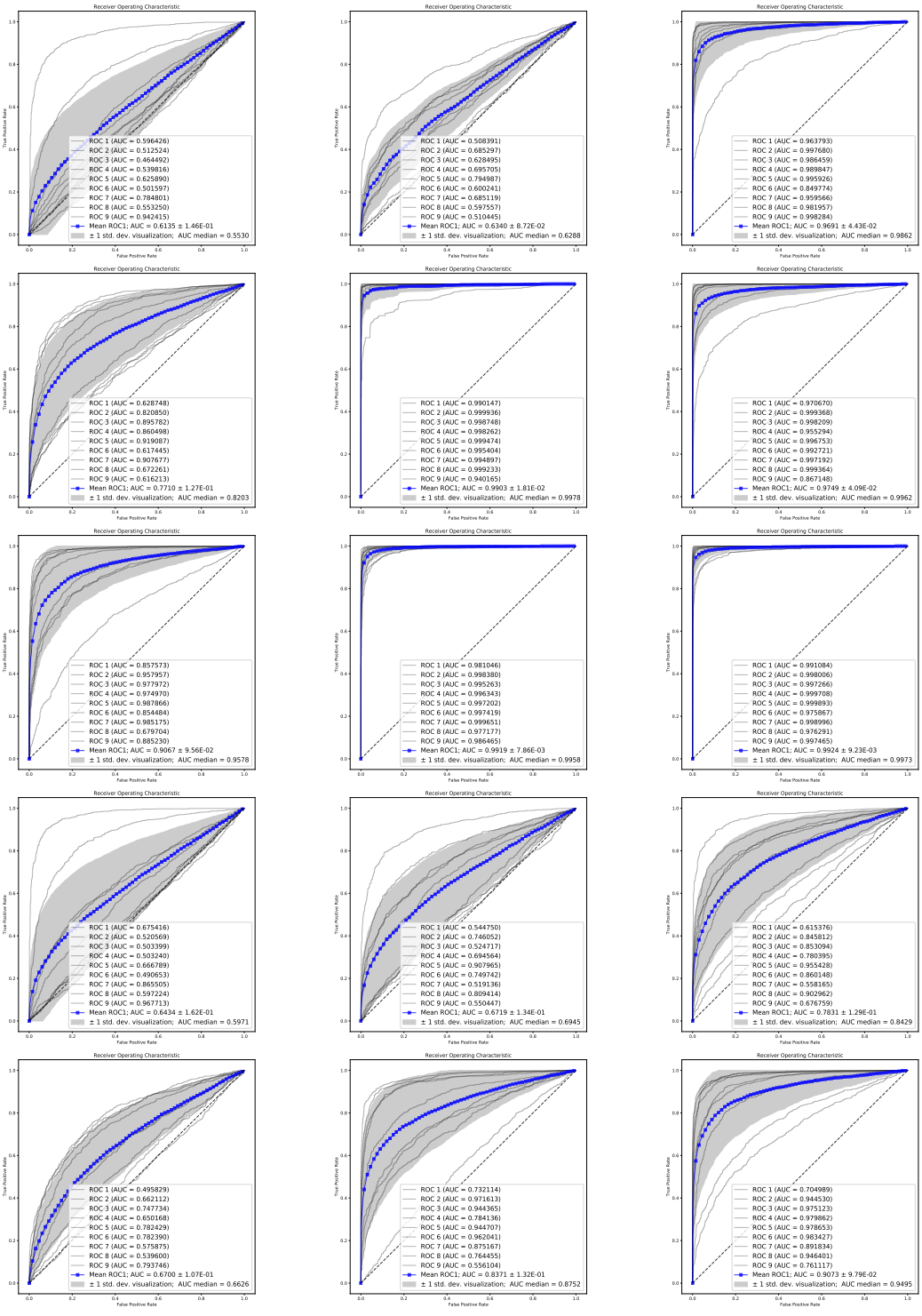



Figure 3.13 – ROC curves from 1 - p-values of the WMW test on an autoencoder MSEs (first column), from 1 - p-values of the WMW test on the confidence (softmax value) on the state prediction from a traditional NN (second column) and from 1 - p-values of the WMW test on the confidence on the state prediction from a Multipath NN which have been given reconstruction errors matrices (third column) for samples of size 25.

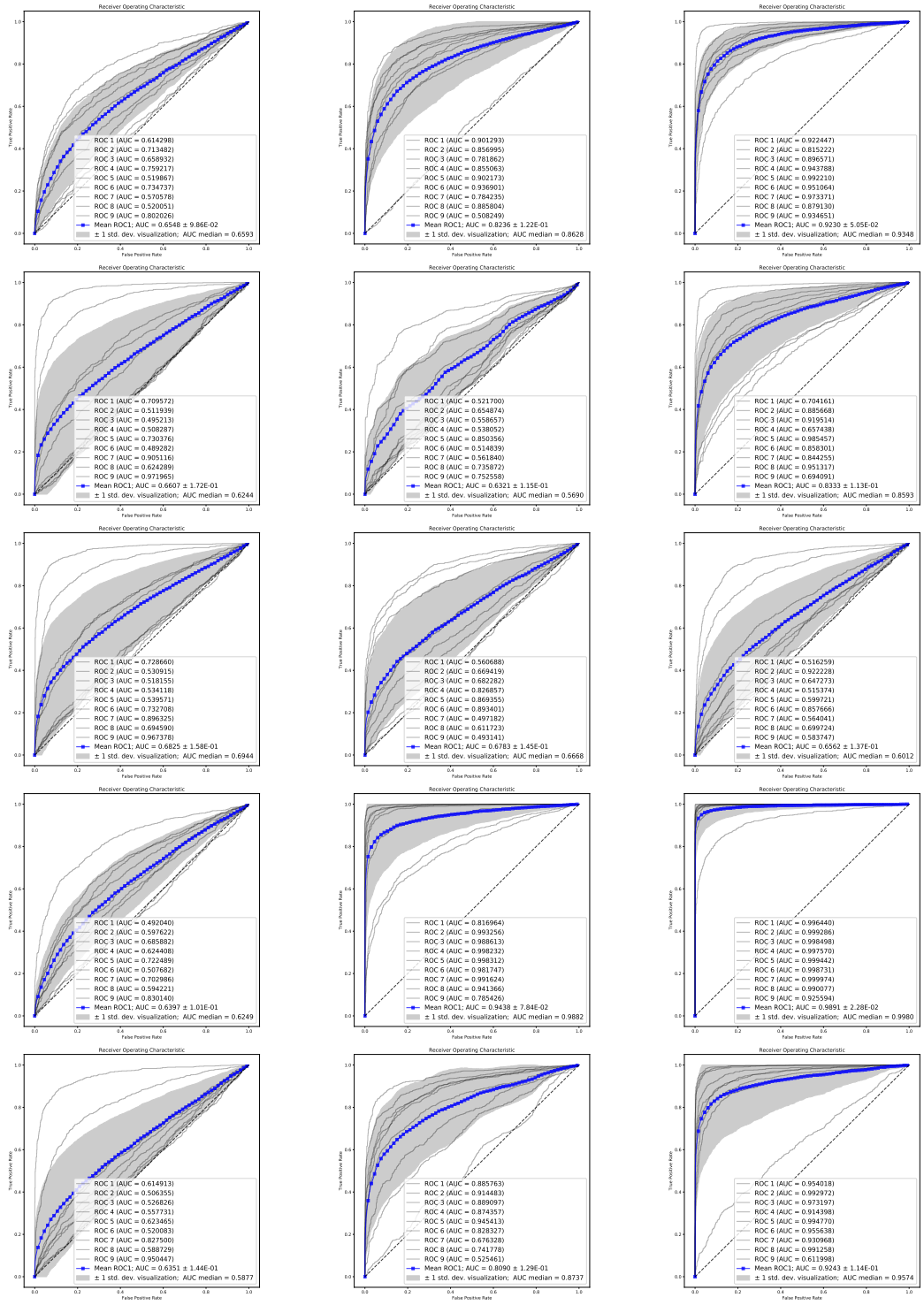



Figure 3.14 – ROC curves from 1 - p-values of the WMW test on an autoencoder MSEs (first column), from 1 - p-values of the WMW test on the confidence (softmax value) on the state prediction from a traditional NN (second column) and from 1 - p-values of the WMW test on the confidence on the state prediction from a Multipath NN which have been given reconstruction errors matrices (third column) for samples of size 25.

history of the ICS working. Therefore, the server storing these input-output pairs have to be itself protected, especially since these input-output pairs provide a lot of information about the autoencoder and the ICS and could thus leak critical information that can be used to bypass the ICS security. Another solution, which, unlike the previous one, cannot weaken the ICS security through information leak, is to not rely on input-output pairs, or equivalently reconstruction errors, from the autoencoder, but on a digest of these data, digest that can serve for a statistical test as in the previous section.

While the WMW test on the autoencoder MSEs can be useful for detecting anomaly in the physical process, it is of course pointless to detect whether the MSEs indeed come from the rightful autoencoder. In effect, an attacker, who tries to spoof the IDS by replacing anomaly scores of it autoencoder by anomaly scores of another autoencoder, will make sure that this second autoencoder produces anomaly scores with the same distribution than the genuine autoencoder. A hypothesis test on another variable is thus necessary in this scenario. Since the anomaly does not come from the physical process but the autoencoder itself, one can try to characterize the autoencoder to produce better digests to perform the test. That is, before proceeding to the recap of the information to carry out the hypothesis test, one can look more carefully at the reconstruction errors. That is where the multipath NN comes into play. We have seen in section 3.2.3 that used as a classifier, this model provides a confidence measure that is robust to changes in the non-discriminative features but sensitive to changes in the discriminative ones. Hence, a multipath NN trained to recover states of a time windows from the reconstruction errors matrices of a 1D-CNN autoencoder will characterize the latter and its confidence measure will be a good digest for the WMW test.

Figures 3.11 and 3.12 show that this digest allows detecting a spoofing attack on the autoencoder with less examples, and thus faster than a WMW test on the MSEs and on the confidence of a traditional classifier NN. It is worth mentioning that if the classical NN is calibrated through temperature scaling the performance decreases: the AUC of mean ROC curves is 0.7481 for samples of size 25 (AUC of median ROC curve: 0.7082) and 0.9086 for samples of size 200 (AUC of median ROC curve: 0.9843). Moreover, when ROC curves are too much close to the first bisector for tests with 25 examples (Figure 3.11), the detection of the corresponding spoofing autoencoder is difficult, if not impossible, with more examples (Figure 3.12). This situation is less likely for the multipath NN than for the traditional classifier NN. Indeed in Figure 3.12, each ROC curve of the third

graph, that is the graph related to the multipath NN confidence measure, is above the first bisector, while some ROC curves are stuck on the first bisector for the two other variables. This means that some spoofing autoencoder will never be detected by the WMW test on these two variables, whereas, since the WMW is consistent under a reasonable assumption already mentioned, the WMW test on our confidence measure will eventually detect every spoofing autoencoder, even if it takes a long time for a few of them. Figures 3.13 and 3.14 show the details of Figure 3.11, that are, for each of the 10 autoencoders, the ROC curves corresponding to the other 9 trying to spoof them. They show that 1) in each case, the framework of the WMW test on the confidence measure of the classifiers is always better than the WMW test on the MSE and 2) that in only two cases of these ten cases, the traditional classifier outperforms our multipath NN. So, not only our model is better on average but it is better more often. Considering the averages of each instance of the models (rows of Figures 3.13 and 3.14) tells us how these instances perform and comparing the medians of these averages tells us which model is generally the best. Our model (median of 0.92) is generally better than the other classifier based model (median of 0.82).

## 3.5 Conclusion

In this chapter, we have presented the WMW test along with a line of research in anomaly detection to make it consistent against malicious actors. Then, after presenting the state of the art of the confidence of a neural network in its own prediction, we introduced a new computational unit that, when well implemented in what we call a multipath NN, can provide a meaningful confidence in the prediction. More specifically, unlike classical NNs, a multipath NN is robust to changes of the non-discriminative features in the input. In the general case, we argue that the consequence of such a property is that, one should be able to derive a model calibrated for any input from a multipath NN along with the proper tools from TDA. In our case, we showed that this property allows to characterize a monitoring model thanks to the WMW test on our measure of confidence in the predictions of the state of model input from its errors. This way, thanks to the WMW test we show to be useful for detecting very long term anomalies, the authentication of each process of transformation of the data, i.e. traceability, can be achieved.

Chapter 4

# Metrics for anomaly scoring functions

We have seen an example of anomaly scoring functions in Chapters 2, i.e. functions that aim to discriminate between normal and abnormal inputs by assigning anomaly scores. We wish to provide metrics for anomaly scoring function. One has to agree on the definition of an anomaly and thus on what we expect from an anomaly scoring function in order to propose such metrics. As propounded by Grubbs [59], abnormal observations can be defined as: «*An outlying observation, or "outlier", is one that appears to deviate markedly from other members of the sample in which it occurs.*». Based on such a definition, metrics on anomaly scoring functions should naturally favor scoring functions that give higher scores to rare observations than scores for frequent observations, or the converse depending on the convention. The criterion to evaluate is thus whether the preoder induced by the anomaly scoring function is similar to the one induced by the opposite of the density underlying the normal class [1]. This criterion is the one of the Mass-Volume curve (MV) [22] and Excess-Mass curve (EM) [52] curve metrics, for example. However, as explained in Chapter 2, one may want to prioritize the detection of some type of anomaly over another type. In doing so, the definition from Grubbs is tempered by other aspects of concern for experts. For example, security experts can take into account the financial cost of some type of attack and the life-threatening risk of another. It can be done in different ways, either by using a method to classify the anomaly types in order to nuance the anomaly score or as part of the scoring funtion as done in Chapter 2. In the latter case, the similarity between the preorders of the scoring function and the density underlying the normal class is not necessarily the adequate criterion to consider. Indeed, one may dread some type of anomaly more than another while the first is more likely than the second. One way to compare anomaly scoring functions is to use statistical metrics such as accuracy, specificity, sensitivity... Statistical metrics concern the statistical performance

1. ... because of the convention *high scores for abnormal points*, but if the convention is *high scores for normal points*, the preorders induced by the anomaly scoring function and the density should be considered.

of a binary classification test. They therefore fit in the context of anomaly detection. However, the metrics cited above tell little about the scoring function itself, just its end result. Thus, we seek for metrics that enlighten us about the relevance of the scoring function. We introduce two metrics in the two subsequent sections.

## 4.1 The disparity

Besides the MV [22] and EM [52] metrics, there are no metrics on anomaly scoring functions themselves. While these two metrics rely on a well-grounded theory, namely multivariate quantiles [34], as explained above they are not suited to every outlier detection problem. That is why we seek for other metrics that, like MV and EM, adopt a topological point of view instead of a strictly statistical point of view, but with another criterion.

