



**HAL**  
open science

# Prediction of interventions and optimization of resources based on machine learning and operations research for fire departments

Selene Leya Cerna Ñahuis

## ► To cite this version:

Selene Leya Cerna Ñahuis. Prediction of interventions and optimization of resources based on machine learning and operations research for fire departments. Artificial Intelligence [cs.AI]. Université Bourgogne Franche-Comté, 2022. English. NNT : 2022UBFCD048 . tel-04368508

**HAL Id: tel-04368508**

**<https://theses.hal.science/tel-04368508>**

Submitted on 1 Jan 2024

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

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

**THÈSE DE DOCTORAT**

**DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ**

**PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ**

École doctorale n°37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

**SELENE LEYA CERNA ÑAHUIS**

**Prediction of Interventions and Optimization of Resources Based on  
Machine Learning and Operations Research for Fire Departments**

**Prédiction des Interventions et Optimisation des Ressources Basées sur L'Apprentissage  
Automatique et la Recherche Opérationnelle pour les Services d'Incendie et Secours**

Thèse présentée et soutenue à Belfort, le 12 Décembre 2022

Composition du Jury :

ASSOCIATE PROF DEKA LIPIKA	De Montfort University	Rapporteure
PROF DEMERJIAN JACQUES	Université Libanaise	Rapporteur
ASSOCIATE PROF HABIB CAROL	Grande École d'Ingénieurs JUNIA	Examinatrice
PROF COUTURIER RAPHAËL	Université Bourgogne Franche-Comté	Examinateur
PROF GUYEUX CHRISTOPHE	Université Bourgogne Franche-Comté	Directeur de thèse
CDT ROYER-FEY GUILLAUME	Service Départemental d'Incendie et de Secours du Doubs	Codirecteur de thèse



# ABSTRACT

## Prediction of Interventions and Optimization of Resources Based on Machine Learning and Operations Research for Fire Departments

Selene Leya Cerna Ñahuis  
University Bourgogne Franche Comté, 2022

Supervisors: Christophe Guyeux, Guillaume Royer-Fey

Around the world, fire departments constantly seek to develop strategies to reduce their response time in interventions, since it is one of the most important factors when saving lives and to measure the quality of their service. The data collected over the years on their interventions during fires, rescues, traffic accidents, etc., could be used to develop data-driven systems for decision making, understand the trends of certain events, improve the efficiency of their service, and reduce operating costs.

For this reason, the main objectives of this thesis are: i) Predict interventions to support decision-making in the short-term deployment of resources, and ii) Develop methodologies to optimize resources in the long-term.

For objective *i*, we will build Machine Learning (ML) based models to predict the number of interventions in the next hours, when a peak in operational load will occur due to rare events (e.g., storms and floods), the mortality of victims, the response time to an incident, and the turnaround time of ambulances in hospitals.

For objective *ii*, we will develop optimization methods based on ML and Operations Research (OR) techniques for the creation of a quality of service indicator, an operational load simulator, the redeployment of ambulances, and the implementation of a new center.

**KEYWORDS:** Fire Brigade, Machine Learning, Operations Research, Neural Networks, Bayesian Optimization, Artificial Intelligence.



# RÉSUMÉ

## Prédiction des Interventions et Optimisation des Ressources Basées sur L'Apprentissage Automatique et la Recherche Opérationnelle pour les SIS

Selene Leya Cerna Ñahuis  
Université Bourgogne Franche Comté, 2022

Encadrants: Christophe Guyeux, Guillaume Royer-Fey

Dans le monde entier, les Services d'Incendie et Secours (SIS) cherchent à développer des stratégies pour réduire leur temps de réponse lors des interventions, car c'est l'un des facteurs les plus importants pour sauver des vies et pour mesurer la qualité de leur service. Les données collectés au fil des ans sur leurs interventions lors d'incendies, de sauvetages, d'accidents de la route, etc., pourraient être utilisés pour développer des systèmes de prise de décision basés sur les données, comprendre les tendances de certains événements, améliorer l'efficacité de leur service et réduire les coûts d'exploitation.

Pour cette raison, les principaux objectifs de cette thèse sont: i) Prédire les interventions pour soutenir la prise de décision dans le déploiement des ressources à court terme, et ii) Développer des méthodologies pour optimiser les ressources à long terme.

Pour l'objectif *i*, nous construisons des modèles basés sur l'apprentissage automatique (ML) pour prédire le nombre d'interventions dans les prochaines heures, quand un pic de charge opérationnelle se produira en raison d'événements rares (par exemple, tempêtes et inondations), la mortalité des victimes, le temps de réponse à un incident et le temps de rotation des ambulances dans les hôpitaux.

Pour l'objectif *ii*, nous développerons des méthodes d'optimisation basées sur des techniques de ML et de recherche opérationnelle (OR) pour la création d'un indicateur de qualité de service, un simulateur de charge opérationnelle, le redéploiement des ambulances, et la mise en place d'un nouveau centre.

**Mots clés:** Sapeur Pompier, Apprentissage Automatique, Recherche Opérationnelle, Réseaux Neuronaux, Optimisation Bayésienne, Intelligence Artificielle.



# ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisors, Prof. Christophe Guyeux and Cdt. Guillaume Royer-Fey of the Service Départemental d'Incendie et de Secours du Doubs (SDIS 25), for giving me the opportunity to work with and learn from them; for giving me the honor of working with such a remarkable organization and which I admire for its noble effort and dedication to civil protection; for guiding and motivating me throughout my research with all their professional and life advice. Also, many thanks to Cne. Céline Chevallier and Cap. Guillaume Plumerel, whom patiently and kindly shared their knowledge about resource planning and distribution.

I am enormously grateful to Héber, my life partner, who was by my side through good and bad times, through successes and errors, who with his jokes, patience, and understanding, motivates my curiosity and advises my reckless logic, with whom I enjoy having long research discussions or philosophizing about life, between laughter and lot of imagination. Everything is more fun and joyful with you.

I want to infinitely thank my parents, Eulalia and Fernando, who never measured their efforts to educate their children and always encouraged us to persevere and fight for what we want. To my brother Edwin, for all his efforts working and helping the family since he was a child. To my sister Liesel, for taking care of me and giving me a profession. And to my brother Fernando, for sharing his knowledge with me since I was little and showing me the scientific path.

I am grateful to professors Raphaël Couturier and David Laymani, for their wise advice, shared work, and guidance in my research. And many thanks to my lab partners Zhì Háo, Ralph, and Hicham for the tips and fun times shared.

Finally, this work was funded with the support of the French Ministry of Higher Education and Research, managed by the National Association of Research and Technology (ANRT), for the CIFRE thesis (N 2019/0372), and by the EIPHI-BFC Graduate School (contract ANR-17-EURE-0002).



# CONTENTS

<b>I</b>	<b>Thesis Introduction</b>	<b>5</b>
<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Introduction . . . . .	7
1.2	Motivation and Objectives . . . . .	9
1.3	Contributions of the Thesis . . . . .	10
1.4	Thesis Outline . . . . .	12
<b>II</b>	<b>Background</b>	<b>13</b>
<b>2</b>	<b>Machine Learning and Operations Research</b>	<b>15</b>
2.1	Machine Learning . . . . .	15
2.1.1	Tasks . . . . .	15
2.1.2	Types of Learning . . . . .	16
2.1.3	Data Preprocessing and Feature Engineering . . . . .	16
2.1.4	Modeling . . . . .	17
2.1.4.1	Linear Methods . . . . .	18
2.1.4.2	Algorithms Based on Decision Trees . . . . .	18
2.1.4.3	Neural Networks for Tabular Data . . . . .	22
2.1.4.4	Neural Networks for Text Processing . . . . .	23
2.2	Operations Research . . . . .	25
2.2.1	Integer Programming . . . . .	26
2.2.1.1	General Integer Problems . . . . .	26
2.2.1.2	Zero-One Problems . . . . .	26
2.2.1.3	Mixed Integer Problems . . . . .	27
2.2.2	Multi-Objective Optimization . . . . .	27

2.2.2.1	Decision and Objective Space . . . . .	28
2.2.2.2	Pareto Dominance . . . . .	29
2.2.2.3	Aggregated Functions . . . . .	29
2.2.2.4	Non-dominated Sorting Genetic Algorithm II (NSGA-II) . . . . .	30
2.2.3	Bayesian Optimization . . . . .	30
2.3	Performance Metrics . . . . .	31
2.3.1	Metrics for Regression Problems . . . . .	31
2.3.2	Metrics for Classification Problems . . . . .	32
2.4	Conclusion . . . . .	34
<b>3</b>	<b>Datasets and Operational Flow</b>	<b>35</b>
3.1	Internal Data . . . . .	35
3.2	External Data . . . . .	37
3.3	Operational Process Flow . . . . .	38
3.4	Conclusion . . . . .	43
<b>III</b>	<b>Contribution: Prediction of Fire Department Interventions</b>	<b>45</b>
<b>4</b>	<b>Predicting the Number of Interventions</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.1.1	Objectives . . . . .	49
4.2	Materials and Methods . . . . .	49
4.2.1	Data Collection . . . . .	49
4.2.2	Data Encoding . . . . .	50
4.2.3	Modeling . . . . .	50
4.3	Results . . . . .	51
4.4	Discussion and Related Work . . . . .	55
4.5	Conclusion . . . . .	56
<b>5</b>	<b>Forecasting Ambulances' Turnaround Time in Hospitals</b>	<b>57</b>
5.1	Introduction . . . . .	57

5.1.1	Description of the Operational Process . . . . .	59
5.1.2	Objectives . . . . .	60
5.2	Materials and Methods . . . . .	61
5.2.1	Data Collection . . . . .	61
5.2.2	Data Analysis . . . . .	62
5.2.3	Service Breakdown Analysis . . . . .	66
5.2.4	Proposed Methodology . . . . .	68
5.2.4.1	Overview . . . . .	68
5.2.4.2	AvTT Model: Preprocessing and Modeling . . . . .	70
5.2.4.3	TT Model: Preprocessing and Modeling . . . . .	71
5.2.5	Baseline Models . . . . .	72
5.2.5.1	Baseline: AvTT per Hour . . . . .	72
5.2.5.2	Baseline: TT for an Ambulance . . . . .	72
5.3	Results . . . . .	72
5.3.1	Forecasting the Average Turnaround Time of Ambulances per Hour . . . . .	73
5.3.2	Forecasting the Turnaround Time for each Ambulance . . . . .	76
5.4	Discussion and Related Work . . . . .	78
5.5	Conclusion . . . . .	81
<b>6</b>	<b>Predicting Ambulance Response Time</b>	<b>83</b>
6.1	Introduction . . . . .	83
6.1.1	Description of the Operational Process . . . . .	84
6.1.2	Objectives . . . . .	85
6.2	Materials and Methods . . . . .	85
6.2.1	Data Collection . . . . .	86
6.2.2	Data Analysis . . . . .	87
6.2.3	Preprocessing and Modeling . . . . .	88
6.3	Results . . . . .	89
6.4	Discussion and Related Work . . . . .	91
6.5	Conclusion . . . . .	92