Let us be clear about what we mean by topological and statistical metrics. A statistical metric on anomaly scoring functions is a performance measure of the end result of the anomaly scoring function it evaluates, i.e. it tells us whether the function provide good insight about the «normality» of a point. A topological metric is a measure with topological properties. For example, MV and EM curves are invariant by increasing transformation of the scoring function. Moreover, MV and EM curves are also statistical metric since they can substitute the ROC curve. As observed earlier, the topological property of the MV and EM curves is inadequate when one wants to compare two anomaly scores in accordance with the pondering of an expert about such or such type of anomaly as it is the case for our integrity measure in Chapter 2. Here, we suppose that the preference of the expert is totally expressed in the construction procedure of the scoring function and that the metric does not assist this construction. In constrast, MV and EM were originally used to build scoring functions [51]. Our purpose is to define a metric that indicates how much the procedure is suited to a problem given that this procedure is consistent, that is given that the scoring function is improved with respect to some statistical metrics during the construction procedure. Hence, we opt for considering real-valued scores by their alikeness to binary scores, viewed as ideal, under their anomaly threshold. The decision boundary is said to be all the more certain that it fits the corresponding binary score jump discontinuity [2]. That is why we assume a weaker topological property for scoring

2. We assume that the binary score the scoring function tries to approximate is continuous almost

functions which is the invariance by increasing affine transformation. In other words, we want to define a metric that underlines the «self-confidence» of an anomaly scoring function, while statistical metrics reflect its «achievement». Together, the «self-confidence» and the «achievement» shall indicate to which extent a scoring function is suited to a given outlier detection problem. The adjustment of the definition from Grubbs will be indirectly imparted in the assessment of the «self-confidence» by the desired topological metric on a scoring function that is supposed to be constructed by an algorithm that includes the pondering of the expert.

In this chapter, score anomaly functions are assumed to be $n$-dimensional functions that assign high scores to abnormal observations and whose image is bounded [3]: $F : E \to \mathbb{R}$, with $E \subset \mathbb{R}^n$. Furthermore, from now on, the underlying density on which $\mathbb{E}_X$ [4] are relying is supposed to describe the normal set-up leading to the observations of $X$. $\mathbb{E}(. \mid X \in B)$ will be written $\mathbb{E}_{X \in B}(.)$. The metrics defined in this chapter are meant to assess the behavior of the scoring function on data that are assumed normal.

**Definition 4.1.** *The* **disparity** *of an anomaly scoring function $F$ is the Gini coefficient of the Euclidean norm of the normal data images under $F$ partial derivatives:*

$$\begin{aligned}\mathscr{D}(F) :&= \frac{\mathbb{E}_X(\mathbb{E}_Y(|\|\nabla F(X)\|_2 - \|\nabla F(Y)\|_2|)}{2.\mathbb{E}(\|\nabla F(X)\|_2)} \\ &= \frac{\mathbb{E}_X(\mathbb{E}_Y(|\|(\nabla_{v^i} F(X))_{1<i\leq n}\|_2 - \|(\nabla_{v^i} F(Y))_{1<i\leq n}\|_2|))}{2.\mathbb{E}(\|(\nabla_{v^i} F(X))_{1<i\leq n}\|_2)}\end{aligned}$$

*with $((v^i_j)_{1<j\leq n})_{1<i\leq n} = (v_j)^\top_{1<j\leq n} = (v^i)_{1<i\leq n}$ any orthogonal basis of the domain of $F$ such that $\exists c / \forall j, \|v_j\|_2 = c$.*

The disparity does not depend on the direction of the partial derivatives as proved by (4.1), as long as the basis is a transformation of the canonical basis by a rotation and a homothety both centered in the origin, i.e. each direction is given the same importance. The Gini coefficient is a measure of statistical dispersion originally introduced to represent economic variables, like incomes of inhabitants of a country, in order to evaluate inequalities within a population. It ranges from 0 to 1, 0 representing perfect equality, for

everywhere.

3. This is necessary to be able to compare two scoring functions with the metrics defined thereafter, as any bounded scoring function $G$ can be transformed into a function $F$ between 0 and 1 without changing their level sets: $F : x \to (G(x) - \inf_{t\in E}(G(t)))/(\sup_{u\in E}(G(u)) - \inf_{v\in E}(G(v)))$.

4. Abbreviated $\mathbb{E}$ when there is no ambiguity.

example everyone has the same income, and 1 representing perfect inequality, that is the limit of the Gini coefficient when one person has all the income in a growing population or when one person has a growing income when others incomes are constant, for instance. It is useful for our problem to quantify how much a scoring function is similar to a binary score up to an increasing affine transformation. Indeed, when the Gini coefficient of the Euclidean norm of the score function partial derivatives is 0, it means that the scoring function is an affine function and thus presents no step alike variation at all. In contrast, the more this Gini coefficient gets close to 1, fewer points are present in the part of the graph with a marked slope. However, the disparity is not enough to compare every anomaly scoring functions. Indeed, it says if there is some step alike variations but it does not say at which level these variations occur in the graph. Yet a function similar to a binary score should have these levels low enough relatively to the function maximum and minimum values, in particular they should be near 0 if the function ranges from 0 to 1. Let us note that the disparity is trivially invariant by affine transformations which is less restrictive that invariant by increasing affine transformations. A higher order of derivation is needed to assess the binary score likeness of a function.

$$\begin{aligned}
\|(\nabla_{v^i} F(X))_{1<i\leq n}\|_2^2 &= \sum_{i=1}^{n}(\sum_{j=1}^{n} v_j^i \nabla_j F(X))^2 = \sum_{k=1}^{n}\sum_{j=1}^{n} \nabla_j F(X)\nabla_k F(X) \sum_{i=1}^{n} v_j^i v_k^i \\
&= \sum_{k=1}^{n}(\nabla_k F(X))^2 \|v_k\|_2^2 + \sum_{j,k,j\neq k}^{n} \nabla_j F(X)\nabla_k F(X)\langle v_j \mid v_k\rangle \\
&= c^2 \sum_{k=1}^{n}(\nabla_k F(X))^2 + 0 \\
&= c^2 \|\nabla F(X)\|_2^2
\end{aligned} \tag{4.1}$$

## 4.2 The susceptibility

Since the disparity misses the levels where thresholds between normal and abnormal inputs can be fixed, it is natural to consider as potential decision boundaries the inter-levelsets $\mathcal{L}_{a,b}(F) := \{x \in E \mid a < F(x) < b\}$ in the definition of an additional metric. A scoring that approximates a binary function and that possesses some regalurity, such as Lipschitz continuity, assuring that small variations in the domain result in small variations in the codomain, should have its image of the decision boundary «more or less convex» under the threshold. Of course, a scoring function is not necessary Lipschitz continuous so

our desire is to define a convexity «trend» measure of the anomaly score under the threshold in the decision boundary. The term «trend» is important because we want to attribute to a non-convex curve a score that is all the greater as it can be approximated by convex curves that admit large parameters of strong convexity. In practice, if $F$ is supposed continuous, it can be estimated by the transformation by the Multivariate Savitzky-Golay filter (M-SGf) [121, 139] of the linear interpolation on the dataset on which the scoring function is applied.

**Definition 4.2.** *Given a decision boundary $B$, that is a soft boundary meant to separate normal data $X$ from anomalies, the* **susceptibility** *of an anomaly scoring function $F$ on $B$ is the conditional expectation given $X \in B$ of the mean of eigenvalues $\lambda_i(.)$ of the Hessian*[5] *$\nabla^2 F(X)$: $S(F,B) := \mathbb{E}_{X\in B}(\sum_{i=1}^{n}\frac{1}{n}\lambda_i(\nabla^2 F(X))) = \sum_{i=1}^{n}\frac{1}{n}\lambda_i(\mathbb{E}_{X\in B}(\nabla^2 F(X)))$. The* **global susceptibility** *is then defined as follows:*

$$\begin{aligned}\mathscr{S}(F) :&= \frac{1}{M-m} \times \sup_{m<t<M}(S(F, \mathcal{L}_{m+(t-m)\times\mathscr{D}(F),t}(F))) \\ &= \sup_{0<t<1}(S(\tilde{F}, \mathcal{L}_{t\times\mathscr{D}(F),t}(\tilde{F})))\end{aligned}$$

*with $\tilde{F} : x \to (F(x)-m)/(M-m)$, $M = \sup_{v\in E}(F(v))$ and $m = \inf_{v\in E}(F(v))$.*

The equality in the definition of the susceptibility is straightforward remembering that the sum of a symmetric matrix eigenvalues equals its trace. Hence, as for the disparity, the susceptibility does not depend on the partial derivatives. The global susceptibility is trivially invariant by increasing affine transformation. We prefer the term susceptibility to sensitivity which, in anomaly detection, denotes the true positive rate. Though, we will use the term sensitive as the adjective to refer to the degree of susceptibility of an anomaly scoring function. In the same vein as similarity to binary scores, the will to compute the susceptibility hinges on the point that an ideal anomaly score is binary. Hence, $S(F,B)$ assesses the alikeness of $F$ to binary scores in the decision boundary $B$. The function $F$ is then considered all the more «self-confident» that it fits a binary score jump discontinuity, i.e. he has high global susceptibility. We will say that the greater is $S(F,B)$, the more sensitive is the function $F$. We use the supremum because the decision boundary of an anomaly scoring function similar to a binary score is situated where the graph has the

5. If $F$ is a variable discrete function, we can consider the discrete Hessian composed with the second difference operators and the discrete gradient composed of the first difference operator [146], otherwise we assume that $F$ is $C^2$.

most convexity trends. The motivation behind the utilization of $t \times \mathscr{D}(F)$ as the lower level in the interlevel set is to automatically find a proper potential decision boundary. Indeed, let us reason with a function $F$ of class $C^2$ that ranges between 0 and 1. If $F$ is an anomaly scoring function that is similar to a binary function with image $\{0, 1\}$, then the set $C = \{x \in E \mid \forall y, tr(\nabla^2(F)(x)) \geq tr(\nabla^2(F)(y)) - r\}$ with $tr(.)$ the trace and $r$ small enough, is a subset of an interlevel set $\mathcal{L}_{a,b}$, with $b-a$ small. Yet as explained earlier, since $F$ is a good approximation of a binary score, $\mathscr{D}(F)$ is high, so $t - t \times \mathscr{D}(F)$ is low and $\mathcal{L}_{t\times\mathscr{D}(F),t}$ is a good candidate as a decision boundary for some $t$. Conversely, if $F$ is not similar to a binary score, then the smallest difference $b - a$ from a $\mathcal{L}_{a,b} \supset C$ is relatively high, that is the potential decision boundaries have modest convexity trends. Yet, in this case, $t - t \times \mathscr{D}(F)$ is high.

The more sensitive is a scoring function, the more pronounced is its separation, so the less the statistical performances should vary from a dataset to another. So the global susceptibility can be viewed as a measure of the trust one can have in the statistical performance on a dataset which might be too small to take the result for granted. Hence, the global susceptibility should help to decide between two scoring functions that has equivalent statistical performances. For now this metric has not been tested empirically but if it works, it will have another advantages in addition to those mentionned above. Thanks to gradient and hessian generalization to discrete values proposed in [146], it will not require the data to be continuous, unlike MV and EM.

## 4.3 Conclusion and perspective

In this chapter, we proposed two metrics on anomaly scoring functions based on a different criterion than the one of the only two existing metrics: the MV curve and EM curve metrics. More precisely, our metrics allow to compare anomaly scoring functions that have been defined so as to prioritize the detection of some types of anomalies against other types assuming that the the construction procedures of the scoring functions are consistent, i.e. that the scoring functions are improved with respect to some statistical metrics during the procedure. Further research needs to be done to develop efficient ways to compute these metrics and also to test them experimentally.

# Conclusion

In this dissertation, we define integrity and traceability of physical processes. We tackle the problem of passively assessing data integrity (quantification of alteration of data linked to specific actions) for ICSs thanks to the broadly used autoencoders' reconstruction errors framework of anomaly detection. Thanks to such NNs and a novel notion of state in CPSs, we were able to push the state of the art in anomaly detection in physical data from the testbed SWaT and to specify the integrity score so that the detection of some type of anomalies is given more emphasis. This integrity score can then be used for outlier detection thanks to the GEV distribution from the well-grounded theory EVT. We introduced Fast Maxima Sampling useful for the estimation of the GEV distribution and proved its efficiency and coherence. As for the traceability of physical processes from ICSs – which refers to verifying the authenticity of each processus of transformation of the data – we formulate the problem as a hypothesis testing and provide a novel DL model, called the multipath NN, that can be used during online monitoring or as a forensic analysis tool to check that the IDS is not compromised. Finally, we propose new metrics to assess the relevance of anomaly scoring functions. Back to our classification of physical characterization methods in Figure 3, our work, both on integrity and traceability, adds a new branch which is the characterization of the near-deterministic working of a system.

Let us now detail the limits of our study.

Our solution to data integrity control allows taking into account the expert knowledge to a certain extent. More precisely, the expert can balance the detection of replay attacks versus other attacks and long-term attacks versus punctual attacks, or vice versa. While this is a first step towards more flexibility in the definition of normality the expert can decide, there are certainly many other attack types that could be interesting to focus on.

Experiments on different ways of choosing states of interest could be done to find out how sensitive our method is to the choice of states.

Our work does not tackle the issue of the security of the monitoring system itself using methods other than adding another layer of security as it is the case for the mon-

itoring autoencoder which is protected by the multipath NN. Thus, one could say that we have just shifted the problem, even if this can be an advantage because some security mechanisms might be implemented within this second layer that could not have been implemented within the first since IDSs security is known to be challenging [44]. Moreover, this layer of protection may be itself incomplete: surprisingly enough, it is known that adversarial examples are transferable from a classifier NN to other classifier NNs of the same type, a different transferability, on a whole sample, could make this second layer of security useless in front of a whole class of attack. In our framework, transferability would not concern the output of the classifier NN but the absence of shift of distribution of its confidence measure when the classifier input undergo a shift of distribution. So, if this kind of transferability holds, an attacker would «just» have to build a multipath NN and find a set of inputs for this NN – which are, in name only, the error matrices of the monitoring autoencoder – such that its confidence measure distribution do not change, in order to make the WMW test on the confidence measure values of the genuine multipath NN useless.

Another limitation of our work is that we did not test our solution for traceability on real data. Indeed, testbeds with physical data do not provide data in a range longer than days, while the lack of traceability may only be detectable after a few weeks, if not months. Finally, as mentionned Chapter 3, there is a condition for the WMW test to be consistent and the attacker could try to bypass this condition. So, we exposed a line of research for making the WMW test consistent against malicious actors supported by some properties of probability density functions, but this line of research, just as our new metrics for anomaly scoring functions, yet needs stronger theoretical and empirical results before being adopted as a potential security solution.

Finally, we propose in Section E, a line of research to make the confidence measure of a multipath NN that is too technical to be presented herein but that we summarize in this last paragraph. This line of research is based on the fact that the confidence in the prediction of NNs is highly dependent on the location of the input in its ambient space (cf. Section E). Hence, topological considerations are necessary to enable a general solution to the issue of confidence in the prediction for an input without hypotheses on its underlying distribution. We think that the limitations of our method can be overcome thanks to the work presented in [12] where a tool from TDA, the persistent diagram, is adapted to incorporate some topological properties in the data representation of a

model. Indeed, they proposed a loss function of a persistent diagram. There is nothing preventing us from using this tool to assess the topological properties of the hidden layer space of a NN. The idea is to transform this new input thanks an optimization problem on the persistence diagrams of these filtrations so that the transformed input matches the topological properties of the training dataset. Indeed, persistence diagrams encode k-dimensional holes (0-d holes: gap between two components, 1-d holes: holes, 2-holes: cavities, etc..) of a dataset and they are stable with respect to pertubations of the data but not stable with respect to outliers. Hence, the idea would be to regularize the input given to the multipath NN so it is more similar to the training set in terms of topological properties rather than just distances. So, our expectation is that the input transformed by the regularization with some cost function, for instance the Wasserstein distance between two diagrams in [12] from the training set and from the one with a new input, will produce a calibrated confidence in the prediction of the multipath NN. Indeed, let us recall that the confidence measure of the multipath NN is robust to some extent to changes in the non-discriminative features. Yet, these changes are the less disruptive for the persitent diagrams than changes in discriminative features (cf. Section E). So if the defense mechanism is to minimize the difference between the two aforesaid persistence diagrams by transforming the test sample point, it will either be closer to one of the two nearby classes or moved in a non-discriminative direction. Since our confidence is robust against this latter kind of change, both options will tend to calibrate it. Therefore, one can expect that the prediction will change or the confidence will decrease in the face of outliers. In that respect, the transformation of the input to be tested thanks to optimization problem on persistence diagrams with tools of which [12] have proven the usefulness in the context of DL, jointly with our multipath NN, seems to us to be a good way to detect adversarial examples, launching a promising research direction in DL. However, one can legitimately object that this line of research will be confronted with the difficult convergence of the NNs using the Wasserstein distance.