<b>7</b>	<b>Predicting Victim’s Mortality and Their Need for Transportation</b>	<b>95</b>
7.1	Introduction . . . . .	95
7.1.1	Description of the Operational Process . . . . .	96
7.1.2	Objectives . . . . .	96
7.2	Materials and Methods . . . . .	97
7.2.1	Data Collection . . . . .	97
7.2.2	Data Analysis . . . . .	98
7.2.3	Methodology . . . . .	99
7.3	Results . . . . .	103
7.4	Discussion and Related Work . . . . .	105
7.5	Conclusion . . . . .	106
<b>8</b>	<b>Predicting Intervention Peaks Due to Rare Events</b>	<b>109</b>
8.1	Introduction . . . . .	109
8.1.1	Objectives . . . . .	110
8.2	Materials and Methods . . . . .	111
8.2.1	Data Acquisition . . . . .	112
8.2.2	Data Preprocessing . . . . .	113
8.2.3	Modeling . . . . .	115
8.3	Results . . . . .	117
8.3.1	Prediction of Interventions Using Basic Univariate Time Series Models . . . . .	117
8.3.2	Prediction of Interventions Using Multivariate Time Series Models . . . . .	119
8.3.3	Prediction of Intervention Peaks Using Multilabel Classification Models Based on Decision Trees and Tabular Data . . . . .	122
8.3.4	Prediction of Intervention Peaks Using Multilabel Classification Models Based on NLP Techniques and Text From Meteorological Bulletins . . . . .	124
8.4	Discussion and Related Work . . . . .	126
8.5	Conclusion . . . . .	128

<b>IV Contribution: Optimization of Fire Department Resources</b>	<b>129</b>
<b>9 Optimizing the breakage calculation</b>	<b>131</b>
9.1 Introduction . . . . .	131
9.2 Optimized Search for Available Adapted Armament . . . . .	132
9.3 Predicting the Number of Breakdowns per Day . . . . .	134
9.3.1 Data Preprocessing . . . . .	134
9.3.2 Modelling . . . . .	135
9.4 Results and Discussion . . . . .	136
9.4.1 Results of the breakage calculation . . . . .	136
9.4.2 Results of the breakdown predictions . . . . .	139
9.4.3 Discussion and related work . . . . .	141
9.5 Conclusion . . . . .	142
<b>10 Development of the Operational Load Simulator</b>	<b>143</b>
10.1 Introduction . . . . .	143
10.2 Scenario Variables . . . . .	144
10.3 Simulator Overview . . . . .	145
10.4 Departure Time Prediction . . . . .	149
10.5 Arrival Time Prediction . . . . .	150
10.6 Results . . . . .	151
10.6.1 Results of Departure Time Predictions . . . . .	151
10.6.2 Results of Arrival Time Predictions . . . . .	152
10.6.3 Results of Base Models . . . . .	153
10.7 Discussion and related work . . . . .	154
10.8 Conclusion . . . . .	157
<b>11 Optimization of Engine Distribution</b>	<b>159</b>
11.1 Introduction . . . . .	159
11.2 Methodology . . . . .	160
11.3 Results . . . . .	162

11.4 Discussion and related work . . . . .	165
11.5 Conclusion . . . . .	166
<b>12 Optimal Positioning of a New Center and its configuration</b>	<b>167</b>
12.1 Introduction . . . . .	167
12.2 Methodology . . . . .	168
12.3 Results . . . . .	171
12.4 Discussion and Related Work . . . . .	173
12.5 Conclusion . . . . .	174
<b>13 Optimal Sizing of Agents</b>	<b>177</b>
13.1 Introduction . . . . .	177
13.2 Optimal Sizing for New Agents . . . . .	179
13.3 Optimal Sizing for New and Existing Agents . . . . .	186
13.4 Optimal Sizing for Existing Agents . . . . .	189
13.5 Results . . . . .	191
13.5.1 Results for Optimal Sizing of New Agents . . . . .	191
13.5.2 Results for Optimal Sizing of New and Existing Agents . . . . .	192
13.5.3 Results for Optimal Sizing of Existing Agents . . . . .	194
13.6 Discussion and Related Work . . . . .	195
13.7 Conclusion . . . . .	196
<b>V Conclusion &amp; Perspectives</b>	<b>197</b>
<b>14 Conclusion &amp; Perspectives</b>	<b>199</b>
14.1 General Conclusion . . . . .	199
14.2 Perspectives . . . . .	201
<b>15 Publications</b>	<b>203</b>

# LIST OF ABBREVIATIONS

<b>AdaBoost</b>	. . . . .	Adaptive Boosting
<b>ACC</b>	. . . . .	Accuracy
<b>ACCOE</b>	. . . . .	Calculated accuracy with margin of error of zero interventions.
<b>ACC1E</b>	. . . . .	Calculated accuracy with margin of error of $\pm 1$ interventions.
<b>ACC2E</b>	. . . . .	Calculated accuracy with margin of error of $\pm 2$ interventions.
<b>ACC5</b>	. . . . .	Calculated accuracy with margin of error of $\pm 5$ minutes.
<b>ACC10</b>	. . . . .	Calculated accuracy with margin of error of $\pm 10$ minutes.
<b>AOD</b>	. . . . .	Ambulance Offload Delay
<b>APOPT</b>	. . . . .	Advanced Process OPTimizer
<b>Arr-DS</b>	. . . . .	Arrivals Dataset
<b>ART</b>	. . . . .	Ambulance Response Time
<b>AUC</b>	. . . . .	Area Under the Receiver Operating Characteristic Curve
<b>AVP</b>	. . . . .	Accident sur la Voie Publique, Accident on Public Roads
<b>AvTT</b>	. . . . .	Average Turnaround Time per hour
<b>AvTT-DS</b>	. . . . .	Average Turnaround Time Dataset
<b>BayesOpt</b>	. . . . .	Bayesian Optimization
<b>BACC</b>	. . . . .	Balanced Accuracy
<b>BERT</b>	. . . . .	Bidirectional Encoder Representations from Transformers
<b>BPS</b>	. . . . .	Breakdowns of Public Service
<b>BRR</b>	. . . . .	Bayesian Ridge Regression
<b>BSAvTT</b>	. . . . .	Baseline Model to predict the AvTT per hour
<b>BSC</b>	. . . . .	Breakdowns to the Speed Contract
<b>BSTT</b>	. . . . .	Baseline Model to predict the TT of an ambulance
<b>CA</b>	. . . . .	Chef d'Agrès
<b>CCFM</b>	. . . . .	Camion Citerne Feux de forêts Moyen, Medium Forest Fire Tank Truck
<b>CCGC</b>	. . . . .	Camion Citerne Grande Capacité, Large Capacity Tanker Truck

<b>CE</b>	.....	Chef d'Equipe
<b>CHIH</b>	.....	Intercommunal Hospital Center of Haute-Comté
<b>CHRU</b>	.....	Besançon Regional University Hospital Center
<b>CNN</b>	.....	Convolutional Neural Network
<b>COND</b>	.....	Conducteur
<b>COVID-19</b>	.....	Coronavirus Disease 2019
<b>CPI</b>	.....	Centre de Première Intervention, First Intervention Center
<b>CPIR</b>	.....	Centre de Première Intervention Renforcé, Reinforced First Intervention Center
<b>CS</b>	.....	Centre de Secours, Rescue Center
<b>CSP</b>	.....	Centre de Secours Principal, Principal Rescue Center
<b>CSR</b>	.....	Centre de Secours Renforcé, Reinforced Rescue Center
<b>DSM</b>	.....	Double Standard Model
<b>ED</b>	.....	Emergency Department
<b>EMS</b>	.....	Emergency Medical Services
<b>EPA</b>	.....	Échelle Pivotante Automatique, Vehicle with Retractable Ladder
<b>EQUI</b>	.....	Équipier
<b>External-DS</b>	..	External Dataset
<b>FPCA</b>	.....	Chef d'Agrès d'un pompe engine, Engine Chief of VSAV
<b>FPCE</b>	.....	Chef d'Equipe d'un pompe engine, Crewman of VSAV
<b>FPCOND</b>	.....	Conducteur d'un pompe engine, Driver of VSAV
<b>FPEQUI</b>	.....	Équipier d'un pompe engine, Crewman of VSAV
<b>FPR</b>	.....	False Positive Rate
<b>FPT</b>	.....	Fourgon Pompe Tonne, Van Pump Ton
<b>FPTGP</b>	.....	Fourgon Pompe Tonne Grande Puissance, High Power Ton Pump Van
<b>GMM</b>	.....	Gaussian Mixture Models
<b>HF</b>	.....	Health Facility
<b>HNFC</b>	.....	North Franche-Comté Hospital
<b>HRAP</b>	.....	Human Resource Allocation Problem
<b>INC</b>	.....	Incendie, Fires
<b>LASSO</b>	.....	Least Absolute Shrinkage and Selection Operator
<b>LGBM</b>	.....	Light Gradient Boosting Machine

<b>LSCM</b>	Location Set Covering Model
<b>LSTM</b>	Long Short-Term Memory
<b>MAE</b>	Mean Absolute Error
<b>MEXCLP</b>	Maximum Expected Coverage Location Problem
<b>MINLP</b>	Mixed-Integer Nonlinear Programming
<b>MCLP</b>	Maximal Covering Location Problem
<b>ML</b>	Machine Learning
<b>MLP</b>	Multi-layer Perceptron
<b>NC-AU</b>	New center based on an Automated Decision
<b>NC-HU</b>	New Center based on a Human Decision
<b>NLP</b>	Natural Language Processing
<b>NSGA-II</b>	Non-dominated Sorting Genetic Algorithm II
<b>NSGA-II-AFO</b>	NSGA-II Adapted for Firemen Optimization
<b>OD</b>	Opérations Diverses, Diverse Operations
<b>OHE</b>	One-Hot-Encoding
<b>OR</b>	Operations Research
<b>OSRM</b>	Open Source Routing Machine
<b>PDD</b>	Plan De Deploiement, Deployment Plan
<b>RMSE</b>	Root Mean Square Error
<b>RNN</b>	Recurrent Neural Networks
<b>RF</b>	Random Forest
<b>ROC</b>	Receiver Operating Characteristic Curve
<b>RTN</b>	Risques technologiques et naturels, Technological and natural risks
<b>SAMU</b>	Service d'Aide Médicale Urgente, Urgent Medical Aid Service
<b>SAP</b>	Secours à Personnne, Rescue People
<b>SDIS25</b>	Service Départemental d'Incendie et de Secours du Doubs
<b>SVM</b>	Support Vector Machine
<b>TabNET</b>	Attentive Interpretable Tabular Learning
<b>TNR</b>	True Negative Rate
<b>TPR</b>	True Positive Rate
<b>TT</b>	Turnaround Time of an ambulance

<b>VPIL</b>	. . . . .	Vehicule Première Intervention avec permis de vehicule Legère, First Response Vehicle with Light Vehicle Permit
<b>VLSM</b>	. . . . .	Véhicule Léger de Secours Médicalisé, Light Medical Rescue Vehicle
<b>VLU</b>	. . . . .	Véhicule Léger Utilitaire, Light Utility Vehicle
<b>VSAV</b>	. . . . .	Véhicule de Secours et d'Assistance aux Victimes, Rescue and Assistance to Victims Vehicle
<b>VSCA</b>	. . . . .	Chef d'Agrès de VSAV, Engine Chief of VSAV
<b>VSCOND</b>	. . . . .	Conducteur de VSAV, Driver of VSAV
<b>VSEQUI</b>	. . . . .	Chef d'Equipe de VSAV, Crewman of VSAV
<b>VTU</b>	. . . . .	Véhicule Tous Usages, All Purpose Vehicle
<b>XGBoost</b>	. . . . .	Extreme Gradient Boosting
<b>Z1</b>	. . . . .	Urban zone
<b>Z2</b>	. . . . .	Semi-urban zone
<b>Z3</b>	. . . . .	Rural zone



# THESIS INTRODUCTION



# INTRODUCTION

## 1.1/ INTRODUCTION

Emergency medical services (EMS) are the first to provide an immediate response to accident victims and transportation to the most appropriate hospital if needed. The manner in which this medical service is implemented depends on the policies of the country of origin and the risks identified throughout its history. Generally, EMSs are composed of hospitals' own transport services, licensed private paramedics, and fire departments. In France, these last ones, fire departments, are not only responsible for extinguishing fires but also for rescuing people in emergencies and transporting them to a medical facility. As a result, their operational burden has increased sharply in the last decade.

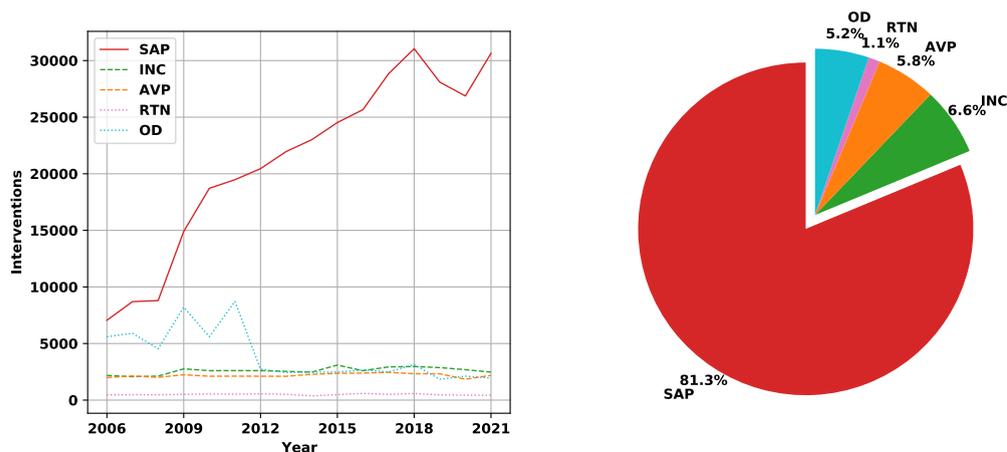


Figure 1.1: The image on the left illustrates by type of operation the increase in the number of interventions over the years, from 2006 to 2021. The image on the right depicts the distribution of the operational load by type and for the last year, 2021. The data analyzed were provided by SDIS25.

This is the case of the Departmental Fire and Rescue of Doubs (SDIS25, in french *Service Départemental d'Incendie et de Secours du Doubs*), in France, where the number of interventions has been increasing over the years. The left side of Fig. 1.1 shows how

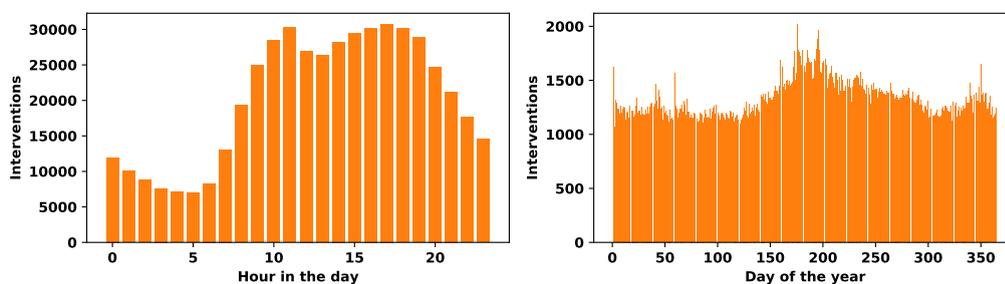


Figure 1.2: The image on the left shows the cumulative number of interventions per hour of the day. The image on the right illustrates the cumulative number of interventions per day of the year. Both images consider data of the last 16 years. February 29th were removed for a better visualization of the pattern.

its demand by type of operation has been increasing from 2006 to 2021, with the exception of 2020, in which there were quarantines due to the COVID-19 pandemic and the demand was reduced. SDIS25 considers 5 types of operations, which are Rescue people (SAP, in french *Secours À Personne*), Fires (INC, in french *INCendie*), Accident on public roads (AVP, in french *Accident sur la Voie Publique*), Technological and natural risks (RTN, in french *Risques Technologiques et Naturels*), and Diverse operations (OD, in french *Opérations Diverses*). The right side of Fig. 1.1 shows the distribution of the operational load and confirms that a large part of the burden is of the SAP type, i.e., more than 80% of the interventions are rescue people.

As in France, fire departments in other countries are also facing a high operational demand and, what is more, a major crisis in emergency medical transport, as their budgets are not sufficient to cover all the resources requested. Some of the causes of this scissors effect are the aging of the population (i.e., more medical care and transport for the elderly), climate change (i.e., more natural disasters and higher demand for operational resources in short periods), epidemics and pandemics (i.e., overflow of hospital care and impact on ambulance waiting times), urban expansion, the indebtedness of countries, economic crises and thus the rationalization of social spending. In addition, a budget cut or constant budget can cause the closure of small centers and the redirection of their demand to larger centers, overloading their resources, and finally generating disruptions in service, since there is no resources to attend an intervention. These disruptions degrade the quality of service and could leave communities unprotected for long periods of time.

The good news is that over the years several fire brigades have been collecting detailed information about their interventions. In addition, it is known that the activity of emergency health transport is strongly related to human activity, for example, the number of interventions carried out is reduced at night and increases during the day, even during the day there are hours with peaks of interventions, there are more pedestrian falls during winters due to ice, there are more drownings in swimming pools during summer, there are more

traffic accidents during vacations, etc. This can be seen in Fig. 1.2, where the cumulative flow of interventions of SDIS25 is presented by hour of the day and by day of the year.

The main goal of fire departments is to do a lot with little, i.e., they seek to optimize the redistribution of their scarce resources to achieve their objectives regarding the reduction of response time and improvement of service quality. It is here, where the recent technologies based on artificial intelligence and data exploitation will alleviate the pressure on fire departments, taking advantage of all the information collected by them for years.

On the one hand, Machine learning is a very active area of artificial intelligence, which aims to detect patterns or a function based on a training dataset, and then predict the outputs of a never-before-seen dataset, i.e., computers have the ability to learn without being explicitly programmed [75, 91, 84, 93, 193]. And it is precisely the rise of big data, the increase in computational power, and ML's flexibility that have driven the explosion of ML techniques in the use of everyday life in the last years (e.g., facial and voice recognition, fraud or virus evasion, construction autonomous cars, improvement in medical diagnoses, etc.). On the other hand, Operations Research (OR), with an onset typically identified during the mobilization of science in the Second World War, is the use of quantitative methods by analysts to design, analyze, and improve the performance of systems, represented as a decision problem, i.e., it leverages domain knowledge to optimize [21, 89]. The most representative fields of OR application are transportation, manufacturing, and production and facilities planning. The integration of both worlds (ML and OR) in the recent years allows the construction of new algorithms where the one's weaknesses are leveled by the other's strengths. For example, using ML to detect failures and OR to optimize repair policies [202]. These advances are the reason why various organizations, included fire brigades, are looking for new ways to exploit their data and gain insights to make smarter decisions.

## 1.2/ MOTIVATION AND OBJECTIVES

Under strict confidentiality agreement with the SDIS25 and on the basis of the author's CIFRE thesis (N 2019/0372), the data provided by the SDIS25 have been used only for the analysis and experimentation of the works described in this manuscript, resulting in feasible solutions for SDIS25.

Since the studies developed in this thesis have received contributions from Héber Hwang Arcolezi and David Laymani, both from the AND team (in french *Algorithmique Numérique Distribuée*) and research department DISC (in french *Département d'Informatique des Systèmes Complexes*), the author will use *we* instead of *I* to highlight the contributions of the collaborators.

The motivation of this thesis is the main objective of a fire department: *help and protect the community*. In several countries, firefighters risk their lives voluntarily to save the lives of others, using few resources or very deteriorated resources. For this reason, this manuscript presents various solutions based on ML and OR techniques to optimize the use of fire department resources (EMS, in general), in order to strengthen their teams and improve the quality of its services. In this way, not only the victims will be protected but also the agents in service.

The thesis is divided into two main objectives:

- i *Predict interventions to support decision-making in the short-term deployment of resources*. We will seek to predict the number of interventions in the following hours, identify the peaks of interventions generated by unusual events, prioritize urgent calls, forecast the response time of the ambulances and their waiting time in hospitals. For this, we will use ML techniques.
- ii *Develop methodologies to optimize resources in the long-term and in line with growing demand*. We will seek to define a service quality indicator and simulate the operational load with various resource configurations. The integration of both will allow us to optimize the current distribution of ambulances and the positioning of a new center. For this, we will use ML and OR techniques.

Thus, we will reduce the response time of fire brigades, which brings benefits such as material, since it would be possible to strategically redistribute their scarce resources in the face of an overload of interventions; human, since advance preparation would protect firefighters (from a fire, for example) and save civilian lives (from cardiac arrest or drowning, for example); and economic, for the protection of property and because premature death has a significant social cost.

### 1.3/ CONTRIBUTIONS OF THE THESIS

The contributions of this thesis are summarized in the following:

1. *Prediction of the number of interventions*. This would allow a fire department to identify the operational load in the next few hours and if the existing resources will be sufficient to meet the demand.
2. *Forecast of the ambulance turnaround time in hospitals*. This would allow a fire department to approximate the time they would spend in the hospital from their arrival to being available again. In periods of pandemic, such as COVID-19, the number of

interventions was reduced, but because hospitals were overcrowded with patients, ambulances spent more time waiting to be attended, i.e., resources were unavailable for more time, and as a consequence, there were breakdowns in services.

3. *Prediction of the ambulance response time.* This would allow a fire department to identify if there will be external disturbances (e.g., weather, traffic, etc.) that will prolong the response time to an incident.
4. *Forecast of victim's mortality and their need for transport to health facilities.* This would allow a fire department to identify urgent calls in which the victim has a high probability of dying and send the most appropriate equipment. In addition, this would allow them to recognize if their resources would be more time unavailable when making a transfer to the hospital.
5. *Prediction of peaks in the number of interventions due to rare events.* Although a large part of the operational load is generated by human activity, there are rare events, such as natural phenomena (storms, floods, forest fires, etc.), that generate very high peaks in the number of simultaneous interventions and for a short period of time, saturating the resources of a fire department. Thus, this prediction would allow the establishment of strategies to avoid the breakdown of the system during these events.
6. *Development of the service quality indicator, named, the breakage calculation.* This would allow a fire department to measure the quality of their services, by identifying the type of breakdown, the quantity, the accumulated time, and its causes.
7. *Development of the operational load simulator.* This would allow a fire department to simulate a given operational load, experimenting with different resource configurations and measuring their efficiency with the breakage calculation.
8. *Optimization of ambulance distribution.* This would allow a fire department to identify the best redistribution of its ambulances over its entire territory, with the main objective of reducing the number of breakdowns in service.
9. *Optimization of the positioning of a new center and its configuration.* This would allow a fire department to determine the best position to establish a new center and the number of engines needed to reduce the operational load on other nearby centers.
10. *Optimization of firefighter sizing for the new center.* This would allow a fire department to experience 3 cases of agent sizing: when all agents are new, when part of the agents are new and others come from existing centers, and when all agents come from existing centers. Thus, a fire department could evaluate the cost-benefit of the sizing type and its impact on the service.

## 1.4/ THESIS OUTLINE

The rest of this manuscript is organized as follows: Chapter 2 presents the scientific background on ML and OR techniques and used metrics. Chapter 3 describes the datasets used, the operational flow of SDIS25, and the design of the breakage calculation. Chapter 4 presents the study developed for predicting the number of interventions in the next hour. Chapter 5 explains the methodology developed for predicting the ambulances' turnaround time in hospitals. Chapter 6 describes the models developed for predicting the ambulances' response time and the most influential variables. Chapter 7 presents the methodology developed to predict victim's mortality and their need for transport to hospitals, it also provides insights on the variables with the highest impact on the models. Chapter 8 presents a broad study about predicting the peaks of interventions due to rare events. Chapter 9 explains the development of the quality indicator and the optimized search of the available resources. Chapter 10 describes the construction of the operational load simulator with two ML models embedded (for the prediction of engine's depart and arrival on-scene). Chapter 11 explains the created methodology to redistribute the ambulances and reduce breakdowns in service. Chapter 12 presents the developed methodology to position a new center and determine its configuration of engines. And, Chapter 13 develops methods for sizing agents in the new center.