# Index of ideas

The coffee machine joke that says that nobody read theses anymore is pretty frustrating. To reduce my frustration, here is the list of my main ideas, some more accomplished than others, with hyperlinks so that they are quickly accessible:


— Page 22: a classification of passive physical characterization methods.

— Page 29: Fast Maxima Sampling, an efficient algorithm proved to be useful for GEV distribution estimators.

— Page 30 (footnote): an independence test for ordinal 1rst-order stationary processes.

— Page 38: an approximation of $\Theta(n)$, i.e., a Poisson binomial distribution of parameters $(\frac{1}{k})_{1\leq k\leq n}$, by a Poisson distribution of parameter $\sum_{k=1}^{n}\frac{1}{k}(1-\frac{1}{k})$ shifted by $\frac{\pi^2}{6}$ to the right.

— Page 43: Sibriz, depicted in Figure 2.2, a simulator that generates probabilistically realistic physical data a binary register ICS with logical and physical parameters easy to change to simulate attacks or other anomalies.

— Page 53: a novel concept of state which allows ML models to benefit from the near-deterministic working of ICSs.

— Page 70: the Rational Sampling, a new pooling layer useful to smoothly change spatial dimensions of a CNN's layers.

— Page 72: the Tempered Center Loss, a loss to be applied on a FC layer useful to make a trade-off between long-term and punctual anomaly detections in order to specify ICS data integrity, also a generalization of the Center Loss with inter-class variation awareness.

— Page 80: the Similarity Transfer Loss, a loss to be applied on a convolutional layer of a CNN with a swelling topology useful to make a trade-off between punctual and long-term anomaly detections in order to specify ICS data integrity.

— Page 89: a line of research based on a secret transformation of the data and a WMW test on a summary of these transformations for making the WMW test consistent against malicious actors.

— Page 89: a line of research based on properties of a density function $f$ applied on its own variable $X$ for defining a consistent test against malicious actors by performing the WMW test on each of $(f^k(X))_{0\leq k\leq n}$.

— Page 92: a new strategy Confidence Through Path Validation (CTPV), that is checking at the path of the information within a NN for estimating a measure of confidence in the trustworthiness of the input data.

— Page 95: the Dissector Layer (Diss-Layer), a new computational unit for DL models, defined to unsupervisedly separate two linearly separable clusters when used in what we call a Hyper-Neuron autoencoder constituted, all in all, of two Dissector Layers.

— Page 100: the multipath NN, a DL classifier, following CTPV, with at least one of its layers composed of Diss-Layers that receive the same input so that one can compute a confidence measure that is robust to changes in the non-discriminative features of the input when proper objectives have been applied during training.

— Page 111: the WMW test on an autoencoder MSE to authenticate physical data.

— Page 112: the WMW test on a confidence measure of multipath NN for its prediction of the state of the reconstructed time window from the recontruction errors of the autoencoder to characterize the latter.

— Page 117: a definition of two metrics that allow comparing anomaly scoring functions that have been defined so as to prioritize the detection of some types of anomalies against other types assuming that the construction procedures of the scoring functions are consistent, i.e., that the scoring functions are improved with respect to some statistical metrics during the procedure. Purely theoretical for now.

— Page 150: a simple method to approximate the anti-function $f^{-1}$ of a bijection $f$ based on a contraction mapping leading to a series converging towards $f^{-1}$.

— Page 154: a definition of a measure of the level of linear separability.

— Page 164: a line of research for a method based on the multipath NN that provides a confidence measure calibrated on any input. The idea is to take advantage of the robustness of the multipath NN's confidence measure facing with changes of non-discriminative features by using recent tools of TDA to transform the input until the two persistence diagrams regarding the training set with and without the point to transform are similar. My expectation is that this transformation will converge when the transformed point is not ambiguous anymore, then one could compute a reliable confidence score.

# Bibliography


[1] Guillaume Alain and Yoshua Bengio, « Understanding intermediate layers using linear classifier probes », *in*: *ICLR*, 2017, URL: https://openreview.net/forum?id=HJ4-rAVtl (cit. on pp. 93, 96, 97).

[2] Karpathy Andrej, *Convolutional Neural Networks (CNNs / ConvNets)*, 2016, URL: http://cs231n.github.io/convolutional-networks/ (visited on 05/29/2019) (cit. on p. 60).

[3] Stefan Axelsson, *Research in intrusion-detection systems: A survey*, tech. rep., Citeseer, 1998 (cit. on p. 112).

[4] Ralph W Bailey, « Polar generation of random variates with the t-distribution », *in*: *Mathematics of Computation* 62.*206* (1994), pp. 779–781 (cit. on p. 30).

[5] Laura Beggel, Michael Pfeiffer, and Bernd Bischl, « Robust Anomaly Detection in Images using Adversarial Autoencoders », *in*: *arXiv preprint arXiv:1901.06355* (2019) (cit. on p. 18).

[6] Jan Beirlant et al., *Statistics of extremes: theory and applications*, John Wiley & Sons, 2006 (cit. on pp. 28, 29).

[7] Rainer Böhme et al., « Multimedia forensics is not computer forensics », *in*: *International Workshop on Computational Forensics*, Springer, 2009, pp. 90–103 (cit. on p. 17).

[8] Boris Danev, David Zanetti, Srdjan Capkun, « On Physical-Layer Identification of Wireless Devices », *in*: (2012) (cit. on p. 19).

[9] Boris Danev, Thomas S. Heydt-Benjamin, Srdjan Capkun, « Physical-layer Identification of RFID Devices », *in*: (2009) (cit. on p. 22).

[10] Vladimir Brik et al., « Wireless device identification with radiometric signatures », *in*: *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ACM, 2008, pp. 116–127 (cit. on p. 22).

[11] Jane Bromley et al., « Signature verification using a "siamese" time delay neural network », *in*: *Advances in neural information processing systems*, 1994, pp. 737–744 (cit. on p. 83).

[12] Rickard Brüel-Gabrielsson et al., « A Topology Layer for Machine Learning », *in*: *arXiv preprint arXiv:1905.12200* (2019) (cit. on pp. 124, 125, 164–166).

[13] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil, « Model compression », *in*: *SIGKDD*, 2006, 535::7 (cit. on p. 94).

[14] Massimo Buscema, Stefano Terzi, and Tattile, « Semeion Handwritten Digit Data Set », *in*: *Center for Machine Learning and Intelligent Systems* (1994) (cit. on p. 159).

[15] Yue Cao et al., « Deep Quantization Network for Efficient Image Retrieval. », *in*: *AAAI*, 2016, pp. 3457–3463 (cit. on p. 83).

[16] Varun Chandola, Arindam Banerjee, and Vipin Kumar, « Anomaly detection: A survey », *in*: *ACM computing surveys (CSUR)* 41.*3* (2009), p. 15 (cit. on p. 13).

[17] Frédéric Chazal and Bertrand Michel, « An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists », *in*: *arXiv preprint arXiv:1710.04019* (2017) (cit. on p. 165).

[18] Wai-Ki Ching, Michael K Ng, and Eric S Fung, « Higher-order multivariate Markov chains and their applications », *in*: *Linear Algebra and its Applications* 428.*2-3* (2008), pp. 492–507 (cit. on p. 44).

[19] François Chollet et al., *Keras*, 2015 (cit. on pp. 71, 159).

[20] Christian Wachsmann, Ahmad-Reza Sadeghi, *Physicallly Unclonable Functions (PUFs) Applications, Models, and Future Directions*, 2014 (cit. on p. 21).

[21] Wenqing Chu and Deng Cai, « Stacked similarity-aware autoencoders », *in*: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, AAAI Press, 2017, pp. 1561–1567 (cit. on pp. 63, 70, 72, 75, 76).

[22] Stéphan Clémençon and Jérémie Jakubowicz, « Scoring anomalies: a M-estimation formulation », *in*: *Artificial Intelligence and Statistics*, 2013, pp. 659–667 (cit. on pp. 117, 118).

[23] Valentina Conotter, James F O'Brien, and Hany Farid, « Exposing digital forgeries in ballistic motion », *in*: *IEEE Transactions on Information Forensics and Security* 7.*1* (2011), pp. 283–296 (cit. on pp. 20, 22).

[24] Gregory W Corder and Dale I Foreman, *Nonparametric statistics for non-statisticians*, 2011 (cit. on p. 88).

[25] Thomas M Cover, « Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition », *in*: *IEEE transactions on electronic computers 3* (1965), pp. 326–334 (cit. on p. 81).

[26] Thomas Daniels, Mani Mina, and Steve F Russell, « Short paper: a signal fingerprinting paradigm for general physical layer and sensor network security and assurance », *in*: *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, IEEE, 2005, pp. 219–221 (cit. on p. 20).

[27] John N Darroch et al., « On the distribution of the number of successes in independent trials », *in*: *The Annals of Mathematical Statistics* 35.*3* (1964), pp. 1317–1321 (cit. on p. 38).

[28] Jesse Davis and Mark Goadrich, « The relationship between Precision-Recall and ROC curves », *in*: *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 233–240 (cit. on pp. 58, 59).

[29] Dorothy E Denning, « Stuxnet: What has changed? », *in*: *Future Internet* 4.*3* (2012), pp. 672–687 (cit. on p. 42).

[30] Terrance DeVries and Graham W Taylor, « Learning confidence for out-of-distribution detection in neural networks », *in*: *arXiv preprint arXiv:1802.04865* (2018) (cit. on p. 95).

[31] DIAMONDS Consortium, « Development and Industrial Application of Multi-Domain Security Testing Technologies », *in*: (2010) (cit. on p. 17).

[32] Scott W Doebling, Kenneth F Alvin, and Lee D Peterson, « Limitations of state-space system identification algorithms for structures with high modal density », *in*: *Proceedings-spie the international society for optical engineering*, Citeseer, 1994, pp. 633–633 (cit. on p. 18).

[33] Bradley Efron, « Bootstrap methods: another look at the jackknife », *in*: *Breakthroughs in statistics*, Springer, 1992, pp. 569–593 (cit. on p. 29).

[34] John HJ Einmahl and David M Mason, « Generalized quantile processes », *in*: *The Annals of Statistics* (1992), pp. 1062–1078 (cit. on p. 118).

[35] Jeffrey L. Elman and David Zipser, « Learning the Hidden Structure of Speech », *in*: *The Journal of the Acoustical Society of America* 83.*4* (1988), pp. 1615–1626 (cit. on pp. 60, 61).

[36] Daniel B Faria and David R Cheriton, « Detecting identity-based attacks in wireless networks using signalprints », *in*: *Proceedings of the 5th ACM workshop on Wireless security*, ACM, 2006, pp. 43–52 (cit. on p. 22).

[37] Ronald Aylmer Fisher and Leonard Henry Caleb Tippett, « Limiting forms of the frequency distribution of the largest or smallest member of a sample », *in*: *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, 2, Cambridge University Press, 1928, pp. 180–190 (cit. on p. 24).

[38] David Formby et al., « Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems », *in*: *NDSS*, 2016 (cit. on pp. 19, 22).

[39] Roy Fox et al., « Multi-Task Hierarchical Imitation Learning for Home Automation », *in*: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2019, pp. 1–8 (cit. on p. 41).

[40] Maurice Fréchet, « Sur la loi de probabilité de l'écart maximum », *in*: *Annales de la societe Polonaise de Mathematique*, 1927 (cit. on p. 24).

[41] Jun-ichiro Fukuchi, « Bootstrapping extremes of random variables », *in*: (1994) (cit. on p. 29).

[42] Kunihiko Fukushima and Sei Miyake, « Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position », *in*: *Pattern recognition* 15.*6* (1982), pp. 455–469 (cit. on p. 15).

[43] Yarin Gal and Zoubin Ghahramani, « Dropout as a bayesian approximation: Representing model uncertainty in deep learning », *in*: *ICML*, 2016, 1050::10 (cit. on p. 95).

[44] Pedro Garcia-Teodoro et al., « Anomaly-based network intrusion detection: Techniques, systems and challenges », *in*: *computers & security* 28.*1-2* (2009), pp. 18–28 (cit. on pp. 112, 124).

[45] Laurent Gardes, *Théorie des valeurs extrêmes*, `http://irma.math.unistra.fr/~gardes/Poly_extreme.pdf`, 2017 (cit. on pp. 24–26).

[46] Hamid Reza Ghaeini et al., « State-aware anomaly detection for industrial control systems », *in*: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, ACM, 2018, pp. 1620–1628 (cit. on p. 63).

[47] Mohamed E Ghitany, Barbra Atieh, and Saralees Nadarajah, « Lindley distribution and its application », *in*: *Mathematics and computers in simulation* 78.*4* (2008), pp. 493–506 (cit. on p. 30).

[48] Xavier Glorot and Yoshua Bengio, « Understanding the difficulty of training deep feedforward neural networks », *in*: *AISTATS*, 2010, 249::8 (cit. on p. 156).

[49] Boris Gnedenko, « Sur la distribution limite du terme maximum d'une serie aleatoire », *in*: *Annals of mathematics* (1943), pp. 423–453 (cit. on p. 24).

[50] Jonathan Goh et al., « A dataset to support research in the design of secure water treatment systems », *in*: *International Conference on Critical Information Infrastructures Security*, Springer, 2016, pp. 88–99 (cit. on pp. 67, 146).

[51] Nicolas Goix, « How to Evaluate the Quality of Unsupervised Anomaly Detection Algorithms? », *in*: *arXiv preprint arXiv:1607.01152* (2016) (cit. on p. 118).

[52] Nicolas Goix, Anne Sabourin, and Stéphan Clémençon, « On anomaly ranking and excess-mass curves », *in*: *Artificial Intelligence and Statistics*, 2015, pp. 287–295 (cit. on pp. 117, 118).

[53] Markus Goldstein and Andreas Dengel, « Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm », *in*: *KI-2012: Poster and Demo Track* (2012), pp. 59–63 (cit. on p. 23).

[54] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, `http://www.deeplearningbook.org`, MIT Press, 2016 (cit. on pp. 15, 59, 70, 72, 75).

[55] Ian J. Goodfellow et al., « Generative adversarial nets », *in*: *NIPS*, 2014, 2672::9 (cit. on p. 95).

[56] Benjamin Graham, « Fractional max-pooling », *in*: *arXiv preprint arXiv:1412.6071* (2014) (cit. on p. 71).

[57] J Arthur Greenwood et al., « Probability weighted moments: definition and relation to parameters of several distributions expressable in inverse form », *in*: *Water resources research* 15.*5* (1979), pp. 1049–1054 (cit. on p. 28).

[58] Catalin Grigoras, « Applications of ENF criterion in forensic audio, video, computer and telecommunication analysis », *in*: *Forensic science international* 167.*2-3* (2007), pp. 136–145 (cit. on p. 22).