## BACKGROUND



# MACHINE LEARNING AND OPERATIONS RESEARCH

In this Chapter, we will briefly review the concepts and techniques of machine learning and operations research from the existing literature and which we consider in our work. Also, we will describe the metrics applied to evaluate our developed models.

## 2.1/ MACHINE LEARNING

Machine Learning is a programming paradigm, where computers learn on their own how to perform a specified task by transforming input data into rules to output meaningful answers [93].

### 2.1.1/ TASKS

Two of the most common machine learning tasks and considered in our works are:

- **Regression.** In this type of task, the computer program must predict a numerical value given some entry  $x$ , i.e., produce a function  $f : \mathbb{R}^k \rightarrow \mathbb{R}$ . For instance, predicting prices, analyzing survey data like customer product preferences, stock predictions, etc [84].
- **Classification.** In this type of task, the learning algorithm is asked to identify which of  $k$  categories an entry  $x$  belongs to, i.e., produce a function  $f : \mathbb{R}^k \rightarrow \{1, 2, \dots, k\}$ . For example, a problem might be object recognition, where the inputs are images and the categories are the object types found in the images. Other examples are the classification of handwritten characters, text classification for drug satisfaction analysis, etc [84].

### 2.1.2/ TYPES OF LEARNING

In the following, we describe the three broad categories of learning, according to [84, 93].

- **Supervised**, in which ML algorithms learn to associate some input with some output, given a training set of examples. Most of the cases the outputs (annotations or labels) are provided by a *human supervisor*. In cases where human-annotated labels are not provided, labels are generated from the input data, usually by a heuristic algorithm. These cases belong to a sub-category called **Self-supervised learning**. Some examples are the prediction of the next frame in a video, given past frames, or the next word in a text, given previous words.
- **Unsupervised**, in which ML algorithms attempt to extract information from a distribution that does not require manual recording of examples. Generally, it is used in tasks such as extracting samples or removing noise from a distribution, clustering the data into groups of related examples, finding the best representation of the data, etc.
- **Reinforcement learning**, where there is an autonomous agent that receives information about its environment to learn how to perform a task by trial and error, without any human guidance. This area got a lot of attention when Google DeepMind applied it to play Go in the match AlphaGo vs Lee Sedol, at the highest level (AlphaGo-The Movie [199]). We can find some of its applications in self-driving cars, robotics, resource management, etc.

In this manuscript, we focused on supervised learning.

### 2.1.3/ DATA PREPROCESSING AND FEATURE ENGINEERING

Before feeding the ML algorithm, we must prepare the input data and targets. In this stage, it is used data-preprocessing and feature-engineering techniques, which usually are domain specific (e.g., to tabular data, to image data, or to text data) [93]. In the following, we describe some of these techniques:

- **Vectorization**. In problems such as predicting house prices, the data already came in vectorized form. However, when processing sound, image, and text data, we have to convert them into tensors, i.e., a list of integers or a one-hot encoded vector.
- **Normalization**. It is the process of transforming data into a 0-1 range or another range. The main idea is to normalize each feature independently to have a mean of 0 and a standard deviation of 1. Decision tree-based techniques have shown

robustness when processing large values. However, in the case of neural networks, it is not safe to feed them with data containing large values, as they can trigger large gradient updates and the network might not converge.

- **Handling Missing Values.** Sometimes the data could be missing due to human error, some system failure, or any error when recording the values. In these cases, the data must be completed artificially with some mechanism such as completing with zeros when 0 is not a meaningful value, interpolation, using mean or mode, copying from previous values, etc., or in the worst case drop them. For instance, when a neural network is trained with completed data, but during test the data come incomplete, one possible solution is to copy several samples and replace some values with 0, considering that 0 has no meaning.
- **Feature Engineering.** It is the process of making a problem easier by expressing it in a simpler way. It requires using our own knowledge about the data to make the job of the model easier by applying non-learned transformations. Good features allow to solve problems using fewer resources and less data.

#### 2.1.4/ MODELING

In machine learning, the goal is to build models that can generalize known data during training and then perform well on unseen data [93].

To evaluate a model, it is recommended to split the available data into three sets: training, validation, and test. The model will be trained on the training data and evaluated (i.e., to guide the selection of hyperparameters) on the validation data. It is important not to use any sample from test to make choices about the model, including its hyperparameters. After all hyperparameter optimization is complete, the generalization error is calculated using the test set [84].

During the training process of a model, two obstacles could raise, the underfitting and the overfitting. Underfitting occurs when the model is not able to achieve a low error value on the training set, i.e., the model is not learning. Overfitting occurs when the gap between the training error and test error is too large, i.e., the model memorized the training data and, therefore, is not able to generalize on new data.

To avoid underfitting, some of the recommendations are to use more data, increase iterations during training, select the most important features, decrease the regularization, etc. In the case of overfitting, some strategies such as *K-Fold validation* or K-Fold validation with shuffling, when the dataset is reduced, could be applied.

#### 2.1.4.1/ LINEAR METHODS

In this manuscript, we used four linear machine learning techniques:

- **Least Absolute Shrinkage and Selection Operator (LASSO)**. It is a method of contracting the coefficients of the regression, whose ability to select a subset of variables is due to the nature of the constraint on the coefficients. We used the LASSO implementation from the Scikit-Learn library [13, 52].
- **Bayesian Ridge Regression (BRR)**. It estimates a linear probabilistic model. It is mainly regularized by the parameters lambda and alpha, which are the precision of the weights and noise respectively. We used the BRR implementation from the Scikit-Learn library [70, 35, 52].
- **Support Vector Machine (SVM)**. Its main goal is to find a hyper-plane that better divides a data set into classes. It is also applied to regression problems, using for example the Epsilon-Support Vector Regression method. We used the SVM implementation from the Scikit-Learn library [50, 52].
- **Prophet**. It is a forecasting tool for time series data, where trends are fit with a certain seasonality, depending on the additive or multiplicative model selected. Furthermore, it allows the addition of changepoints such as holidays, and is robust enough to missing data. In our studies, we used the library built by the author for python language [108].

#### 2.1.4.2/ ALGORITHMS BASED ON DECISION TREES

In this manuscript, two types of ensemble methods were considered: bootstrap aggregating (bagging) and boosting.

As described in [12, 75], bagging generates a number of training datasets by bootstrap sampling the original training data. These datasets generate a set of models with a single configuration. The models' outcomes are combined using averaging, if it is a regression problem, or voting, if it is a classification problem.

Boosting is another ensemble technique that trains predictors one after another, trying to correct the previous one [59, 75].

The five techniques are described below:

- **Random Forest (RF)**. It is a bagging technique. This is an ensemble learning method, which randomly generates several decision trees in parallel with subsets of the samples. A set of trees forms a random forest. In the case of regression, the result is given by the average of individual trees [24].

- **Adaptive Boosting (AdaBoost).** It is a boosting technique. It increases the relative weights of the misclassified instances at every iteration to tweak them with the predictor of the next iteration [15, 99]. Initially, all instance weights start with values  $\frac{1}{m}$ , where  $m$  is the number of training instances. In the following iterations, each weight is updated according to Eq. 2.1, where  $w^{(i)}$  is the  $i^{\text{th}}$  training instance weight,  $\hat{y}_j^{(i)}$  is the prediction of the  $i^{\text{th}}$  sample using the  $j^{\text{th}}$  predictor,  $y^{(i)}$  is the real target value and  $\alpha_j$  is the predictor weight. If the predictor is more accurate, the greater its weight. And if it is less accurate, its weight will be negative. Then, all weights are standardized and the process is repeated until the best predictor is found. To compute predictions, each weak classifier makes predictions that will be weighted using the predictor weight  $\alpha_j$ . The resulting class is the one that has the majority of weighted votes. This is represented in Eq. 2.2, where  $x$  is the instance to be predicted,  $N$  is the number of weak classifiers and  $k$  is the prediction of the  $j^{\text{th}}$  predictor. In our experiments, we used the *AdaBoostRegressor* implementation from the Scikit-learn library [52].

$$w^{(i)} \leftarrow \begin{cases} w^{(i)}, & \text{if } \hat{y}_j^{(i)} = y^{(i)} \\ w^{(i)} \exp(\alpha_j), & \text{if } \hat{y}_j^{(i)} \neq y^{(i)} \end{cases} \quad (2.1)$$

$$\hat{y}(x) = \underset{k}{\operatorname{argmax}} \sum_{\substack{j=1 \\ \hat{y}_j(x)=k}}^N \alpha_j \quad (2.2)$$

- **Gradient Boosting.** It is a boosting technique. It takes the residual errors made by the previous weak learner, which is a decision tree, to fit the new predictor, improving the model at each iteration. Thus, the prediction for each sample results on adding up the predictions of all the trees used [14, 99]. As it is described in [52] documentation and [44], Eq. 2.3 depicts the additive strategy with the created model  $F_m$  in the iteration  $m$ , where  $F_{m-1}$  is the previous ensemble model,  $h_m$  is the new tree added that is built in Eq. 2.4 while tries to reduce the loss  $L$ ,  $\gamma_m$  is the step length chosen by using line search in Eq. 2.5,  $x_i$  is the  $i^{\text{th}}$  instance and  $y_i$  is the target value of the  $i^{\text{th}}$  instance. The first model  $F_0$  is the mean of the target values when the least-squares regression is used as loss function. In our experiments, the *GradientBoostingRegressor* method from the Scikit-learn library [52] was used.

$$F_m = F_{m-1} + \gamma_m h_m \quad (2.3)$$

$$h_m = \underset{h}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + h(x_i)) \quad (2.4)$$

$$\gamma_m = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) - \gamma \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}) \quad (2.5)$$

- **Extreme Gradient Boosting (XGBoost).** It is a boosting technique. It seeks to minimize the complexity of the model in each iteration. XGBoost establishes an objective function considering the loss function  $L(\theta)$  and the regularization  $\Omega(\theta)$  on the model, penalizing its complexity to avoid overfitting. The model  $\hat{y}_i^{(t)}$  at iteration  $t$  is the combination of  $k$  trees (Eq. 2.7), i.e., a new tree that optimizes the system  $f_i(x_i)$  is added to the model  $\hat{y}_i^{(t-1)}$  created in the previous round, where  $x_i$  is the input instance. To evaluate the complexity of the tree  $\Omega(f)$ , [81] presented an approach depicted in Eq. 2.8. The first term  $\gamma T$  evaluates the number of leaves  $T$ , where  $\gamma$  is a constant, and the second term calculates  $L2$  norm of leaves scores  $w_j$ . Taking the Mean Squared Error (MSE) as instance for the loss function and computing its Taylor expansion to the second order, the objective function outcomes in Eq. 2.9 and describes how the partition of the nodes is done.  $G$  and  $H$  are defined as Eq. 2.10 and Eq. 2.11 respectively, where  $g_i$  and  $h_i$  are the first and the second order partial derivative after applying the Taylor expansion,  $I_j = \{i|q(x_i) = j\}$  are the samples assigned to the  $j$ -th leaf and  $q(x)$  is the tree structure. Lastly, from the objective function, the argument of the minimum and the minimum of the quadratic function for the variable  $w_j$  are taken, where  $q(x)$  is fixed and  $\lambda$  is a small constant, the outcomes are Eq. 2.12 and Eq. 2.13, where the latter evaluates the score of the tree structure, i.e., if it is smaller, it is better [81]. In this manuscript, we used the library built by the author [81] for python language.

$$obj(\theta) = L(\theta) + \Omega(\theta) \quad (2.6)$$

$$\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_i(x_i) \quad (2.7)$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (2.8)$$

$$obj^{(t)} = \sum_{j=1}^T [G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2] + \gamma T \quad (2.9)$$

$$G_j = \sum_{i \in I_j} g_i \quad (2.10)$$

$$H_j = \sum_{i \in I_j} h_i \quad (2.11)$$

$$w_j^* = -\frac{G_j}{H_j + \lambda} \quad (2.12)$$

$$obj^* = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T \quad (2.13)$$

- **Light Gradient Boosting Machine (LGBM)**. It is a gradient boosting decision tree technique. It establishes a leaf-wise tree growth strategy to speed up computation and reduce memory consumption. In this manuscript, we used the library built by the author for python language [102]. .