[59] Frank E. Grubbs, « Procedures for detecting outlying observations in samples », *in*: *Technometrics* 11.*1* (1969), pp. 1–21 (cit. on pp. 41, 117).

[60] Frank E Grubbs, « Sample criteria for testing outlying observations », *in*: *The Annals of Mathematical Statistics* 21.*1* (1950), pp. 27–58 (cit. on p. 23).

[61] Emil Julius Gumbel, *Statistical theory of extreme values and some practical applications: a series of lectures*, vol. 33, US Government Printing Office, 1948 (cit. on p. 24).

[62] Chuan Guo et al., « On calibration of modern neural networks », *in*: *ICML*, 2017, 1321::10 (cit. on pp. 94, 160, 166).

[63] Kaiming He et al., « Deep residual learning for image recognition », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778 (cit. on p. 85).

[64] Kaiming He et al., « Spatial pyramid pooling in deep convolutional networks for visual recognition », *in*: *IEEE transactions on pattern analysis and machine intelligence* 37.*9* (2015), pp. 1904–1916 (cit. on p. 81).

[65] Donald O. Hebb, « The organization of behavior; a neuropsychological theory. », *in*: (1949) (cit. on p. 14).

[66] Kevin E. Hemsley and Ronald E. Fisher, *History of industrial control system cyber incidents*, tech. rep., Idaho National Lab.(INL), Idaho Falls, ID (United States), 2018 (cit. on p. 41).

[67] Dan Hendrycks and Kevin Gimpel, « A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks », *in*: *ICLR*, 2017, URL: `https://openreview.net/forum?id=Hkg4TI9xl` (cit. on p. 95).

[68] Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean, « Distilling the Knowledge in a Neural Network », *in*: *stat* 1050 (2015), p. 9 (cit. on p. 94).

[69] Hui Huang et al., « Blind integrity verification of medical images », *in*: *IEEE transactions on information technology in biomedicine* 16.*6* (2012), pp. 1122–1126 (cit. on p. 18).

[70] Ibrahim Ethem Bagci, Utz Roedig, Ivan Martinovic, Matthias Schulz, Matthias Hollick, « Using Channel State Information for Tamper Detection in the Internet of Things », *in*: (2015) (cit. on p. 22).

[71] Jun Inoue et al., « Anomaly detection for a water treatment system using unsupervised machine learning », *in*: *2017 IEEE international conference on data mining workshops (ICDMW)*, IEEE, 2017, pp. 1058–1065 (cit. on p. 67).

[72] Alekseĭ Grigorevich Ivakhnenko and Valentin Grigorévich Lapa, *Cybernetic predicting devices*, tech. rep., PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGINEERING, 1966 (cit. on p. 14).

[73] Markus Jakobsson, Susanne Wetzel, and Bülent Yener, « Stealth attacks on ad-hoc wireless networks », *in*: *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)*, vol. 3, IEEE, 2003, pp. 2103–2111 (cit. on p. 42).

[74] Jeyanthi Hall, Michel Barbeau, Evangelos Kranakis, « Detection of transient in radio frequency fingerprinting using signal phase », *in*: (2003) (cit. on p. 22).

[75] Jie Xiong, Kyle Jamieson, « SecureAngle: Improving Wireless Security Using Angle-of-Arrival Information », *in*: (2010) (cit. on p. 22).

[76] Micah K Johnson and Hany Farid, « Exposing digital forgeries in complex lighting environments », *in*: *IEEE Transactions on Information Forensics and Security* 2.*3* (2007), pp. 450–461 (cit. on pp. 20, 22).

[77] Donggyu Joo, Junho Yim, and Junmo Kim, « Unconstrained Control of Feature Map Size Using Non-integer Strided Sampling », *in*: (2018) (cit. on pp. 70, 71).

[78] Kai Zeng, Kannan Govindan, Prasant Mohapatra, « Non-cryptographic authentication and identification in wireless networks », *in*: (2010) (cit. on p. 20).

[79] Myeong K Kang et al., « Enhancing Inter-Class Representation with a New Global Center Loss », *in*: *Proceedings of the 2017 International Conference on Industrial Design Engineering*, ACM, 2017, pp. 112–115 (cit. on pp. 75, 76).

[80] Gil Keren, Nicholas Cummins, and Björn Schuller, « Calibrated prediction intervals for neural network regressors », *in*: *IEEE Access* 6 (2018), p. 9 (cit. on p. 94).

[81] Nitin Khanna et al., « A survey of forensic characterization methods for physical devices », *in*: *digital investigation* 3 (2006), pp. 17–28 (cit. on p. 22).

[82] Diederik P Kingma and Jimmy Ba, « Adam: A method for stochastic optimization », *in*: *arXiv preprint arXiv:1412.6980* (2014) (cit. on pp. 58, 106, 159).

[83] Istvan Kiss, Bela Genge, and Piroska Haller, « A clustering-based approach to detect cyber attacks in process control systems », *in*: *2015 IEEE 13th international conference on industrial informatics (INDIN)*, IEEE, 2015, pp. 142–148 (cit. on p. 18).

[84] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy, « Remote physical device fingerprinting », *in*: *IEEE Transactions on Dependable and Secure Computing* 2.*2* (2005), pp. 93–108 (cit. on p. 22).

[85] Moshe Kravchik and Asaf Shabtai, « Detecting cyber attacks in industrial control systems using convolutional neural networks », *in*: *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, ACM, 2018, pp. 72–83 (cit. on p. 67).

[86] Hans-Peter Kriegel, Arthur Zimek, et al., « Angle-based outlier detection in high-dimensional data », *in*: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, pp. 444–452 (cit. on p. 82).

[87] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, « Simple and scalable predictive uncertainty estimation using deep ensembles », *in*: *NIPS*, 2017, 6402::12 (cit. on p. 95).

[88] J Maciunas Landwehr, NC Matalas, and JR Wallis, « Probability weighted moments compared with some traditional techniques in estimating Gumbel parameters and quantiles », *in*: *Water Resources Research* 15.*5* (1979), pp. 1055–1064 (cit. on p. 29).

[89] Lucien Le Cam et al., « An approximation theorem for the Poisson binomial distribution. », *in*: *Pacific Journal of Mathematics* 10.*4* (1960), pp. 1181–1197 (cit. on p. 37).

[90] M. Ross Leadbetter, *Extremes and Local Dependence in Stationary Sequences.* Tech. rep., North Carolina Univ at Chapel Hill Dept of Statistic, 1983 (cit. on p. 28).

[91] M. Ross Leadbetter, « On extreme values in stationary sequences », *in*: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 28.*4* (1974), pp. 289–303 (cit. on p. 28).

[92] Yann LeCun, *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*, Université Pierre et Marie Curie (Paris 6), June 1987 (cit. on p. 15).

[93] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton, « Deep learning », *in*: *nature* 521.*7553* (2015), pp. 436–444 (cit. on p. 15).

[94] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges, *MNIST handwritten digit database. AT&T Labs*, 2010 (cit. on pp. 159, 160).

[95] Yann LeCun et al., « Handwritten digit recognition with a back-propagation network », *in*: *Advances in neural information processing systems*, 1990, 396::9 (cit. on p. 157).

[96] Kimin Lee et al., « Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples », *in*: *ICLR*, 2018, URL: `https://openreview.net/forum?id=ryiAv2xAZ` (cit. on pp. 95, 105).

[97] Shiyu Liang, Yixuan Li, and R. Srikant, « Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks », *in*: *ICLR*, 2018, URL: `https://openreview.net/forum?id=H1VGkIxRZ` (cit. on pp. 95, 166).

[98] Qin Lin et al., « TABOR: A graphical model-based approach for anomaly detection in industrial control systems », *in*: *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 525–536 (cit. on p. 67).

[99] Jie Liu et al., « Anomaly detection in manufacturing systems using structured neural networks », *in*: *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, IEEE, 2018, pp. 175–180 (cit. on p. 18).

[100] Hélène Lubes and JM Masson, « Méthode des moments de probabilité pondérés: application à la loi de Jenkinson », *in*: *Hydrologie continentale* 6.*1* (1991), pp. 67–84 (cit. on p. 29).

[101] Mark Luchs and Christian Doerr, « Last line of defense: A novel ids approach against advanced threats in industrial control systems », *in*: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, 2017, pp. 141–160 (cit. on p. 18).

[102] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan, « Detecting digital image forgeries using sensor pattern noise », *in*: *Security, Steganography, and Watermarking of Multimedia Contents VIII*, vol. 6072, International Society for Optics and Photonics, 2006, 60720Y (cit. on p. 53).

[103] Pankaj Malhotra et al., « LSTM-based encoder-decoder for multi-sensor anomaly detection », *in*: *arXiv preprint arXiv:1607.00148* (2016) (cit. on p. 18).

[104] Henry B Mann and Donald R Whitney, « On a test of whether one of two random variables is stochastically larger than the other », *in*: *The annals of mathematical statistics* (1947), pp. 50–60 (cit. on p. 88).

[105] Bertrand Masset and Olivier Taburiaux, « Simulating Industrial Control Systems Using Mininet », *in*: (2017) (cit. on p. 43).

[106] Matti Mantere, Ilkka Uusitalo, Mirko Sailio, Sami Noponen, « Challenges of Machine Learning Based Monitoring for Industrial Control System Networks », *in*: (2012) (cit. on p. 17).

[107] Warren S. McCulloch and Walter Pitts, « A logical calculus of the ideas immanent in nervous activity », *in*: *The bulletin of mathematical biophysics* 5.*4* (1943), pp. 115–133 (cit. on p. 14).

[108] Aline Mefleh et al., « Permutation bootstrap and the block maxima method », *in*: *Communications in Statistics-Simulation and Computation* (2019), pp. 1–17 (cit. on p. 29).

[109] Nicholas Metropolis et al., « Equation of state calculations by fast computing machines », *in*: *The journal of chemical physics* 21.*6* (1953), pp. 1087–1092 (cit. on p. 30).

[110] Marvin Minsky and Seymour Papert, « Perceptrons: An introduction to computational geometry », *in*: (1969) (cit. on pp. 14, 96).

[111] R von Mises, « La distribution de la plus grande de n valeurs », *in*: *Revue Math6matique de l'union interbalkanique* 1 (1936), pp. 141–160 (cit. on p. 24).

[112] Higinio Mora et al., « Distributed computational model for shared processing on Cyber-Physical System environments », *in*: *Computer Communications* 111 (2017), pp. 68–83 (cit. on p. 41).

[113] Mathilde Mougeot, Robert Azencott, and Bernard Angeniol, « Image Compression with Back propagation: Improvement of the Visual Restoration using Different Cost Functions », *in*: *Neural networks* 4.*4* (1991), p. 10 (cit. on p. 59).

[114] Azadeh Sadat Mozafari et al., « Unsupervised Temperature Scaling: Post-Processing Unsupervised Calibration of Deep Models Decisions », *in*: *arXiv preprint arXiv:1905.00174* (2019) (cit. on p. 94).

[115] Murat Demirbas, Youngwhan Song, « An RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks », *in*: (2006) (cit. on p. 22).

[116] Radford M Neal, « Slice sampling », *in*: *The annals of statistics* 31.*3* (2003), pp. 705–767 (cit. on p. 30).

[117] Yuval Netzer et al., « Reading digits in natural images with unsupervised feature learning », *in*: (2011) (cit. on p. 159).

[118] Tian-Tsong Ng et al., « Physics-motivated features for distinguishing photographic images and computer graphics », *in*: *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 239–248 (cit. on pp. 20, 22).

[119] Anh Nguyen, Jason Yosinski, and Jeff Clune, « Deep neural networks are easily fooled: High confidence predictions for unrecognizable images », *in*: *CVPR*, 2015, 427::10 (cit. on p. 94).

[120] Alexandru Niculescu-Mizil and Rich Caruana, « Predicting good probabilities with supervised learning », *in*: *ICML*, 2005, 625::8 (cit. on p. 94).

[121] Paul O'Leary, Matthew Harker, and Richard Neumayr, « Savitzky-Golay smoothing for multivariate cyclic measurement data », *in*: *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, IEEE, 2010, pp. 1585–1590 (cit. on p. 121).

[122] Yaniv Ovadia et al., « Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift », *in*: *Advances in Neural Information Processing Systems*, 2019, pp. 13969–13980 (cit. on p. 95).

[123] Nicolas Papernot and Patrick McDaniel, « Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning », *in*: *arXiv preprint arXiv:1803.04765* (2018) (cit. on pp. 95, 165).

[124] Fabio Pasqualetti, Florian Dörfler, and Francesco Bullo, « Attack detection and identification in cyber-physical systems », *in*: *IEEE transactions on automatic control* 58.*11* (2013), pp. 2715–2729 (cit. on p. 18).

[125] Ce Qi and Fei Su, « Contrastive-center loss for deep neural networks », *in*: *arXiv preprint arXiv:1707.07391* (2017) (cit. on pp. 75, 76).

[126] Pierre Ribereau, Esterina Masiello, and Philippe Naveau, « Skew generalized extreme value distribution: Probability-weighted moments estimation and application to block maxima procedure », *in*: *Communications in Statistics-Theory and Methods* 45.*17* (2016), pp. 5037–5052 (cit. on p. 28).

[127] Sebastian Rohjans et al., « mosaik-A modular platform for the evaluation of agent-based Smart Grid control », *in*: *Innovative Smart Grid Technologies Europe (ISGT EUROPE), 2013 4th IEEE/PES*, IEEE, 2013, pp. 1–5 (cit. on p. 43).

[128] Paul R Rosenbaum, « An exact distribution-free test comparing two multivariate distributions based on adjacency », *in*: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.*4* (2005), pp. 515–530 (cit. on p. 87).

[129] Frank Rosenblatt, « The perceptron: a probabilistic model for information storage and organization in the brain. », *in*: *Psychological review* 65.*6* (1958), p. 386 (cit. on pp. 14, 96).

[130] Ryan M. Gerdes, Mani Mina, Steve F. Russell, Thomas E. Daniels, « Physical-Layer Identification of Wired Ethernet Devices », *in*: (2012) (cit. on pp. 19, 22).

[131] Takaya Saito and Marc Rehmsmeier, « The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets », *in*: *PLOS ONE* 10.*3* (Mar. 2015), pp. 1–21, DOI: 10.1371/journal.pone.0118432, URL: https://doi.org/10.1371/journal.pone.0118432 (cit. on p. 58).

[132] Sakthi Vignesh Radhakrishnan, A. Selcuk Uluagac, Raheem Beyah, « GTID: A Technique for Physical Device and Device Type Fingerprinting », *in*: (2014) (cit. on p. 22).

[133] Richard W Sanders, « Digital audio authenticity using the electric network frequency », *in*: *Audio Engineering Society Conference: 33rd International Conference: Audio Forensics-Theory and Practice*, Audio Engineering Society, 2008 (cit. on p. 22).

[134] Dhruv Mauria Saxena, Vince Kurtz, and Martial Hebert, « Learning robust failure response for autonomous vision based flight », *in*: *ICRA*, 2017, 5824::6 (cit. on p. 94).

[135] Jürgen Schmidhuber, « Deep learning in neural networks: An overview », *in*: *Neural networks* 61 (2015), pp. 85–117 (cit. on pp. 14, 15).

[136] Bernhard Schölkopf et al., « Support vector method for novelty detection », *in*: *Advances in neural information processing systems*, 2000, pp. 582–588 (cit. on p. 23).

[137] Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han, « Learning for single-shot confidence calibration in deep neural networks through stochastic inferences », *in*: *CVPR*, 2019, 9030::9 (cit. on pp. 94, 166).