#### 2.1.4.3/ NEURAL NETWORKS FOR TABULAR DATA

Five techniques were used in our experiments with tabular data. These are described as follows:

- **Multi-Layer Perceptron (MLP)**. It is an artificial neural network of the feed-forward type, since the information flows through several neurons organized in interconnected layers classified as: input, intermediate (which can be more than one) and output [84]. In our studies, we used the implementation developed by the Scikit-learn library [71, 52] and we also built our own architectures with the Keras library [74].
- **Convolutional Neural Network (CNN)**. Convolutional neural networks (CNN) is a class of deep neural networks, most often applied to visual image analysis. A CNN is generally build around the following layers: 1) Convolutional layers are composed of neurons whose purpose is to detect patterns (features map) from their inputs, 2) Pooling layers whose purpose is to reduce the feature map dimensionality in order to be more computationally efficient. These layers are often chained after convolutional layers, and 3) These two previous set of layers are generally followed by one or more fully connected layers [84]. In our studies, we built architectures with CNN layers with the Keras library [74].
- **Long Short-Term Memory (LSTM)**. It has emerged as effective and scalable model for several learning problems related to sequential data (e.g., handwriting recognition, speech recognition, human activity recognition and traffic prediction), and it does not suffer from the effect of the vanishing or exploding gradient problem as simple recurrent networks do [58]. One of the central keys behind LSTM success is its memory cell, which can maintain its state over time by learning what to store in the long-term state, what to throw away and what to read from it by passing on nonlinear gating units regulating the information flow into and out of the cell. Initially, the long-term state  $c_{(t-1)}$  goes through a forget gate, dropping out some memories, and then it adds some new memories that were properly selected by an input gate. The result  $c_{(t)}$  is sent straight out without any further transformation, i.e., at each time step, some memories are dropped and others are added. Furthermore, after passing by the input gate, the long-term state is copied and passed through an

activation function  $\tanh$ , and finally the result is filtered by an output gate. This procedure generates the short-term state  $h(t)$  that is equal to the cell's output for this time step  $y(t)$ .

Mathematically, it is expressed as:

$$i_{(t)} = \sigma(W_{xi}^T \cdot x_{(t)} + W_{hi}^T \cdot h_{(t-1)} + b_i) \quad (1)$$

$$f_{(t)} = \sigma(W_{xf}^T \cdot x_{(t)} + W_{hf}^T \cdot h_{(t-1)} + b_f) \quad (2)$$

$$o_{(t)} = \sigma(W_{xo}^T \cdot x_{(t)} + W_{ho}^T \cdot h_{(t-1)} + b_o) \quad (3)$$

$$g_{(t)} = \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot h_{(t-1)} + b_g) \quad (4)$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \quad (5)$$

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)}) \quad (6)$$

where  $W_{xi}$ ,  $W_{xf}$ ,  $W_{xo}$  and  $W_{xg}$  are the weight matrices of each four layers for their connection to the input vector  $x_{(t)}$ ;  $W_{hi}$ ,  $W_{hf}$ ,  $W_{ho}$  and  $W_{hg}$  are the weight matrices of each four layers for their connection to the previous short-term state  $h_{(t-1)}$ ; and  $b_f$ ,  $b_g$ ,  $b_i$  and  $b_o$  are the bias terms of each of the four layers. For more details about LSTM architecture, interested readers are strongly advised to read and consult [17, 99, 58, 100]. In our studies, we built architectures with LSTM layers with the Keras library [74].

- **Attentive Interpretable Tabular Learning (TabNET)**. It is an interpretable canonical deep learning architecture for tabular data. It uses instance-wise feature selection strategy (attentive transformer) to improve nonlinear processing without falling into overfitting. In our studies, we used the library built by the author for python language [119].

#### 2.1.4.4/ NEURAL NETWORKS FOR TEXT PROCESSING

We also perform experiments with Natural Language Processing (NLP). For this, we used the LSTM and CNN neural networks, previously described, and Transformers.

Transformers models provides a solution by processing the whole sequence all at once, i.e., to no longer process texts sequentially by Recurrent Neural Networks (RNN) or LSTM. Furthermore, it uses the attention mechanism which allow capture different types of relationships between tokens. A Transformer is built based on an encoder and a decoder, each of them consisting of a stack of attention and dense layers. Since 2017 and the first Transformer model, many models have been developed such as ELMo, GPT-

2 and GPT-3, BERT, XLNet, RoBERTa, Turing-NLG, etc. The main advantages of the Transformer model are: i) The distance between 2 tokens is no longer a parameter taken into account by the model (the model can take into account long-term dependencies), ii) The attention matrix calculation allows to parallelize the process of encoding and then decoding the sequences, thus accelerating the calculations, and iii) No labeled data are required to pretrain these models and it is then possible to train a transformer-based model by providing a huge amount of unlabeled text data. The latter means that it is possible to do transfer learning with this trained model in order to perform other NLP tasks like text classification, named entity recognition, text generation, etc.

In June 2017, Google presents BERT (Bidirectional Encoder Representations from Transformers) [124]. It is a Transformer composed of a suite of encoders only ( $N = 12$  or  $24$  depending on the version: base with 110 millions parameters or large with 340 millions parameters). Bert was originally pretrained by using two tasks. It hides some of the words (15%, although this is actually more complex) and learns how to find them. This allows him to acquire a general and bi-directional knowledge of the language. BERT also learns to recognize if two sentences are consecutive or not. The corpus used for this pretraining was the BooksCorpus with 800M words and a version of the English Wikipedia with 2,500M words. When it came out, BERT was able to outperform state of the art models for a large set of NLP tasks such as GLUE (General Language Understanding Evaluation) or SQuAD (Stanford Question Answer Dataset).

In this manuscript, we used two pretrained models based on the BERT architecture and French corpus.

- **CamemBERT**. It is based on RoBERTa, which improves the original implementation of BERT by using dynamic masking and training with larger batches and for longer. However, the main differences between RoBERTa and CamemBERT are that the latter uses the whole-word masking and SentencePiece tokenization instead of WordPiece. CamemBERT has been pre-trained on the French subcorpus (138GB) of a huge multilingual corpus (Common Crawl Oscar corpus) and proved that crawled data with high variability is preferable to using Wikipedia-based data [133].

- **FlauBERT**. It is also a RoBERTa-based model for French. It has been pre-trained on less but more edited data (71GB). Its text corpus consists of 24 sub-corpora collected from different sources with diverse subjects and writing styles. Its performance is very close to CamemBERT. In fact, in its original paper, it is demonstrated that the performance of a parsing model can be improved with an ensemble of FlauBERT and CamemBERT [130].

## 2.2/ OPERATIONS RESEARCH

Operations research is the use of quantitative methods by analysts to design, analyze, and improve the operation of systems, represented as a decision problem or mathematical model [21, 89].

Fig.2.1 illustrates the operations research process. First, the decision problem must be identified. Then, the problem is modeled, i.e., describe approximately the behavior of the system, for this, the variables and their relevant relationships must be defined. Then, the constructed model is analyzed to extract conclusions. Remember that the model is an approximate representation of the real problem, so the conclusions must be as significant as possible. From these conclusions, we will infer decisions. Finally, the decisions are evaluated to determine their possible implementation in real life [89].

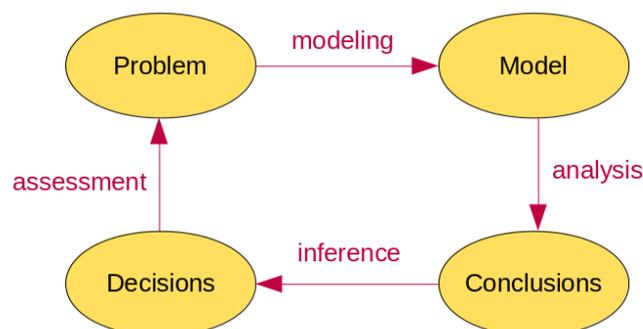


Figure 2.1: Scheme of the operations research process [89].

A constructed model is a simplified representation called mathematical model because its components are mathematical structures (e.g., functions, equations, matrices, operators, etc). The processes of solving the model involves applying mathematical process or quantitative tools to estimate the behaviour of the system [21]. In the following sub-sections, we will describe the operational research tools analyzed and considered in the development of our works.

### 2.2.1/ INTEGER PROGRAMMING

Mathematical programming problems with variables constrained to have integer values are called integer programming problems. In real life, many industrial and financial applications set these constraints, e.g., the manufacture of a number of airplanes demands an integer value, since a fractional value would be infeasible. Thus, the feasible region of an integer programming problem is neither continuous nor convex. While the formulations of integer programming problems are quite similar to continuous mathematical programming problems, the restriction that several or all variables should be integers makes it much more difficult to find a solution, from a computational point of view. Hence, many integer programming problems belong to the *Non-Deterministic Polynomial-Time Hardness* (NP-Hard) class. This means that while a general linear programming problem can be solved in polynomial time, finding an optimal integer solution for the same formulation requires an exponential amount of computational time [21].

The case where all decision variables have to be integers, the problem is named as *General Integer Programming Problems*. In the case where all variables are restricted to values zero or one (binary or boolean constraints), the problem is called *Zero-One Programming Problem*. And, when some of the variables must be integers, others must be zero or one, and others must be continuous, the problem is named as *Mixed Integer Programming Problem*. These cases were considered in the development of methodologies for the optimization of firefighters' resources. In the following, we will present examples of each case:

#### 2.2.1.1/ GENERAL INTEGER PROBLEMS

An example of general integer problem is the *employee scheduling problem*, in which we define a number of shift patterns for employees and a number of employees that will work for a specific pattern. The objective is to minimize total salary costs of workers and ensure their availabilities in each shift.

#### 2.2.1.2/ ZERO-ONE PROBLEMS

In this case, all variables are binary and denote whether or not some assignment takes place or whether or not some activity occurs. Some examples are *production scheduling environment*, *distributed computing*, *vehicle routing*, and *examination timetabling*. For instance, in production scheduling environment, we could define the variable  $a_{jm}$  to take the value 1 if a job  $j$  is assigned to a machine  $m$ , and zero otherwise. Thus,  $a_{jm}$  would allow creating restrictions on the available time of the machines, the time limit to complete a job, and the consequent total costs. A distributed computing problem emerges in a

multiprocessor computing environment. The objective is to reduce the total execution and communication costs of allocating programs and files to various machines in diverse locations. For this, we could define the variable  $x_{ij}$  to take the value 1 if module  $i$  is assigned to processor  $j$ , and zero otherwise.