[138] Alireza Shafaei, Mark Schmidt, and James J Little, « A Less Biased Evaluation of Out-of-distribution Sample Detectors », *in*: (2019) (cit. on pp. 79, 85).

[139] Chandra Shekhar, « On Simplified Application of Multidimensional Savitzky-Golay Filters and Differentiators », *in*: *American Institute of Physics Conference Series*, vol. 1705, 2, 2016 (cit. on p. 121).

[140] Alban Siffer et al., « Anomaly detection in streams with extreme value theory », *in*: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1067–1075 (cit. on p. 23).

[141] Ridha Soua, « Wireless sensor networks in industrial environment: energy efficiency, delay and scalability », PhD thesis, 2014 (cit. on p. 13).

[142] Matthew C Stamm, Min Wu, and KJ Ray Liu, « Information forensics: An overview of the first decade », *in*: *IEEE access* 1 (2013), pp. 167–200 (cit. on pp. 20, 22).

[143] R.G. Stanton, « The work of L.H.C. Tippett », *in*: *Ars Textrina* 7 (1987), pp. 179–185 (cit. on p. 24).

[144] Suman Jana, Sneha K. Kasera, « On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews », *in*: (2008) (cit. on p. 22).

[145] Albert Thomas, Vincent Feuillard, and Alexandre Gramfort, « Calibration of One-Class SVM for MV set estimation », *in*: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, 2015, pp. 1–9 (cit. on p. 23).

[146] Emre Tokgöz, Sara Nourazari, and Hillel Kumin, « Convexity and Optimization of Condense Discrete Functions », *in*: *International Symposium on Experimental Algorithms*, Springer, 2011, pp. 33–42 (cit. on pp. 121, 122).

[147] John Von Neumann, « Various techniques used in connection with random digits », *in*: *Monte Carlo Method* 12 (1951) (cit. on p. 30).

[148] Vladimir Vovk, Alex Gammerman, and Glenn Shafer, *Algorithmic learning in a random world*, Springer Science & Business Media, 2005 (cit. on p. 95).

[149] Waloddi Weibull, « A statistical theory of strength of materials », *in*: *IVB-Handl.* (1939) (cit. on p. 24).

[150] Yandong Wen et al., « A discriminative feature learning approach for deep face recognition », *in*: *European Conference on Computer Vision*, Springer, 2016, pp. 499–515 (cit. on pp. 70, 72, 75).

[151] Paul J Werbos, « Applications of advances in nonlinear sensitivity analysis », *in*: *System modeling and optimization*, Springer, 1982, pp. 762–770 (cit. on p. 15).

[152] Bernard Widrow and Marcian E.T. Hoff, *Adaptive switching circuits*, tech. rep., Stanford University, California, Stanford Electronics Labs, 1960 (cit. on p. 15).

[153] Frank Wilcoxon, « Individual Comparisons by Ranking Methods », *in*: *Biometrics* 1.*6* (1945), pp. 80–83 (cit. on p. 88).

[154] Christian Wressnegger, Ansgar Kellner, and Konrad Rieck, « ZOE: Content-based Anomaly Detection for Industrial Control Systems », *in*: *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, IEEE, 2018, pp. 127–138 (cit. on pp. 13, 18).

[155] Yue Wu et al., « Deep face recognition with center invariant loss », *in*: *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, ACM, 2017, pp. 408–414 (cit. on pp. 75, 76).

[156] Jie Xiong and Kyle Jamieson, « Securearray: Improving wifi security with fine-grained physical-layer information », *in*: *Proceedings of the 19th annual international conference on Mobile computing & networking*, ACM, 2013, pp. 441–452 (cit. on p. 22).

[157] Jie Yang et al., « Detection and localization of multiple spoofing attackers in wireless networks », *in*: *IEEE Transactions on Parallel and Distributed systems* 24.*1* (2013), pp. 44–58 (cit. on p. 22).

[158] Zang Li, Wenyuan Xu, Rob Miller, Wade Trappe, « Securing Wireless Systems via Lower Layer Enforcements », *in*: (2006) (cit. on p. 22).

[159] Junxing Zhang et al., « Advancing wireless link signatures for location distinction », *in*: *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ACM, 2008, pp. 26–37 (cit. on p. 22).

# Appendices

Appendix A

# Details on Experimental data

## A.1 Real data from the testbed SWaT

The SWaT dataset consists of an 11 days record of sensors and actuators, 51 in total sampled every second, and network captures from a water treatment system that «*replicates large modern plants for water treatment such as those found in cities*»[50]. As already mentioned, we focus on the physical process, so we do not work on network captures. During the last 4 days, 41 attacks were performed, 36 of which implied physical data. In order for our model to be able to use these data, we preprocessed them as following. First we define the training set and the testing set: the training set starts with the 19905-th sample point (before that point the system is not stabilized) and ends with the 389853-th sample point, the testing set consists of the rest of the normal operation dataset (label 0) and the dataset of the attacks from the moment of the first attack (1757-th point in the dataset of attacks) until the end (label 1). Attacks, detailed in [50], are of four types called Single/Multi Stage Single/Multi Point Attacks depending on whether one or several sensor(s)/actuator(s) are in involved and whether they are from a single stage or different stages of the process. Each attribute is normalized with the min-max normalization from the training set. The system operation is cyclic, and we define states as explained in Section 2.3. The easiest way to define cycles is to rely on the actuator MV-301 values, a cycle starts and ends as soon as MV-301 value changes from 0 to 2. Then, we define 12 states by retaining time windows of size 600 (covering 10 minutes of operation) positioned in a cycle relatively to its length. At testing time, states are ignored and anomaly scores are determined as explained in Section 2.5 with time windows of reconstruction errors of size 366 (that is the ceil of the length of the greatest cycle in the training set divided by 12). We did not define a state overlapping two subsequent cycles, which would have aimed at modeling dependencies between two subsequents cycles, because there is not enough cycles to do so (the training set contains only 119 cycles).

## A.2 Visualization of cycles simulated with Sibriz

The simulation model of cyclic industrial systems with binary registers (sensor values, actuators commands, and automata responses are binary) is presented in chapter 2. Whereas we showed that we can produce with a same template different cycles (since our model can detect attacks), we did not dwell on the legitimacy of the simulation model. In particular, one can expect from such model that 1) it can produce varying cycles with some deterministic behaviors, 2) it can produce a normal regime and an abnormal regime that seems similar to the naked eye.

That is why we plot in this appendix fifty cycles from the normal regime (Figure A.1) and fifty cycles from the brutal attacks of type i) (Figure A.2). We can easily see the effect of retention points defined in section 2.2 as well as the difference in the retention points of the normal and the abnormal regime. For example, at the bottom left of each cycle, in the normal regime, there is always two succesive bars separated by one blank point, while in the abnormal regime, there is sometimes only one long bar as the one circled in red in Figure A.2. This is an example of deterministic behavior that the attack does not respect. Of course, there are more subtle deviations in theseé abnormal cycles that only anomaly detection models can detect.

It is to the best of our knowledge the first simulation of physical processes inducing binary values in a probabilistically realistic way.

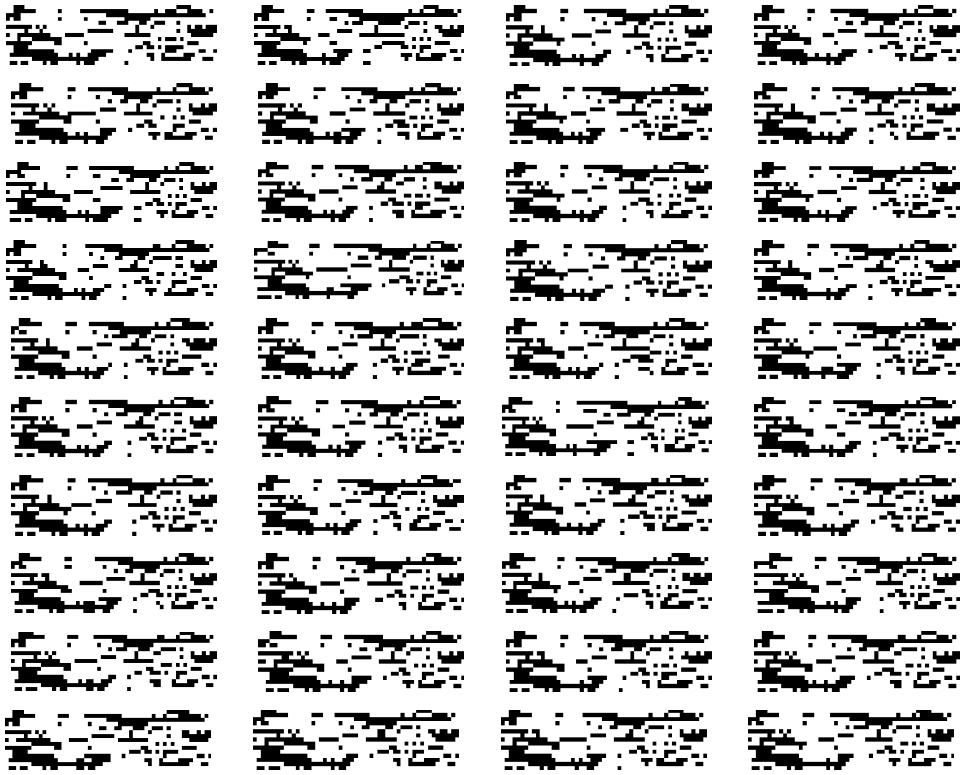


Figure A.1 – Normal cycles simulated with Sibriz, height corresponds to registers length corresponds to time, black dots are for value 1, white dots are for value 0. The cycles are of different sizes but for a better visualization, the images are plotted in the same dimensions.

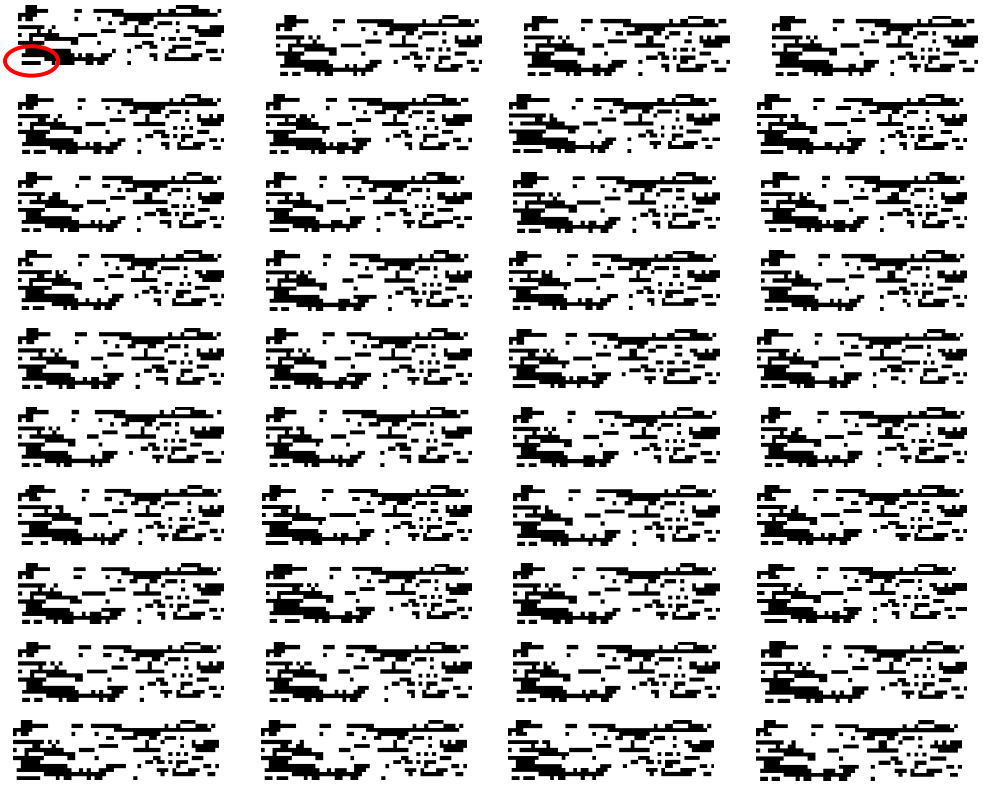


Figure A.2 – Cycles from the brutal attacks of type i) simulated with Sibriz on the same template than the one from Figure A.1, height corresponds to registers length corresponds to time, black dots are for value 1, white dots are for value 0. The cycles are of different sizes but for a better visualization, the images are plotted in the same dimensions. See the effect of the attack on the pattern in the red circle in comparison to cycles in Figure A.1.

Appendix B

# A simple method to approximate the anti-function of a bijection

$$(f^{-1})'(y) = \frac{1}{f'(f^{-1}(y))} \tag{B.1}$$

To approximate the anti-function of a bijection, it is natural to build on the well-known formula of the derivative of the anti-function (B.1). Indeed, through integration, the derivative leads to the anti-function. Of course, this formula relies on the derivative of the function itself but also on the anti-function, it's the snake that bites its own tail. Fortunately, it turns out that the sequence of the integral of the inverse of the derivative composed with the previous element and starting this sequence with the identity often converges to the anti-function. Without loss of generality, we can assume that the function $f$ has a domain and an image both equal to $[0, 1]$, since if it is not the case, one can consider $g = h_1 \circ f \circ h_2$ with $h_i$ the linear functions such that the domain of $g$ is $[0, 1]$, $\min(g) = 0$, and $\max(g) = 1$, then once $g^{-1}$ is approximated by $\widehat{g^{-1}}$, $f^{-1}$ is simply approximated by $h_2 \circ \widehat{g^{-1}} \circ h_1$. More formally, let $u_0 = Id$, and $u_n = \hat{V}_n$ an estimation of

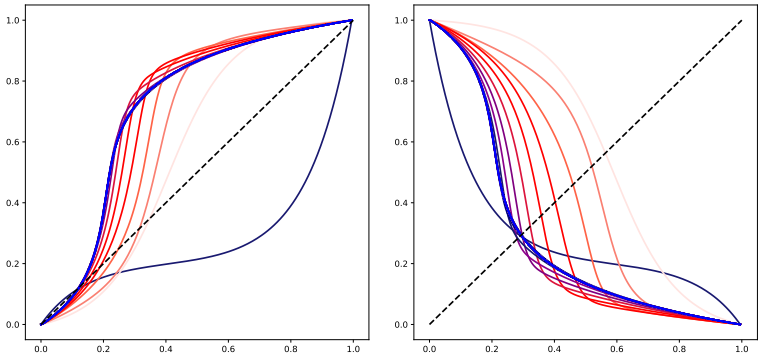


Figure B.1 – Approximation of anti-functions thanks to the sequence of the integral of the inverse of the derivative composed with the previous element. (100 iterations)

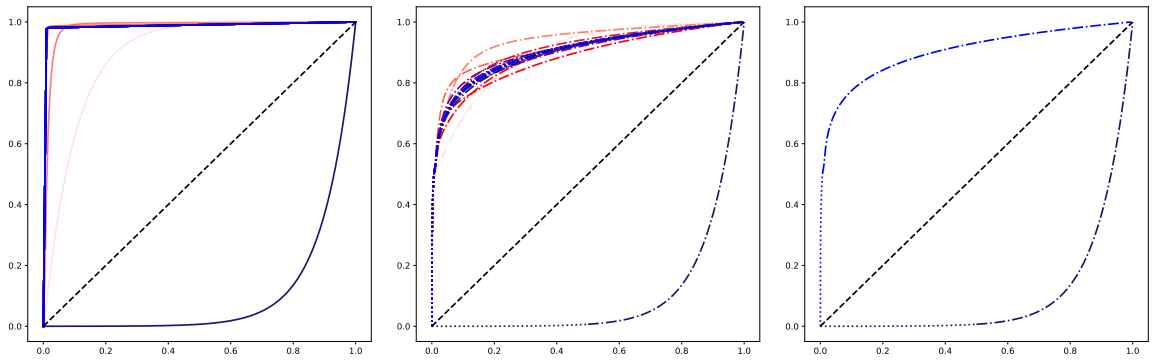


Figure B.2 – Bad approximation of the anti-function of a function with some low derivatives from the whole function itself (left), concatenation of two independent good approximations of two parts of the function (middle) and end result of the second method (right). (100 iterations)

$V_n : x \in [0,1] \to \int_0^x \frac{1}{f'(u_{n-1}(t))}dt$. To have a good approximation of $V_n$, it is useful to define $\hat{V}_n$ by computing the trapezoidal sum to approximate the area of $\frac{1}{f'(u_{n-1}(t))}$ from the right $(a_n)$ and from the left $(b_n)$, with $N$ a normalizing functional:

$$u_n = \frac{1}{2}(N(a_n + \mathbb{1}_{f'<0}) + N(b_n - b_n(0) + \mathbb{1}_{f'<0})) \tag{B.2}$$
$$a_n : x \in [0,1] \to \widehat{\int_0^x} \frac{1}{f'(u_{n-1}(t))}dt; \quad b_n : x \in [0,1] \to -\widehat{\int_x^1} \frac{1}{f'(u_{n-1}(t))}dt;$$

with $\hat{\int}$ the trapezoidal estimation of the integral. In pratice, we used as $N$, $u \to \frac{u-\min(u)}{\max(u)-\min(u)}$ ($u_n = 0.5 \times (N(a_n) + N(b_n))$ is thus sufficient) but the algorithm is only proved to converge for a probably less efficient functional $u \to (t \to u(t).\mathbb{1}_{0<u(t)\leq 1} + \mathbb{1}_{u(t)>1})$ (cf. Proposition B.1). If $|f'|$ has not *too much values near zero or very high*, then $u_n$ converges towards $f^{-1}$ as in Figure B.1. We will not pursue what is the meaning of "too much values near zero or very high", but we will give a simple method to overcome this issue. We can see in the first graph of Figure B.2 that because of the first part of the function, which has almost no slope, the sequence converges towards the constant function $x \to 1$. Indeed, the inverse of very low values of the derivatives leads to dramatically bad approximations. A simple way to overcome this is to approximate the anti-functions of different parts of the function and merge them to obtain the anti-function of the whole as shown in the other graphs of Figure B.2. Since these parts of the function does not verify $[0,1] \to [0,1]$, they are normalized and the anti-function of the original function is retrieved as previously explained ($g = h_1 \circ f \circ h_2 \implies f^{-1} = h_2 \circ g^{-1} \circ h_1$). In Figure B.2, two parts of the function are sufficient, but in other cases more parts may be necessary.

**Lemma B.1.** *Let $f : \mathbb{R} \to \mathbb{R} \in \mathcal{C}^1$ such that $f([0,1]) = [0,1]$, $\mathcal{C}^{1*}$ the set of functions differentiable everywhere on $\mathbb{R}$ except maybe on a finite set of points and functionals $A_f : u \in \mathcal{C}^{1*} \to (x \to \int_0^x \frac{1}{f'(u(t))} dt + \mathbb{1}_{f'_{]0,1[}<0})$ and $N : u \in \mathcal{C}^{1*} \to (t \to u(t).\mathbb{1}_{0<u(t)\leq 1} + \mathbb{1}_{u(t)>1})$. If $\sup_{x\neq y}(|\frac{f'(x)-f'(y)}{x-y}|) < \min(|f'|)^2 \vee (f \in \mathcal{C}^2 \wedge \max(|f''|) < \min(|f'|)^2)$, then $A_f$ and $A_f \circ N$ are contraction mappings on $\mathcal{C}^{1*}$ with respect to the norm $\|.\|_{L^1([0,1])} : f \to \int_0^1 |f(t)|dt$.*

*Proof of Lemma B.1.* Let $(u, v) \in \mathcal{C}^1([0,1])^2$, two functions between 0 and 1, we have: $\|A_f(u) - A_f(v)\|_{L^1([0,1])} \leq \|A_f(u) - A_f(v)\|_{L^\infty([0,1])} \leq \|\frac{1}{f'\circ u} - \frac{1}{f'\circ v}\|_{L^1([0,1])}$.
The first inequality comes from $\|.\|_{L^1([a,b])} \leq (b-a)\|.\|_{L^\infty([a,b])}$, and the second inequality comes from the triangle inequality that yields $\sup_{x\in[0,1]}(\int_0^x |\frac{1}{f'(u(t))} - \frac{1}{f'(v(t))}|dt)$ on the right-hand side of the inequality, whose supremum is reached for $x = 1$, leading to $\|.\|_{L^1([0,1])}$. Hence, we have: $\|A_f(u) - A_f(v)\|_{L^1([0,1])} \leq k.\|u - v\|_{L^1([0,1])}$, with, when $\min(f') \neq 0$:

$$k = \sup_{x\neq y}(|\frac{\frac{1}{f'(x)} - \frac{1}{f'(y)}}{x-y}|) \leq \sup_{x\neq y}(|\frac{f'(x) - f'(y)}{f'(x)f'(y)(x-y)}|) \leq \frac{1}{\min((f')^2)} \sup_{x\neq y}(|\frac{f'(x) - f'(y)}{x-y}|)$$

As for $A_f \circ N$ when $u([0,1]) \neq [0,1]$ or $v([0,1]) \neq [0,1]$, one can notice that $A_f \circ N$ is the same as $A_f$ except that, given $\bar{} : g \to (t \to \bar{g}(t) = g(0).\mathbb{1}_{t<0} + g(t).\mathbb{1}_{0\leq t\leq 1} + g(1).\mathbb{1}_{t>1})$, one has to replace $f'$ with $\overline{f'}$. Then, $\sup_{\substack{x\neq y \\ (x,y)\in(u([0,1])\cup v([0,1]))^2}} (|\frac{1/\overline{f'}(x) - 1/\overline{f'}(y)}{x-y}|) = k$ closes the proof. □

**Proposition B.1.** *If $f : [0,1] \to [0,1] \in \mathcal{C}^2$ such that $|f'| > 0$ and $f([0,1]) = [0,1]$, then there exists a partition of $[0,1]$ composed of intervals $(I_k)_{1\leq k\leq K}$, such that the $K$ series $((u_{k,n})_n)_k$ defined as in (B.2) with $N : (u : [0,1] \to \mathbb{R}) \to (t \to u(t).\mathbb{1}_{0<u(t)\leq 1} + \mathbb{1}_{u(t)>1})$ for functions $(g_k)_k$ obtained by applying linear transformations $h_{1,k}$ and $h_{2,k}$ this way: $\forall k, g_k = h_{1,k} \circ f_{|I_k} \circ h_{2,k}$, $h_{2,k}([0,1]) = I_k$, $g_k([0,1]) = [0,1]$, allow building a function series $(h_{2,1} \circ u_{n,1} \circ h_{1,1}{}^\frown h_{2,2} \circ u_{n,2} \circ h_{1,2}{}^\frown \ldots{}^\frown h_{2,K} \circ u_{n,K} \circ h_{1,K})_n$, with $^\frown$ the symbol for concatenation of functions, that converges, in norm on $L^1([0,1])$, towards $f^{-1}$ while the trapezoidal approximation of the integrals is refined.*

*Proof of Proposition B.1.* First of all, let us remark that in order to prove Proposition B.1, it suffices to prove that the algorithm works when only the normalized version of $a_n$ is used instead of the average of the normalized versions of $a_n$ and $b_n$, since they are equal modulo some trapezoidal appromixation errors that can be made as small as possible.

Let $f : [0,1] \to [0,1]$ be such that $|f'| > 0$ (thus monotonic) and $f([0,1]) = [0,1]$, $h_1$ and $h_2$ two linear transformations such that $g = h_1 \circ f \circ h_2$ is defined on $[0,1]$ and span $[0,1]$.

Then, $h_2(x) = a.x + b$ such that $a \in ]0,1[$ and: $g(x) = \frac{f(a.x+b)-\min_x(f(a.x+b))}{\max_x(f(a.x+b))-\min_x(f(a.x+b))}$, $g'(x) = \frac{a.f'(a.x+b)}{\max_x(f(a.x+b))-\min_x(f(a.x+b))}$, and $g''(x) = \frac{a^2.f''(a.x+b)}{\max_x(f(a.x+b))-\min_x(f(a.x+b))}$. Thus, $|g''(x)| < |g'(x)|^2 \iff |f''(a.x+b)| < \frac{|f'(a.x+b)|^2}{\max_x(f(a.x+b))-\min_x(f(a.x+b))}$. Given $x_0 \in ]0,1[$, we can thus find $a \in ]0,1[$ and $b \in \mathbb{R}$ such that $|g''(x_0)| < |g'(x_0)|^2$, it suffices to lower $a$ and thus also $\max_x(f(a.x+b)) - \min_x(f(a.x+b))$ while keeping $a.x_0+b$, and thus $f'(a.x_0+b)$ and $f''(a.x_0+b)$, constant by adjusting $b$. By continuity of $g'$ and $g''$, there is a closed interval $I = [d_1, d_2] \ni x_0$, such that $\max_{x\in I}(|g''(x))| < \min_{x\in I}(|g'(x)|^2)$ and $d_2 - d_1 < a$. Redefining $a$ and $b$ s.t. $I = [b, a+b]$, does not change $\max_{x\in I}(|g''(x))| < \min_{x\in I}(|g'(x)|^2)$.

Since $g'' \in \mathcal{C}^0$, $\max_{x\in[0,1]}(g'')$ exists and thus, thanks to $0 < |f'| < \infty$ ($f \in \mathcal{C}^2$), one can repeat this procedure to obtain a finite partition of $[0,1]$ from intervals $(I_k)_{1\leq k\leq K}$, modulo their closures, and corresponding functions $g_k$ with $g_k = h_{1,k} \circ f_{|I_k} \circ h_{2,k}$, and proper linear transformations $h_{1,k}$ and $h_{2,k}$ as specified in Proposition B.1. Since $A_{g_k}$ from Lemma B.1 is a contraction mapping with respect $\|.\|_{L^{[0,1]}}$, and since $L^1([0,1])$ is a Banach space, $g_k^{-1}$ is the unique fixed point of $A_{g_k}$. Since $(I_k)_{1\leq k\leq K}$ is a finite partition of $[0,1]$, modulo the closures, the concatenated functions of Proposition B.1 can be computed in parallel. So without loss of generality, we can assume $K=1$.

Let $\hat{A}_f : u \in \{v \in \mathcal{C}^1 \mid v([0,1]) = [0,1]\} \to (x \in [0,1] \to \widehat{\int_0^x} \frac{1}{f'(u(t))}dt + \mathbb{1}_{f'<0})$. Let $N$ be the functional $N : (u : [0,1] \to \mathbb{R}) \to (t \to u(t).\mathbb{1}_{0<u(t)\leq 1} + \mathbb{1}_{u(t)>1})$. In Lemma B.1's proof, replacing the integral by its trapezoidal approximation $\hat{\int}$ leads to a similar inequality: $\|\tilde{A}_f(u) - \tilde{A}_f(v)\|_{L^1([0,1])} \leq \|\frac{1}{f'\circ v} - \frac{1}{f'\circ u}\|_{L^1([0,1])} + \epsilon \leq k.\|u-v\|_{L^1([0,1])} + \epsilon$, with $\epsilon$ from the trapezoidal approximation errors that can be made as small as desired and $\tilde{A}_f = \hat{A}_f \circ N$. We have $\tilde{A}_f(f^{-1}) = \hat{A}_f(f^{-1}) = A_f(f^{-1}) + \varepsilon = f^{-1} + \varepsilon$, with $\varepsilon$ the trapezoidal approximation error function. Let us define $u_0 = Id$ and $\forall i > 0$, $u_{i+1} = \tilde{A}_f(u_i)$.

Let $n \in \mathbb{N}$ such that $k + \frac{1}{n} < 1$, and $\epsilon' = \epsilon + \|\varepsilon\|_{L^1([0,1])} < \min_{i\leq n}(\frac{1}{n}\|u_i - f^{-1}\|_{L^1([0,1])})$, being as small as desired. So, we have

$$\begin{aligned}\|u_{n+1} - f^{-1}\|_{L^1([0,1])} &= \|\tilde{A}_f(u_n) - \tilde{A}_f(f^{-1}) + \varepsilon\|_{L^1([0,1])} \leq k.\|u_n - f^{-1}\|_{L^1([0,1])} + \epsilon' \\ &\leq (k + \frac{1}{n})\|u_n - f^{-1}\|_{L^1([0,1])} \leq (k + \frac{1}{n})^{n+1}\|u_0 - f^{-1}\|_{L^1([0,1])}\end{aligned}$$

Then, if one wants to continue to decrease $\|u_l - f^{-1}\|_{L^1([0,1])}$ for $l > n+1$, one has to repeat this procedure the same way with even smaller $\epsilon'$, $u_{n+1}$ acting as $u_0$. Assuming an infinite precision capacity for the trapezoidal estimation of the integral, one can thus make the series from our algorithm converges towards $f^{-1}$ in norm $\|.\|_{L^1([0,1])}$. □

Appendix C

# A measure of the level of linear separability

In this appendix, we introduce a notion of liaison between a subset of $\mathbb{R}^n$ and a set, $\mathcal{E}$, of subsets of $\mathbb{R}^n$. Then considering the liaison to $\mathcal{E}$ of every elements of $\mathcal{E}$ should lead to a difficulty for an algorithm to find hyperplanes that allows the classification in $\mathcal{E}$.