As an example of the mathematical formulation of a zero-one programming problem, we present the classical formulation of the *assignment problem*, mainly analyzed in our optimization works. Its formulation is described in 2.14a, and extracted from [21]. In 2.14a, the term  $c_{ij}$  denotes the cost of assigning job  $i$  to employee  $j$ ,  $x_{ij}$  represents the activation of the assignment, the constraint 2.14b indicates that every job has to be assigned to just one employee, and 2.14c indicates that every employee has to do just one job. The objective is to minimize the cost of the assignments.

$$\min \sum_i \sum_j c_{ij} x_{ij} \quad (2.14a)$$

$$\text{s.t.} \quad \sum_j x_{ij} = 1 \quad \forall i = 1, \dots, n, \quad (2.14b)$$

$$\sum_i x_{ij} = 1 \quad \forall j = 1, \dots, n, \quad (2.14c)$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall i, j \quad (2.14d)$$

### 2.2.1.3/ MIXED INTEGER PROBLEMS

An example of this case is the *warehouse location problem*, where given a set of possible locations for the positioning of the warehouses, the objective is to select the best positions that minimize the delivery total cost from the warehouses to the clients. For this, we could define the zero-one variables  $x_j$  to have the value 1 if the location  $j$  is chosen. Then, we will have to solve a transportation problem, where the real-valued variables  $y_{ij}$  will represent the quantity of transported product from the producer  $i$  to the warehouse  $j$ , and the real-valued variables  $z_{jk}$  will indicate the quantity of distributed product from warehouse  $j$  to customer  $k$ . The total cost will be based on the distances that the products travel.

### 2.2.2/ MULTI-OBJECTIVE OPTIMIZATION

In real life, many decision problems evaluate multiple criteria that often conflict. For example, imagine that we have a low market budget and we try to buy the necessary products of higher quality but at the lowest price. We could say, in a general way, that we are seeking to optimize several objectives simultaneously and generally in opposition, where there is not a single optimal solution, but rather a set of good solutions. This set will be

presented to the decision maker, who will choose according to his convenience [77].

In [1], the mathematician Vilfredo Pareto was the first to study the aggregation of conflicting goals. The author introduced the definition of efficiency or *Pareto Optimality* (central definition of *Multi Criteria Decision Making* - MCDM), i.e., given a set of resources and a set of individuals, a Pareto-optimal allocation of resources means that it is not possible to benefit more to one individual without harming another. The MCDM presents 2 types of problems:

- **Multiple-criteria evaluation problems**, where solutions known a priori are compared to recover the best one.
- **Multiple-criteria design problems (multiple objective mathematical programming problems)**, where the optimization of a mathematical model is performed to find possible solutions to the problem. Here, we find the metaheuristics.

### 2.2.2.1/ DECISION AND OBJECTIVE SPACE

There are 2 concepts to differentiate: the decision space or search space and the objective space, illustrated in Fig. 2.2. The decision space is the domain of the function or functions to be optimized and is multidimensional. The objective space corresponds to the co-domain of the function or functions. In single-objective problems, each decision vector corresponds to a scalar number, i.e., the objective space is one-dimensional. In multi-objective problems, each objective function to be optimized represents one dimension, i.e., the objective space is multi-dimensional [77].

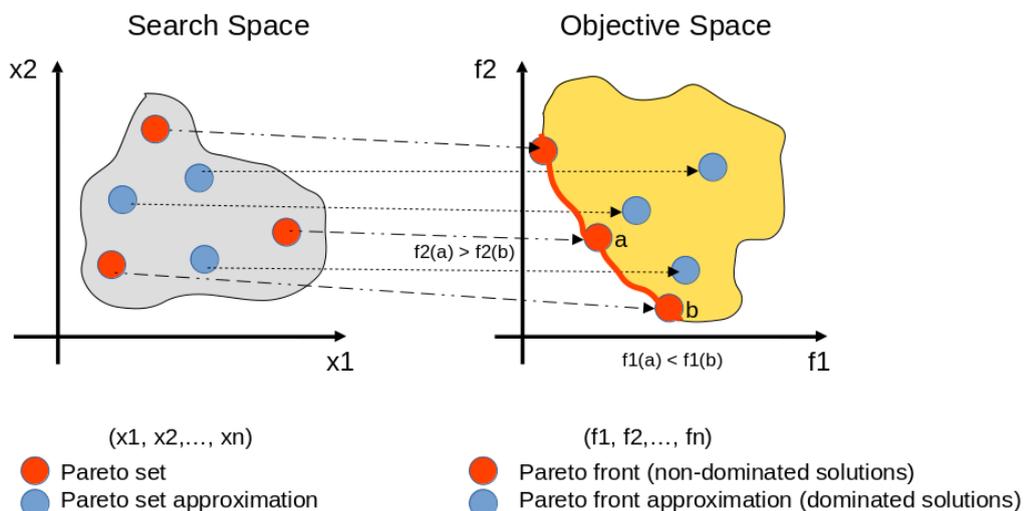


Figure 2.2: Illustration of the search and objective space.

### 2.2.2.2/ PARETO DOMINANCE

In a multi-objective problem there is a set of  $n$  solutions to be optimized. For each solution of the search space, an  $n$ -dimensional vector of variables is obtained. In this case, to identify the best solutions, the Pareto criterion is used. A solution  $a$  belongs to the set of Pareto-optimal solutions if there is no a solution  $b$  that could improve one of the objectives without worsening at least one of the others. This is formally defined in Eq. 2.15a.

$$\min f(x) = [f_1(x), f_2(x), \dots, f_n(x)] \quad (2.15a)$$

$$\text{s.t. } g_i(x) \leq 0, i = 1, \dots, q, \quad (2.15b)$$

$$h_j(x) = 0, j = q + 1, \dots, m \quad (2.15c)$$

Where  $x = [x_1, x_2, \dots, x_n]$  is the vector of decision variables,  $f(x) = [f_1(x), f_2(x), \dots, f_n(x)]$  are the objective functions to minimize, and  $g_i(x)$  and  $h_j(x)$  are the functions that represent the constraints and define the feasible and non-feasible areas of the search space.

In Fig. 2.2, we can identify the Pareto set, represented by the set of points in the search space to which the best solutions of the problem correspond (red points), and the Pareto front, composed of the set of optimal solutions in the target space or also known as non-dominated solutions (red points). The Pareto front can be linear, concave, convex, etc., depending on the objective functions. All these solutions are equally good. For this reason, the goal of multi-objective optimization is to find the Pareto front subject to resource constraints [77]. Therefore, we could conclude:

- $a$  is better than  $b$ ,
- $b$  is better than  $a$ ,
- $a$  and  $b$  are equally good, and
- $a$  and  $b$  are incomparable

Finally, in [6] and [5], it was shown that the set of optimal solutions increases exponentially as the number of dimensions of the objective space increases.

### 2.2.2.3/ AGGREGATED FUNCTIONS

The first approach used to solve multi-objective problems was to apply single-objective techniques, i.e., to convert a set of objectives into a scalar. For this, each objective function is assigned a weighting factor, which represents the relative importance compared to the other objectives. Also, the sum of the factors must equal 1.

On the one hand, the advantages of this approach are the reuse of single-objective techniques and better performance if we use few objectives with a convex search space. On the other hand, the disadvantages are the difficulty of recognizing the weighting factors and generating members of the Pareto front when the search space is concave.

#### 2.2.2.4/ NON-DOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)

The NSGA technique is based on the structure of a genetic algorithm that classifies individuals according to the Pareto ranking and diversifies them through a sharing procedure. Since the best individuals receive a higher fitness value, the search is more developed in the non-dominated zones. However, the algorithm presents a high computational complexity when it orders and classifies the individuals, it does not use elitism to improve convergence, and it needs to define the sharing parameter [11]. Due to this, in [25], the authors developed the NSGA-II algorithm, an improvement of the NSGA, which is characterized by reducing the complexity of the classification phase, adding elitism, and replacing the sharing method with the crowded operator. The latter compares how populated the areas near each solution are and improves the performance of the selection phase. Fig. 2 of [25] illustrates the NSGA-II procedure and this is the algorithm that we have followed and adapted in our resource optimization works.

#### 2.2.3/ BAYESIAN OPTIMIZATION

Bayesian optimization (BayesOpt) is an approach to optimize objective functions that is computationally complex. It has been shown to be tolerant to stochastic noise in function evaluations. It creates a surrogate for the objective and computes its uncertainty using a Bayesian machine learning technique (e.g., Gaussian Process Regression). Then, an acquisition function is defined from this surrogate to generate samples. Finally, a new configuration is selected and tested in the original model [114].

BayesOpt focus on solving the problem 2.16, where  $x$  is in  $\mathbb{R}^d$  for a value of  $d$  that is not too large (recommended  $d \leq 20$ ). The feasible set  $A$  is a simple set, where is easy to assess a membership. And, the objective function  $f$  is continuous and expensive to evaluate [114].

$$\min_{x \in A} f(x) \tag{2.16}$$

The basic structure of Bayesian Optimization consists of two main components:

- **Surrogate Model.** This is a method for statistical inference. Probabilistic surrogate models are used to balance exploration and exploitation by estimating uncertainty. For this, the Gaussian Process is generally used as a surrogate model. The idea is that the values of the objective function correspond to realizations of a multivariate Gaussian process, where each  $x$  returns a mean and a variance [114, 126].
- **Acquisition Function.** This is also called *utility function* or *infill criterion* and is used for deciding where to sample from. The most commonly used are expected improvement, entropy search, and knowledge gradient [114, 126].

The interaction of both components can be visualized in Fig. 2.3, where the Bayesian optimization process is summarized.

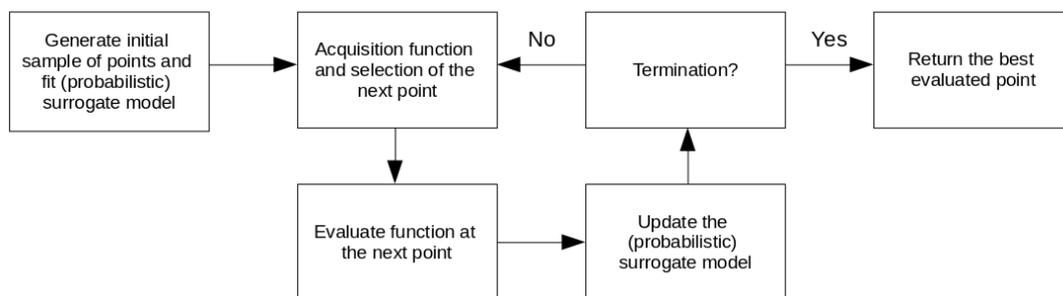


Figure 2.3: A schematic representation of the Bayesian optimization process [126].

To apply Bayesian Optimization in our works, we have used [67] and HyperOpt [61] libraries. In the case of predicting interventions we used them to find the best configuration of hyperparameters and for resource optimization we applied them to identify the best combination of resources.

## 2.3/ PERFORMANCE METRICS

In this section, we describe metrics from the literature used to measure the performance of our ML regression and classification models.

### 2.3.1/ METRICS FOR REGRESSION PROBLEMS

The metrics used in the regression tasks are described below.

Variables:

$n$ : number of samples.

$y_i$ : target value of sample  $i$ .

$\hat{y}_i$ : predicted value for sample  $i$ .

- **Root Mean Square Error (RMSE).** In Eq. 2.17, it is defined as the square root of the average squared error between the actual and predicted values. In RMSE, the outliers get more attention and dominance in the final error.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.17)$$

- **Mean Absolute Error (MAE).** In Eq. 2.18, it is defined as the average absolute difference between the target and predicted values. Since in MAE the difference is not squared, there is no dominance of the outlier reflected in the final error, making MAE more robust than RMSE.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.18)$$

### 2.3.2/ METRICS FOR CLASSIFICATION PROBLEMS

Before describing the metrics used, we will present the error matrix, also known as **Confusion Matrix**. This matrix allows us to visualize in tabular form the performance of our models.

Fig. 2.4 shows the confusion matrix for a binary classification. The main diagonal represent the number of true positive cases (*TP*) and the number of true negative cases (*TN*), i.e., it denotes the samples correctly classified for classes 0 and 1, respectively. The antidiagonal represent false negative cases (*FN*) and the number of false positive cases (*FP*), i.e., it denotes the samples incorrectly classified for classes 0 and 1, respectively.