**Definition C.1.** *Let $\mathcal{E}$ be a set of subsets of $\mathbb{R}^n$ such that for any couple $(e_1, e_2)$ in $\mathcal{E}^2$, there is a hyperplane that separates $e_1$ from $e_2$. A subset $c$ of $\mathbb{R}^n$ is said to be $k$-linked to $\mathcal{E}$ if there is a hyperplane that separates $c$ from the union of $|\mathcal{E} \setminus \{c\}| - k$ element(s) of $\mathcal{E}$ without intersecting the convex hull of any set of $\mathcal{E}$. A subset $c$ of $\mathbb{R}^n$ is said to have a degree of liaison $d$ to $\mathcal{E}$ if it is $d$-linked to $\mathcal{E}$ and if $d$ is minimal against this property. The set $\mathcal{E}$ is said to be plainly (resp. complexly) linearly separable if each of its elements has a degree of liaison to itself equal to 0 (resp. $|\mathcal{E}| - 1$).*

**Definition C.2.** *Let $\mathcal{E}$ be a set of subsets of $\mathbb{R}^n$. A set $\mathcal{H}$ of hyperplanes. We call $\mathcal{H}$-cell any open convex subset of $\mathbb{R}^n$ delineated by elements of $\mathcal{H}$. If the knowledge of the belonging of a point of a set $c \in \mathcal{E}$ to a $\mathcal{H}$-cell gives the knowledge of the set membership of this point within $c$: $\forall c \in \mathcal{E}, \exists (C_i)_{i=1,..,m},\ m$ $\mathcal{H}$-cells $\mid c \subset \bigcup_{i=1}^{m} C_i \wedge \bigcup(\mathcal{E} \setminus \{c\}) \subset \mathbb{R}^n \setminus \bigcup_{i=1}^{m} C_i$, $\mathcal{H}$ is said to be a solution to the linear separability of $\mathcal{E}$. It is said to be minimal if a solution of smaller size does not exist.*

**Lemma C.1.** *Let $\mathcal{E}$ be a set of subsets of $\mathbb{R}^n$. If $\mathcal{E}$ is plainly linearly separable then there is a solution to the linear separability of $\mathcal{E}$, $\mathcal{H}$, such that $|\mathcal{H}| \leq |\mathcal{E}| - 1$.*

*Proof of Lemma C.1.* Let $\mathcal{E}$ be a plainly linearly separable class, in other words, each elements of $\mathcal{E}$ has a degree of liaison to $\mathcal{E}$ equal to 0. Therefore, for each element $e \in \mathcal{E}$ we can find a hyperplane $h_e$ that separates it from the union of the other elements in $\mathcal{E}$. Let us take an arbitrary element $c \in \mathcal{E}$, then $\mathcal{H} = \{h_e \mid e \in \mathcal{E} \setminus \{c\}\}$ is a solution to the linear separability of $\mathcal{E} \setminus \{c\}$. Yet, $\forall e \in \mathcal{E} \setminus \{c\}$, $h_e$ separates $c$ from $e$. Hence, it is also a solution for the linear separability of $\mathcal{E}$ and $|\mathcal{H}| = |\mathcal{E} \setminus \{c\}| = |\mathcal{E}| - 1$, concluding the proof. □

**Lemma C.2.** *Let $\mathcal{E}$ be a set of subsets of $\mathbb{R}^n$. If $\mathcal{H}$ is a minimal solution to the linear separability of $\mathcal{E}$ such that there is $h \in \mathcal{H}$ that does not intersect the convex hull of any elements of $\mathcal{E}$ and whose one of the two half-spaces is a portion of the space where there is no pair of hyperplanes of $\mathcal{H}$ that intersect, then $|\mathcal{H}| > \min_{e\in\mathcal{E}}(d_{\mathcal{E}}(e))$ with $d_{\mathcal{E}}(e)$ the degree of liaison of $e$ to $\mathcal{E}$.*

*Proof of Lemma C.2.* Let $\mathcal{E}$ be a set of subsets of $\mathbb{R}^n$ and $\mathcal{H}$ a minimal solution to the linear separability of $\mathcal{E}$. Let $h \in \mathcal{H}$ be a hyperplane verifying the lemma hypothesis. So $h$ separates the union of a set $\mathcal{C} \subset \mathcal{E}$, from the union of the rest of the class. Since $\mathcal{H}$ is mininal, $\mathcal{C} \neq \emptyset$, we have $\min_{e\in\mathcal{E}}(d_{\mathcal{E}}(e)) \leq \min_{e\in\mathcal{C}}(d_{\mathcal{E}}(e)) < |\mathcal{C}|$ by definition of the degree of liaison. Since there is no pair of hyperplanes of $\mathcal{H}$ that intersect on the side of $h$ where $\bigcup\mathcal{C}$ is, each hyperplane of $\mathcal{H}$ that intersects this half-space delineates only two $\mathcal{H}$-cells within this half-space. So at least $|\mathcal{C}| - 1$ hyperplanes of $\mathcal{H} \setminus \{h\}$ are needed to seperate the $|\mathcal{C}|$ elements of $\mathcal{C}$, therefore $|\mathcal{C}| \leq |\mathcal{H}|$. Finally, $\min_{e\in\mathcal{E}}(d_{\mathcal{E}}(e)) < |\mathcal{H}|$. □

**Conjecture.** *Let $\mathcal{E}$ be a set of subsets of $\mathbb{R}^n$. If $\mathcal{E}$ is complexly linearly separable, then for every set $\mathcal{H}$ of hyperplanes that is solution to the linear separability of $\mathcal{E}$, we have $|\mathcal{H}| \geq |\mathcal{E}|$, and conversely.*

Lemma C.2 is a simple property that arised from the attempt to prove the above conjecture. This leads us to think that the sum of the degree of liaison to $\mathcal{E}$ of every elements in $\mathcal{E}$ is a measure of the difficulty to find a solution to the linear separability of $\mathcal{E}$.

Appendix D

# Details for reproducibility

## D.1 Initialization and regularization of Diss-Layers

For the autoencoders and the classifier of chapter 3, we initialize the $K$ Diss-Layers biases $(b^k)_k$ at zero and the weights $(w^k)_k, (s^k)_k, (v^k)_k$ from (3.1) by adapting the normalized initialization of [48] (glorot_uniform in keras) for our use of Diss-Layers:

$$(w_j^k)_{jk}, (s_j^k)_{jk}, (v_j^k)_{jk} \sim \mathcal{U}[-\frac{\sqrt{6}}{\sqrt{J+K}}, \frac{\sqrt{6}}{\sqrt{J+K}}]$$

with $J$ the size of the input of the Diss-Layers. In our experiments, NNs use the sum of the $K$ Diss-Layers, hence the normalization constant. In other contexts, different initializations might be needed.

Let us explain the reasoning behind (3.1). Let us reformulate it so that no confusion is possible between indices: $y_j = g[((\sum_{o\neq j} w_o.h_o).s_j + v_j).h_j + b] = f(\boldsymbol{h}, P(\boldsymbol{h}))_j$, with $P(\boldsymbol{h}) = \sum_j w_j.h_j$. We note $h$ instead of $x$, unlike in (3.1), to keep in mind that the input might be the output of a previous layer as it will be the case in section **??** and we use bold font for multidimensional vectors to remove doubt when needed. Let us note that the input of the activation function $g$ is a linear function of $h_j$ of which the slope is determined by the inputs $(h_o)_{o\neq j}$ and some parameters. The parameter $w_o$ determines how much the input $h_o$ contributes to the outputs $(y_j)_{j\neq o}$ of the Diss-Layer, the parameter $s_j$ determines how much $y_j$ is influenced by the inputs $(h_o)_{o\neq j}$, and $w_o \times s_j$ determines how much $y_j$ is influenced by $h_o$, all proportionally to $h_j$. This structure makes possible for a Diss-Layer to extract, or put in another way: to filter, some features of interest. Hence the element-wise sum of several Diss-Layers which take inputs of the same size is related to the filters in convolutional layers. Indeed a filter output is the element-wise sum of its kernels cross-correlations outputs, given as the input of an activation function to obtain

what is called feature maps [95]. However, a Diss-Layer ignores the topology of the input, so it is not restricted to some data types. Furthermore, the pre-potentials $P(h)$ can be used to ensure that the Diss-Layers capture different patterns.

The Diss-Layer itself is quite simple, yet one needs to add some activity regularization in the loss function in order to achieve proper convergence. In any case, a flaw that needs to be averted is that a Diss-Layer may not learn anything. For example, in an autoencoder with two Diss-Layers, which have the identity as activation function, a possible solution for the equation $\boldsymbol{y^{(1)}} + \boldsymbol{y^{(2)}} = \boldsymbol{h}$, with $\boldsymbol{y^{(k)}}$ the Diss-Layers outputs, is to have arbitrary $\boldsymbol{w^{(k)}}$, $\boldsymbol{s^{(k)}} = 0_{\mathbb{R}^2}$, $b^{(k)} = 0$, for $k = 1, 2$ and $\boldsymbol{v^{(1)}} + \boldsymbol{v^{(2)}} = 1_{\mathbb{R}^2}$. A second issue, that appears when several Diss-Layers have the same input $\boldsymbol{h}$, is that without any regularization most of the information can go through some Diss-Layers while other Diss-Layers would learn few patterns and would be most of the time incidental to the output of the NN.

To simplify the notation, we will note the outputs of the Diss-Layers as follows: $l_k(h) = f_k(h, P_k(h))$ and the sum indices are noted by letters that range from 1 to their capital counterparts: $I$ represents the size of the mini-batch ($I \geq 2$ and $\boldsymbol{h_i}$ is the $i$-th input from the mini-batch given to the Diss-Layers $l_k$), $J$ the dimension of the Diss-Layers outputs (and inputs) and $K$ the number of Diss-Layers within the same layer.

$$S = \frac{1}{J.K} \sum_{j,k} \frac{\frac{1}{I.K} \sum_{i,k'} l_{k'}(\boldsymbol{h_i})_j}{0.1 + \frac{1}{I} \sum_i (l_k(\boldsymbol{h_i})_j - \frac{1}{I} \sum_{i'} l_k(\boldsymbol{h_{i'}})_j)^2} \tag{D.1}$$

The quantity $S$ is related to the mean of the inverse of element-wise relative standard deviations of the Diss-Layers outputs on the mini-batch $(\boldsymbol{h_i})_i$:

$$\frac{1}{J.K} \sum_{j,k} \frac{\frac{1}{I} \sum_i l_k(\boldsymbol{h_i})_j}{\sqrt{\frac{1}{I} \sum_i (l_k(\boldsymbol{h_i})_j - \frac{1}{I} \sum_{i'} l_k(\boldsymbol{h_{i'}})_j)^2}}$$

We consider $S$ instead because it is a smoother function with respect to $h$ than its counterpart.

$$M = \sum_k (\frac{1}{I.J} \sum_{i,j} l_k(\boldsymbol{h_i})_j - \frac{1}{I.J.K} \sum_{i,j,k'} l_{k'}(\boldsymbol{h_i})_j)^2 \tag{D.2}$$

The quantity $A = S + M$ must be small to mitigate the first issue of Diss-Layers that may not learn anything because of trivial solutions to the problem enforced by the main objective. $S$ (D.1) forces every output of each Diss-Layer to be different for different inputs, that is the images of the Diss-Layers span sufficiently large subspaces of their codomain,

while $M$ (D.2) prevents the Diss-Layers outputs to have very different amplitudes.

To overcome the second issue, i.e. some Diss-Layers may capture no information from the input, we use a trick that consists in handling the softmax of the Diss-Layers outputs as meaningful probabilities. Let us note $\sigma : z \to (\exp(z_i)/\sum_j \exp(z_j))_i$, the softmax function. We first randomly choose two different Diss-Layers $k_1$ and $k_2$ and compute the quantities $p_i^{(k)} = (\min(1, \max(10^{-7}, \sigma(l_k(\boldsymbol{h_i}))_j)))_{1 \leq j \leq J}$, then:

$$B = \frac{1}{I} \sum_i (D_{KL}(p_i^{(k_1)} \mid\mid p_i^{(k_2)}) + D_{KL}(p_i^{k_2} \mid\mid p_i^{k_1})) \tag{D.3}$$

The function $D_{KL}$, applied on the clipped softmax outputs $(p_{ij}^{(k)})_j$ is the Kullback–Leibler divergence: $D_{KL}(p \mid\mid q) = \sum_j p_j \log(\frac{p_j}{q_j})$. These objectives are conflicting, so to better compensate themselves, they are merged into $C = \frac{A}{1+B} + \frac{B}{1+A}$.

Finally, we want the Diss-Layers to have different pre-potentials $P_k$ so that they capture different patterns. This is done thanks to:

$$\Lambda = \frac{1}{I} \sum_i \frac{1}{0.01 + \frac{1}{K} \sum_k (P_k(\boldsymbol{h_i}) - \frac{1}{K} \sum_{k'} P_{k'}(\boldsymbol{h_i}))^2} \tag{D.4}$$

It is useless to have different pre-potentials as long as the Diss-Layers have not learned anything yet, hence we want the quantity $\Lambda$ (D.4) to influence the training when the first objective is near to be achieved. So the **final quantity** to minimize along with the main loss is $R = C + \frac{\Lambda}{1+C}$. The loss is then : $\mathcal{L} + \lambda.R$, with $\mathcal{L}$ the main loss on the mini-batch $(\boldsymbol{h_i})_i$. In our experiments on antoencoders, we have $\lambda = 0.001$ or $0.01$ (depending on the dimension of the input). The loss weight of the main objective is always equal to 1 and $\mathcal{L}$ is the average Mean Squared Logarithmic Error (MSLE) on the mini-batch. In practice, $\lambda$ is found by having the total loss converging towards a small enough limit. This condition was sufficient for the Hyper-Neuron autoencoder to be able to unsupervisedly separate two separable clusters.

May we remind the reader that the regularization $R$ is needed only when the Diss-Layers have the same input $h$, otherwise only the quantity $C$ has to be minimized and only $A$ if there is only one Diss-Layer. To recap, thanks to $A$, **some** Diss-Layers capture useful information from $h$, thanks to $C$, **each** Diss-Layer captures useful information from $h$, thanks to $R$, the Diss-Layers capture **different** patterns in $h$.

The constants in all the denominators in these equations can be changed but in our

experience, those have always enable a proper convergence of the NN. Finally, we didn't notice any difference in terms of whether we enforce the contribution in the gradient of the quantities in the denominators of $\frac{A}{1+B} + \frac{B}{1+A}$ and $C + \frac{\Lambda}{1+C}$, so to simplify the back-propagation one can consider them constant with respect to other variables. This can be done with stop_gradient in keras [19]: $C + \frac{\Lambda}{1+\text{stop\_gradient}(C)}$, for instance. In our experiment concerning autoencoders, $I = 16$, the number of epochs is 100 and the activation function of each Diss-Layer is the sigmoid. Note that the regularization $R$ depends on the mini-batch $(x_i)_i$. Finally, we use the Adam optimizer [82] of keras with the default parameters for the gradient descent.

## D.2 Datasets and Neural Networks for the experiment on confidence measures

**Datasets**. For the training phase of the NNs, we use the MNIST [94] training set. As for the testing phase, we use three datasets. The first dataset (Figure 3.7, left) is an augmented dataset from the test set of the MNIST database. We created images that are slightly rotated, shifted, «zoomed» or «dezoomed» from the test set of MNIST so that we can use a different sample for temperature scaling and better estimate the in-distribution ECE. The second dataset used for tests is the SVHN [117] train set (Figure 3.8, left) «dezoomed» to fit in the MNIST format (28,28) and transformed into black and white images. The third dataset (cf. Figure 3.9, first row, left column) is the Semeion handwritten digit Dataset [14] whose images are padded with zero values to fit in the MNIST format and then augmented with the same methods than previously. Finally for the analysis, we use transformed images from the augmented Semeion dataset and the MNIST augmented test set. In the second row of Figure 3.9, images are transformed with a gaussian filter of standard deviation equal to 0.7. In the third row, the images of the latter dataset are transformed with the element-wise max function: $\max(u, \text{image})$ where $u$ is a random image with half of the pixels randomly chosen to have value zero while other pixels are drawn from $\mathcal{U}(\left[0, \frac{1}{4}\right])$. In the other rows of Figure 3.9, images from the dataset of the first row are transformed with element-wise linear functions. Likewise, the transformations of the images from the MNIST augmented test set is just linear transformations $f_1 : x \to 0.3 + 0.7 \times x$ and $f_2 : x \to 0.1 + 0.9 \times x$.

**Neural Networks**. The experiment consists in training a lite NN image classifier with

14154 learning parameters (in 5 epochs, batch size equals 64), a big NN image classifier with 59933 learning parameters (in 10 epochs, batch size equals 200) and a multipath NN image classifier with 221460 learning parameters (in 5 epochs, batch size equals 64) on the training set of the MNIST database [94]. We test the confidence of the classical NNs, that is the maximum value of their softmax outputs, and our confidence (3.2) against different digit datasets with the performance measure ECE. The big NN is calibrated through temperature scaling on the 10000 sample points of the MNIST test set. Since the first network is lite, it is well calibrated on in-distribution samples [62]. Our multipath NN has 10 Diss-Layers in its penultimate layer as in Figure 3.5.