		Predicted Values	
		Positive Class (1)	Negative Class (1)
Real Values	Positive Class (1)	TP	FN
	Negative Class (0)	FP	TN

Figure 2.4: Illustration of the confusion matrix for binary classification.

From the confusion matrix we will define the metrics used in our classification tasks below, considering  $n$  as the total number of samples.

- **Accuracy (ACC).** In 2.19, accuracy is the fraction of correct predictions made by our model.

$$ACC = \frac{(TP + TN)}{n} \quad (2.19)$$

- **Recall / Sensitivity / True Positive Rate (TPR).** In 2.20, recall represents the proportion of the real positives that has been correctly classified by the model.

$$Recall = \frac{TP}{TP + FN} \quad (2.20)$$

- **Precision.** In 2.21, precision denotes the proportion of the predicted positives got identified correctly.

$$Precision = \frac{TP}{TP + FP} \quad (2.21)$$

- **F1-Score.** In 2.22, F1-score represents the harmonic mean of precision and recall.

$$F1 - Score = \frac{2 * Precision * Recall}{(Precision + Recall)} \quad (2.22)$$

- **Specificity / True Negative Rate (TNR).** In 2.23, specificity is the proportion of true negatives that were correctly identified by the model.

$$Specificity = \frac{TN}{TN + FP} \quad (2.23)$$

- **False Positive Rate (FPR).** In 2.24, FPR is the proportion of the negative class that was misclassified by the model.

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity \quad (2.24)$$

- **Balanced Accuracy (BACC).** In 2.25, BACC is a weighting of the sensitivity and specificity.

$$BACC = \frac{1}{2} * (Sensitivity + Specificity) \quad (2.25)$$

- **Receiver Operating Characteristic Curve (ROC Curve).** It is a graph that illustrates the performance of a binary model at all classification thresholds. It compares the fraction of correct positive class predictions (TPR, y-axis) versus the fraction of errors for the negative class (FPR, x-axis), both axis with values from 0 to 1. An unskilled classifier will show an antidiagonal line, while a classifier with perfect skill will show a steep curve at the top-left of the graph, i.e., TPR and FPR will tend to be 1.

- **Area Under the Receiver Operating Characteristic Curve (AUC).** It calculates the area under the ROC curve and its values are between 0 and 1. The higher the AUC of a classifier the better it is.

## 2.4/ CONCLUSION

In this section, we have reviewed basic concepts of machine learning and operations research, the techniques considered in our experiments, and the metrics used to evaluate our models.

Among the basic concepts of ML, we mentioned the tasks developed (classification and regression) and the 3 major learning areas (supervised, unsupervised, and reinforcement learning). Furthermore, we gave some recommendations for data preprocessing, feature engineering, and modeling. Next, we describe the algorithms used as linear models, decision tree-based models, and neural network models for processing tabular and text data. In the case of operations research, we reviewed concepts and models of optimization such as integer programming problems, multi-objective optimization problems, Pareto dominance, NSGA, and Bayesian optimization.

## DATASETS AND OPERATIONAL FLOW

In the previous Chapter, we have reviewed the definitions of the techniques that we used. In this Chapter, we will present the sources and data used during the development of all the works in this thesis.

### 3.1/ INTERNAL DATA

In a strict confidentiality agreement, the main dataset is from the Departmental Fire and Rescue Service of Doubs. The data contain a set of interventions from 2006 to 2021, calls, operators, victims, the departure of engines and agents, characteristics of the centers and their resources, deployment plan (in french PDD), and the geometries of the geographical divisions in the Doubs territory.

For a better understanding of the terms, from now on, the firefighters will be named as *agents* and an engine and its crew as *armament*.

The dataset is briefly detailed below:

1. **Geometries.** It contains a set of shapefiles for *communes* (city), *quartiers* (districts), and *groupements* (territorial clusters), where a groupement contain several communes and a commune contain 0 or more quartiers.
2. **Calls.** It contains the delay time to answer the phone, the total call duration, and the delay time to diffuse the alert (i.e., to notify a center with an available armament), and the intervention's identifier.
3. **Operators.** The operator is a call center agent. Their features are: agent's identifier, age, gender, grade (e.g., caporal, captain), and seniority (experience time in days).
4. **Interventions.** Their features are: intervention's identifier, geolocation coordinates in Lambert-93, commune, start and end datetime, and its type, which is divided in three subcategories: the type of operation (e.g., aid to person, fire), the subtype of

operation (e.g., fire on the public road, fire in an individual room), and the departure's motive (e.g., cardiac arrest, respiratory distress).

5. **Centers.** It contains information about all the centers deployed on the Doubs territory. Its features are: center's identifier, name, type, and geolocation coordinates in Lambert-93.
6. **Deployment Plan.** It provides the coverage of the centres deployed in the territory. It contains a list of the communes and their quarters by pairs. Each pair *commune-quartier* has an assigned zone (i.e., Z1, Z2, and Z3), belongs to a groupement, and has a list of centers organised by geographic coverage priority, i.e., if the first center in the list has an available armament, it will be the first center to cover the intervention happening in the pair, otherwise, it will be the second center and so on. The zone provides the time limit for a center and its engines to reach the intervention site.
7. **Victims.** Their features are: age, gender, the commune where the intervention occurred, and the organization to which the victim was transported (e.g. hospitals, clinics, morgue, etc).
8. **Engines Departures.** It contains the intervention's identifier, the type of operation, the center from where the engine departed, the engine registration number, the type of engine, the datetimes of the alert on the center, the departure from the center, the arrival at the scene, the departure from the scene, the arrival to the hospital, the departure from the hospital, and the end of its mission.
9. **Engines per Center.** It contains the number of engines by type and center.
10. **Engines Crew.** It presents a list of all types of engines, where each engine has a number of agents to depart depending on adapted or degraded mode, the functions of agents, and the skills that an agent must have to be part of the crew.
11. **Agent Departures.** It contains the intervention's identifier, the center from where the agent departed, the engine of which he was part of the crew, the agent registration number, the agent status (i.e., professional or volunteer), grade, function, the datetimes of the alert, the departure, and the end of its mission.
12. **Agent Availability.** It provides the availability of the agent by his registration number, worked period (start and end datetime), center, and status.
13. **Agent Skills.** It contains the skills obtained over the years by an agent to operate an engine. It contains the agent registration number, the center, the status, the skill, date of obtaining the skill, and expiration date.

## 3.2/ EXTERNAL DATA

To enrich our internal dataset, we also use data from external open sources described below:

1. **Météo-France (tabular data) [209].** We imported hourly meteorological features from three stations, located in Dijon-Longvic, Bale-Mulhouse, and Nancy-Ochey: precipitation, temperature, barometric trend, pressure, humidity, dew point, wind direction, wind speed and gust speed.
2. **Météo-France (text data) [209, 210].** The previous tabular data were supplemented by textual vigilance alert bulletins. They contain the type of vigilance (heat-wave, extreme cold, snow or ice, thunderstorms, strong winds), the beginning and end of the vigilance period, the level of the alert (green, orange or red), a detailed description of the risk (including the locations impacted, the conditions to be expected, etc.), as well as a set of very detailed advice to users.
3. **Hydro [201].** It provided us with the hydrometric height of the rivers by hour in the region. The data was extracted from twelve stations: L'Allan à Courcelles-lès-Montbéliard, Le Doubs à Voujeaucourt, Le Doubs à Besançon, La Loue à Ornans, L'Ognon à Montessaux, L'Ognon à Bonnal, Le Dessoubre á Saint-Hippolyte, Le Doubs á Mouthe, Le Doubs á Mathay, Le Drugeon á Rivière Drugeon, Le Gland á Meslières and La Loue á Vuillafans. Also, we generated features such as the average, the standard deviation, the maximum height of the records captured in one hour, and an indicator to represent if the height of the river exceeds a limit established as a flood alert.
4. **Atmo [195].** We imported hourly air pollution features such as PM2.5 fine particles, PM10 fine particles, ozone, and nitrogen oxide from the stations Besançon Prévoyance and Montbéliard Centre.
5. **Sentinelles [212].** It provided us with weekly epidemiological statistics related to the incidence of chicken-pox, influenza and acute diarrhea.
6. **Bison-Futé [196].** It gave us daily traffic features related to the departure and the return in the form of color indicators, i.e., according to the region the days are categorized as green = fluid traffic, orange = dense traffic, red = difficult traffic, and black = to avoid because of traffic jams and slow traffic.
7. **Data Gouv [211].** We extracted daily information on the situation of the COVID-19 pandemic from March 2020 to June 2020 for the region of Doubs. The data were: the number of hospitalized individuals, the number of patients in reanimation

or critical care, the cumulative number of individuals who returned home, and the cumulative number of individuals who died.

- 8. Institut National de la Statistique et des Études Économiques (INSEE) [207].** We imported the *insee code* for all communes in the Doubs region. The insee code is a numerical indexing code to identify various entities (e.g., communes, departments, etc) in France.
- 9. Collectivites Locales [198].** We imported the agglomeration of communes and their respective list of communes.
- 10. Google Trends.** We used Pytrends library to extract hourly data from this source. The data were in a scale from 0 to 100 for the keywords in french: 'H1N1', 'coronavirus', 'SARS', 'Influenza', 'COVID-19', 'diarrhee' (diarrhea), 'grippe' (flu), 'varicelle' (chickenpox), 'incendie' (fire), 'inondation' (flood), 'greve' (strike), 'samu', and 'suicide'.

### 3.3/ OPERATIONAL PROCESS FLOW

In this section, we will describe the operational process that SDIS25 performs to respond to an emergency and the flow design to recognize and calculate breaks in service.

The SDIS25 has a call center and 71 centers deployed in the Doubs territory. The emergency service is activated through two telephone numbers (18 or 112). Next, the call is processed by an operator, who collects the necessary information about the emergency (victim data, location, reason for the call, etc.) and notifies the center with an available armament and closest to the scene to respond to the emergency (intervention). Once the armament is ready, the engine departs to the emergency scene and upon arrival, the crew provides first aid treatment to the victim(s). After that, depending on the victim's status, the victim(s) is transported to hospital, private clinics, morgue, etc. Last, the engine and its crew return to the center and are available again to attend other intervention. Fig. 3.1 shows an overview of the aforementioned operational process.

During the operational process, it is possible that the closest centers to the scene do not have sufficient resources at the right time to deal with the emergency. The causes may be the lack of budget to acquire more resources (e.g., less budget, greater increase in interventions over time), the operational overload in a given hour (e.g., occurrence of many simultaneous interventions due to a flood), a failure in the distribution of resources for the centers (e.g., failure caused by a high season that could have been predicted), etc. The consequences of this are the potential damage to the territory and the life of the residents, i.e., economic, social and environmental vulnerability. Therefore, we need a way

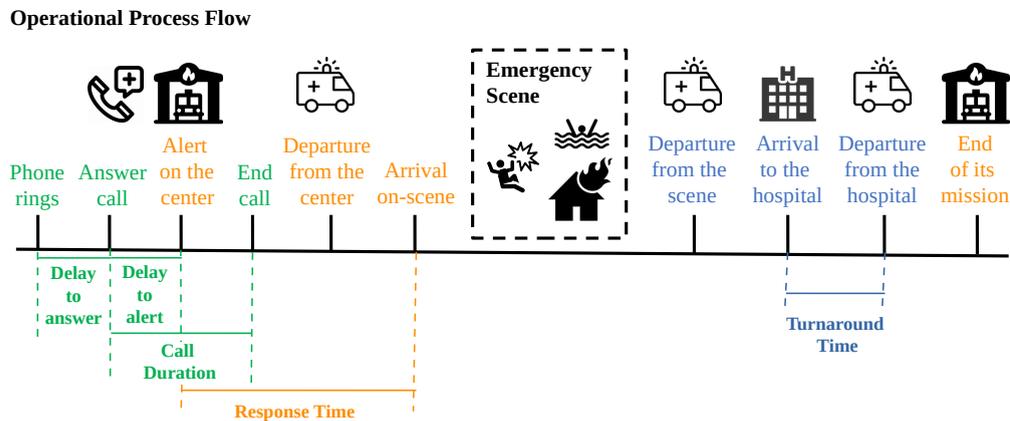


Figure 3.1: Overview of operational process of SDIS25 when dealing with an intervention. In green, we can see the times related to the emergency call received at the call center. In orange, the main times related to the displacement of the available armament. And in blue, the times related to a possible displacement of the armament taking the victim to a hospital, clinic, morgue or other.

to measure the service quality to improve it, developing resource distribution strategies and anticipating disasters. For this, we will start describing the resources below.