Herein, sampling layers are denoted by $Up$ for upsampling layers and $Down$ for downsampling layers: $Up(n)$ repeats data $n$ times in each axis and $Down(n, m)$ operates a max-pooling with pool size $n$ and strides $m$ in each axis (e.g. $Up(2)([[1, 2]]) = [[1, 1, 2, 2], [1, 1, 2, 2]]$) and $Down(2, 3)([1, 1, 2, 2, 3, 3]) = [1, 3]$). $Down$ can potentially be applied after some zero padding so that the output size along a certain axis is the ceiling value of the its input size divided by $m$. After each layer we indicate its output shape, for example $InputLayer(28, 28, 1)$ means that the inputlayer has height and width of size 28 and depth of size 1. $Conv$ and $Dense$ stand for convolutional layer and FC layer respectively. $Flat$ is just a layer, without parameter, to change the shape of the sample point to be able to feed a FCs layer after a convolutional layer.Finally $Droput(p)$ is a dropout layer with probability $p$.

We use the NLL loss for the classical NNs. Our multipath NN architecture is:

$$
\begin{aligned}
&InputLayer(28, 28, 1) \to Conv(28, 28, 4) \\
&\to Down(2, 2)(14, 14, 4) \to Conv(14, 14, 8) \\
&\to Up(2)(28, 28, 8) \to Down(2, 3)(10, 10, 8) \\
&\to Conv(10, 10, 16) \to Up(2)(20, 20, 16) \\
&\to Down(2, 3)(7, 7, 16) \to Conv(7, 7, 32) \\
&\to Up(2)(14, 14, 32) \to Down(2, 3)(5, 5, 32) \\
&\to Flat(800) \to Dense(256) \\
&\to 10DissLayers(256) \to Dense(10)
\end{aligned}
$$

The lite classifier CNN architecture is as follows:

$$\begin{aligned}
&InputLayer(28, 28, 1) \rightarrow Conv(28, 28, 4) \\
&\rightarrow Down(2, 2)(14, 14, 4) \rightarrow Conv(14, 14, 8) \\
&\rightarrow Up(2)(28, 28, 8) \rightarrow Down(2, 3)(10, 10, 8) \\
&\rightarrow Conv(10, 10, 16) \rightarrow Up(2)(20, 20, 16) \\
&\rightarrow Down(2, 3)(7, 7, 16) \rightarrow Conv(7, 7, 32) \\
&\rightarrow Up(2)(14, 14, 32) \rightarrow Down(2, 3)(5, 5, 32) \\
&\rightarrow Flat(800) \rightarrow Dense(10)
\end{aligned}$$

The big classifier CNN architecture is as follows:

$$\begin{aligned}
&InputLayer(28, 28, 1) \rightarrow Conv(24, 24, 30) \\
&\rightarrow Down(2, 2)(12, 12, 30) \rightarrow Conv(10, 10, 15) \\
&\rightarrow Down(2, 2)(5, 5, 15) \rightarrow Dropout(0.2)(5, 5, 15) \\
&\rightarrow Flat(375) \rightarrow Dense(128) \\
&\rightarrow Dense(50) \rightarrow Dense(10)
\end{aligned}$$

Appendix E

# Analysis and future research on the multipah NN

As detailed in Section 3.2.3, our multipath NN classifier is able to extract and assess useful patterns from out-of-distribution examples to have a reasonable ECE. Indeed, even if our measure is a bit less calibrated and accurate on in-distribution data (Figure 3.7) than traditional NNs, it keeps capturing the reality on some out-of-distribution data for which the classical measures of both the lite and the big calibrated NNs are much less representative of the reality (Figure 3.8). However, our measure is not guaranted to be calibrated on every dataset since, like the classical measures, it fails with the Semeion dataset. What is the fundamental difference between SVHN and the Semeion Data Set? Why our measure is calibrated on an out-of-distribution dataset that seems much more challenging than a dataset on which our measure has a high ECE? What method could be useful to overcome this drawback? These are the questions we try to answer in this section.

## E.1 Analysis of the multipath NN confidence measure

Figure 3.9 reveals an interesting property of multipath NNs that gives a glimpse of what could be futur DL methods robust against adversarial attacks. It shows that the confidence measure of the multipath NN is more robust to transformations of a normal input when the changes mainly occurs in the non-discriminative features (the image background in our case). Indeed, the three confidence measures are affected by abnormal values of the discriminative features (the first and the three last rows in Figure 3.9) compared to the training set. Discriminitive features refer to input features that are responsible for

its membership to a class – two images from the same class can have totally different discriminative features, because of translation for instance. On the contrary, for the second, the third and the fourth rows of Figure 3.9, discriminative features are similar those of MNIST images (while non-discriminative features are quite different for the third and fourth rows) and our measure, unlike the classical measures, produces a small calibration error in these three cases. It is coherent with the goal of Diss-Layers and the results on SVHN images. Indeed, Diss-Layers are meant to filter features of interest (see sections 3.2.2 and D.1). Moreover, changing the non-discriminitive features comes down, to some extent, to move the data point along the boundary determined by the Diss-Layers of the Hyper-Neurons responsible for its class within the hidden space of the current layer, so the element-wise sum of these Diss-Layers will barely change and so will the confidence measure. For further results, we tested the transformations $f_1 : x \to 0.3 + 0.7 \times x$ and $f_2 : x \to 0.1 + 0.9 \times x$ on the MNIST augmented test set, and the ECE increased by a factor of 14 and 2.13 (respectively for $f_1$ and $f_2$) for the confidence measure of the traditional NN, of 8.14 and 2.23 (respectively for $f_1$ and $f_2$) for the confidence measure of the traditional big NN calibrated through temperature scaling, and only by a factor of 8 and 0.39 (hence decreased) (respectively for $f_1$ and $f_2$) for our confidence measure while the accuracy stayed nearly the same (except for the big NN whose accuracy dropped to 95.546% for $f_1$). This supports the claim that our confidence measure is more robust to transformations of a normal input when the changes mainly occurs in the non-discriminative features. In contrast, the confidence measure of a traditional NNs does not have this topological property. This property explains why our measure is still calibrated on the SVHN dataset where patterns of digits are quite similar to those in the MNIST dataset but coexists with patterns not useful for the prediction that affect the calibration of the confidence measure of traditional NNs. Unlike the traditional NN, the multipath NN will most likely benefit from a tool from TDA.

In brief, temperature scaling on a big NN is the method the more robust to shifts in the dataset as shown in Figure 3.9, and when the dataset is not shifted but utterly changed, our measure, thanks to the redundancy made possible by Diss-Layers that extract meaningful patterns, is the more robust provided that the discriminative features are similar to those of the training set (Figure 3.8). Based on this property, we propose in the next section a line of research to arrive at a model that is calibrated for any input.

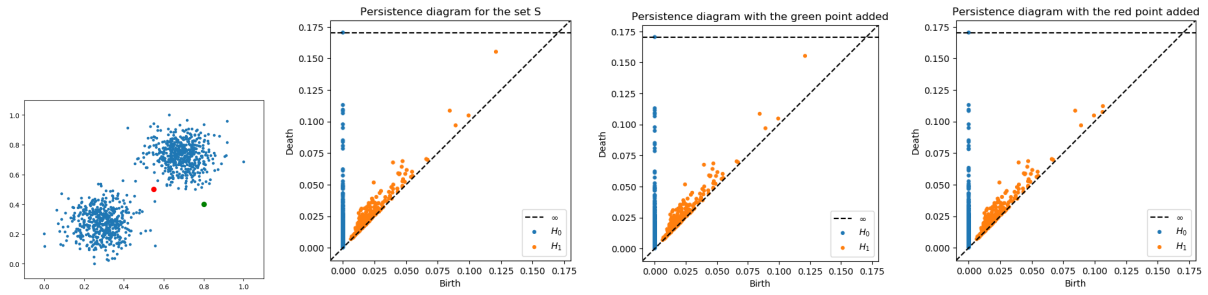



Figure E.1 – Left: Two clusters and two new points. From left to right: Persistence diagrams for 0-d and 1-d holes regarding a set S of two clusters, regarding S $\bigcup\{(0.55, 0.5)\}$ (red point) and regarding S $\bigcup\{(0.8, 0.4)\}$ (green point).

## E.2 Future Research

As explained above, it turns out that the confidence in the prediction of NNs is highly dependent on the location of the input in its ambient space. Hence, topological considerations are necessary to enable a general solution to the issue of confidence in the prediction for an input without hypotheses on its underlying distribution. We think that the limitations of our method can be overcome thanks to the work presented in [12] where a tool from TDA, the persistent diagram, is adapted to incorporate some topological properties in the data representation of a model. Indeed, they proposed a loss function of a persistent diagram – which is a representation in $\mathcal{P}(\mathbb{R}^2)$ of a filtration, describing some topological properties of a set. Among other things, they were able to regularize an image, through a filtration related to its set of pixels to improve its visual quality, and to regularize the weights of a model – through a filtration related to its set of weights. For the image regularization, they did it with the superlevel set filtration of the image – a superlevel set is all the pixels whose value is greater than some threshold – in order to promote a single pixel with the global maximum value and mitigate the noise in the image. For the weight regularization, they proposed to use another filtration, the Rips filtration, which is based on distance between points of a set, the weights in their case. There is nothing preventing us from using the Rips filtration to assess the topological properties of the hidden layer space of a NN with respect to a certain dataset. In our context, the first set on which the Rips filtration would be computed is a subset – for computational reasons – $D$ of the training set encoded by a subnetwork [1] $SN$ of the NN, composed of its first layers, and the

1. One could do it directly on the training set, but it is more appropriate to use the first layers of the network that has learned to represent the clusters in a linearly separable way and has prior knowledge on the data type, as some translation invariance in the case of images thanks to pooling layers.

second would be $D$ to which a new input encoded by $SN$ is added. The idea is to transform this new input thanks an optimization problem on the persistence diagrams of these filtrations so that the transformed input matches the topological properties of $D$. From now on, through a misuse of langage, we will miss the part of a subset of the training set encoded by the first layers of the NN and refers to persistence diagrams as regarding the training set and the training set with a new input. The overall idea is similar to the one of the post-hoc method called DkNN [123] that assesses the conformity of a test input to the training set thanks to another set, called the calibration set. Other considerations than statistical ones are indeed necessary to guide the final decision, however the Rips filtration enables considering patterns of much higher order than the k-nearest neighbors method: it allows to capture the persistence of a k-dimensional holes (0-d holes: gap between two components, 1-d holes: holes, 2-holes: cavities, etc..) of the Vietoris–Rips complex from the data points when the scale parameter determining this simplicial complex varies. Find an introduction to TDA in [17]. This persistence is represented in a diagram where the x-axis corresponds to the scale parameter values that gives birth to k-dimensional holes and the y-axis corresponds to the scale parameter values that kills them. Moreover, the persistence diagram is stable with respect to pertubations of the data, so this method will not capture noise of the training set. However, the persistence diagram is not stable with respect to outliers. Hence with this filtration, one could regularize the input so it is more similar to the training set in terms of topological properties rather than just distances.

Our expectation is that the input transformed by the regularization with some cost function, for instance the Wasserstein distance between two diagrams in [12] from the Rips filtrations of the training set and from the one with a new input, will produce a calibrated confidence in the prediction of the multipath NN. Indeed, let us recall that the confidence measure of the multipath NN is robust to some extent to changes in the non-discriminative features. Yet, these changes are the less disruptive for the persitent diagrams than changes in discriminative features. In effect, in Figure E.1, we can see how an ambiguous point (in red) can interfere with the persistence diagram of a set, while the point that results from a change in a non discriminative direction (in green) is less disruptive. Yet adversarial attacks aims at slightly changing a data point to draw it near the boundary, that is to say between two clusters: where the persistence diagram regarding the training set with the new input is susceptible to be different from the persistence diagram regarding the training set alone. So if the defense mechanism is to minimize the difference between the two aforesaid persistence diagrams by transforming the test sample point, it will either

be closer to one of the two nearby classes or moved in a non-discriminative direction. Since our confidence is robust against this latter kind of change, both options will tend to calibrate it. Therefore, one can expect that the prediction will change or the confidence will decrease in the face of outliers. In that respect, the transformation of the input to be tested thanks to optimization problem on persistence diagrams with tools of which [12] have proven the usefulness in the context of DL, jointly with our multipath NN, seems to us to be a good way to detect adversarial examples, launching a promising research direction in DL. However, one can legitimately object that this line of research will be confronted with the difficult convergence of the NNs using the Wasserstein distance.

Modifying the input as a defense mechanism has been proposed in [97] where controlled perturbations is added to an image to separate in- and out-of-distribution examples thanks to the softmax prediction of a pre-trained NN. However our goal is more general than outlier detection, we wish to provide a calibrated confidence measure, even for outliers. Only [12] opt for incorporating topological considerations in the NNs architecture, yet with different goals than ours. They focus on generalization, i.e. having a high test accuracy, while our objective is to have a calibrated confidence for any input, which we propose to refer to as meta-generalization. Further work is needed to know whether it is possible to increase the test accuracy of a multipath NN up to the accuracy of classical big NNs without increasing its test ECE and if not, whether the property of meta-generalization holds on large multipath NNs (re)-calibrated through temperature scaling [62] or thanks to the Variance-Weighted Confidence-Integrated Loss [137].

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

**Titre :** Traçabilité et intégrité de l'information au sein de systèmes critiques – Analyse et proposition de méthodes statistiques

**Mot clés :** Securité, Apprentissage automatique, Système de Contrôle Industriel

**Résumé :** Les systèmes industriels sont voués à fonctionner des années durant et leurs dispositifs font parfois face à des contraintes énergétiques empêchant la mise en place de nouveaux moyens de sécurité. Nous étudions donc des solutions passives, c'est-à-dire n'ayant besoin que des données, au problème de surveillance des processus physiques de systèmes industriels par l'observation des valeurs des capteurs, des actionneurs et des commandes des automates. La majeure partie de nos travaux concerne l'intégrité de ces données qui se traduit par le fait que les données liées à un ensemble d'actions du système n'ont pas subies un changement inattendue et la traçabilité de l'information que nous définissons comme la capacité d'authentifier chaque processus de transformation des données depuis leur création par le système industriel jusqu'à leur dernière utilisation. Nous proposons un nouveau concept d'état de Système Cyber-Physique que les modèles d'apprentissage automatique peuvent utiliser pour répondre aux questions de l'intégrité et de la traçabilité des données et nous l'appliquons plus particulièrement à l'autoencoder. Nous proposons un nouveau type de réseau de neurones classifieur accompagné d'une mesure de confiance qui nous permet de répondre à notre problème de traçabilité.

**Title:** Traceability and integrity of information within critical systems – Study and proposal of statistical methods

**Keywords:** Security, Machine learning, Industrial Control System

**Abstract:** Industrial systems often work for years and their devices are sometimes energetically constrained so that new security measures are not practicable. Passive solutions, i.e. with only data at hand, to the problem of system physical processes security through sensors, actuators and automata values monitoring are studied. The major part of our work concerns data integrity, that is the fact that data linked to a set of actions of the system are not unexpectedly modified, and information traceability which we define as the capability to authenticate each process of data transformation from their creation by the industrial system to their end use. We propose a new concept of Cyber-Physical System state that machine learning models can use to handle the issue of data integrity and we use it with the autoencoder. We propose a new class of Deep Learning classifiers with a measure of confidence in the prediction which we use for information traceability.