The centers are divided into 5 categories: principal rescue centers (quantity=4, abbreviation CSP, in french Centre de Secours Principal), reinforced rescue centers (quantity=11, abbreviation CSR, in french Centre de Secours Renforcé), rescue centers (quantity=24, abbreviation CS, in french Centre de Secours), reinforced first intervention centers (quantity=8, abbreviation CPIR, in french Centre de première Intervention Renforcé), and first intervention centers (quantity=24, abbreviation CPI, in french Centre de première Intervention). The main difference between the types of centers is the amount of armament they possess. First intervention centers have first aid and other engines, while the other types of centers have first aid, adapted, and other engines. We are going to focus mainly in 2 engine mode types as follows. A first aid engine or non-adapted engine is a simple engine with a minimum number of agents, while an adapted engine is a type of engine specific to the type of operation and with a theoretical number of agents. When an adapted engine does not have the requested number of agents it becomes a non-adapted engine. In general, each type of engine has a minimum and theoretical number of agents, each agent fulfills a function, and for each function there are several types of skills that the agents have according to the training they received. For example, in the case of an intervention of type *rescue people*, a *rescue and assistance to victims vehicle* (VSAV, in french Véhicule de Secours et d'Assistance aux Victimes) is considered an adapted engine if it has 3 agents. Each agent must fulfill the functions: driver (COND, in french Conducteur), engine chief (CA, in french Chef d'Agrès), and crewman (EQUI, in french Équipier). For this, the agents must have the skill codes: VSCOND, VSCA, and VSEQUI,

Table 3.1: Time frame for first aid and adapted engines according to the zone.

Zone	First Aid Engine	Adapted Engine	
		Without First Aid Engine	With First Aid Engine
Z1	10	10	10
Z2	20	20	25
Z3	20	20	25

respectively. The real skill codes have been replaced by fictitious ones to protect agents' privacy.

An intervention is located in a commune and quartier. Each commune-quartier is categorized as a zone: Z1 (urban), Z2 (semi-urban), or Z3 (rural). In the deployment plan, updated at least once a year, each commune-quartier is protected by a list of defense centers according to its proximity and zone. When an adapted or first aid engine departs to an intervention, depending on the zone, it has a time frame to arrive at the scene. These time frames are described in Table 3.1. Initially, an emergency must always be assisted by an adapted engine, but sometimes an adapted engine is not available. Thus, a first aid engine is sent until an adapted engine becomes available and is sent to the intervention, and if needed, other types of engines will enter in action to help in the intervention, which do not have a time frame. Generally, the other types of engines are support engines (e.g., vehicle with swivel ladder, light medical rescue vehicle, etc).

With all these elements in mind, the SDIS25 designed a modality for calculating the breakdowns in service. A breakdown is defined as the inability to ensure relief within the time frames set as objectives. There are 2 types of breakdowns: *breakdowns to the speed contract* (BSC) and *breakdowns of public service* (BPS). BSCs occur when an engine fails to arrive on scene within the time frame and BPSs are generated when there is no adapted engine available in the centers to handle simultaneous interventions. In Fig. 3.2, we can see in green the cases in which there are no breakdowns (R1 and R2) since the adapted engine arrived on the scene either from the 2 main centers with adapted armament or from the other centers (generally with less adapted armament and further from the location of the intervention). In yellow the cases in which BSC occur (R3) and in orange the cases in which BSC and BPS occur (all C cases). From now on, a center with adapted armament for the type of intervention will be named as an *adapted center*.

To calculate the breakdown times according to the zones (i.e., Z1, Z2, and Z3) and depending on the case, we have detailed in Figures 3.3, 3.4, 3.5, and 3.6 the times in which the first aid and adapted engines arrive at the scene, whether or not there was another adapted engine available in centers 1 or 2 of the defense list, time frames, the BSC time in yellow, and the BPS time in red. To be more specific, in Fig. 3.3, we calculate only the BSC time for case R3, i.e., the adapted engine arrived late to the scene and there is another adapted engine available at the first adapted center on the defense list. In Fig. 3.4, we calculate the BSC and BPS break times for the 3 cases: C2, C2BIS, and C2TER.

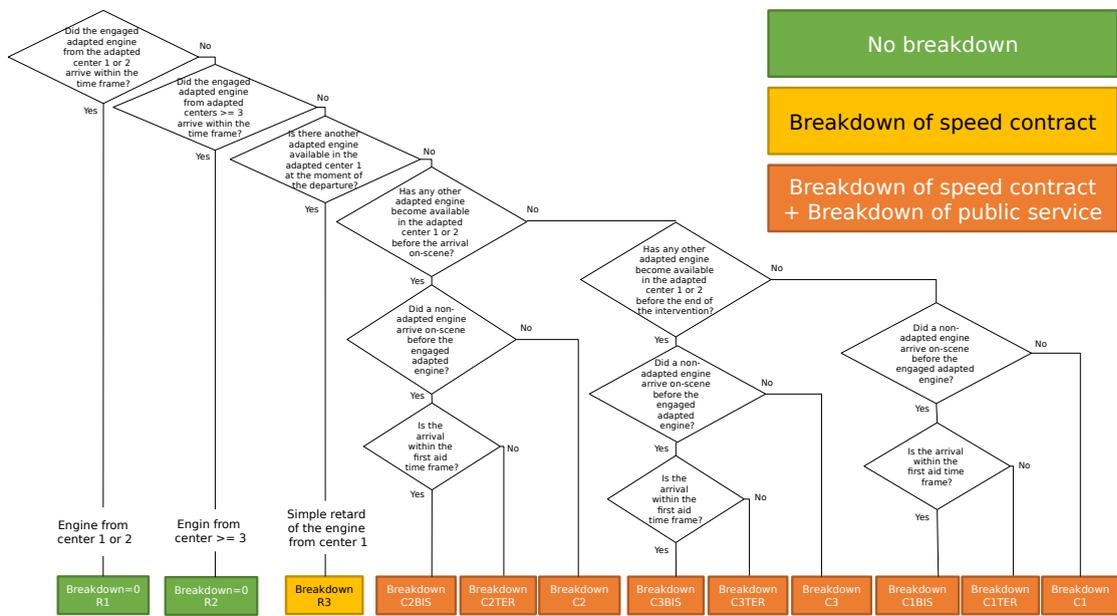


Figure 3.2: The breakage calculation is the indicator of the service quality. In green, we can see the cases that do not generate breaks. In yellow, the cases that generate simple breaks such as arriving late at the scene. In orange, the complex cases, where the engine is late to the scene and there are no other resources available in the two principal centers from the defense list to attend a new simultaneous intervention.

The C2BIS case is when the engaged adapted engine arrives late at the scene (BSC), another adapted engine comes into availability at adapted centers 1 or 2 before the engaged adapted engine arrives at the scene (BPS), and a first aid engine arrives before the adapted engine and within its time frame (no break). However, if the first aid engine does not arrive within its time frame (BSC), it will be a C2TER case. Finally, if there was no first aid engine, it will be a C2 case. This process is repeated for cases C3, C3BIS, and C3TER (Fig. 3.5), with the difference that the other adapted engine that becomes avail-

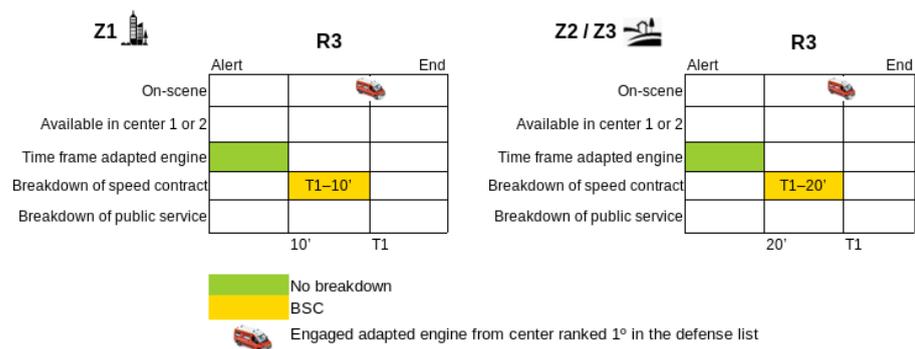


Figure 3.3: Breakdown calculation for case R3. In the image on the left, we present the calculation of the breaking time for the urban zone (Z1) and in the image on the right, for the semi-urban (Z2) and rural (Z3) zones. For both images, the calculation is applied to the engaged adapted engine.

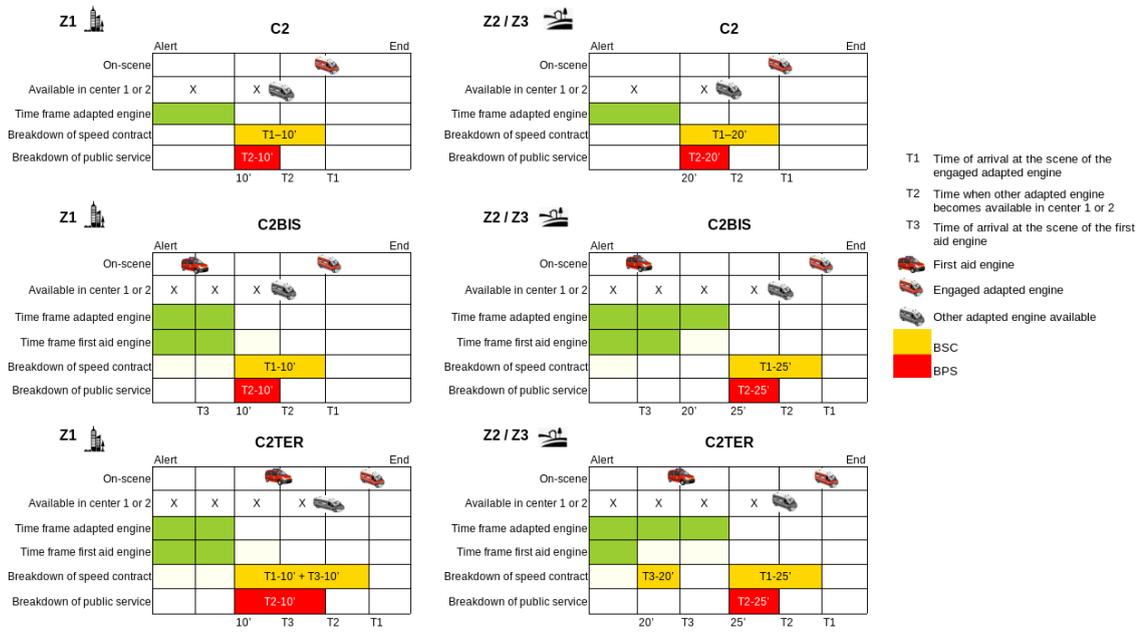


Figure 3.4: Breakdown calculation for cases C2, C2BIS, and C2TER. On the left, the definitions for the urban zone (Z1) and on the right, for the semi-urban (Z2) and rural (Z3) zones. These are the cases when another adapted engine becomes available in centers 1 or 2 of the defense list and before the arrival at the scene of the engaged adapted engine.

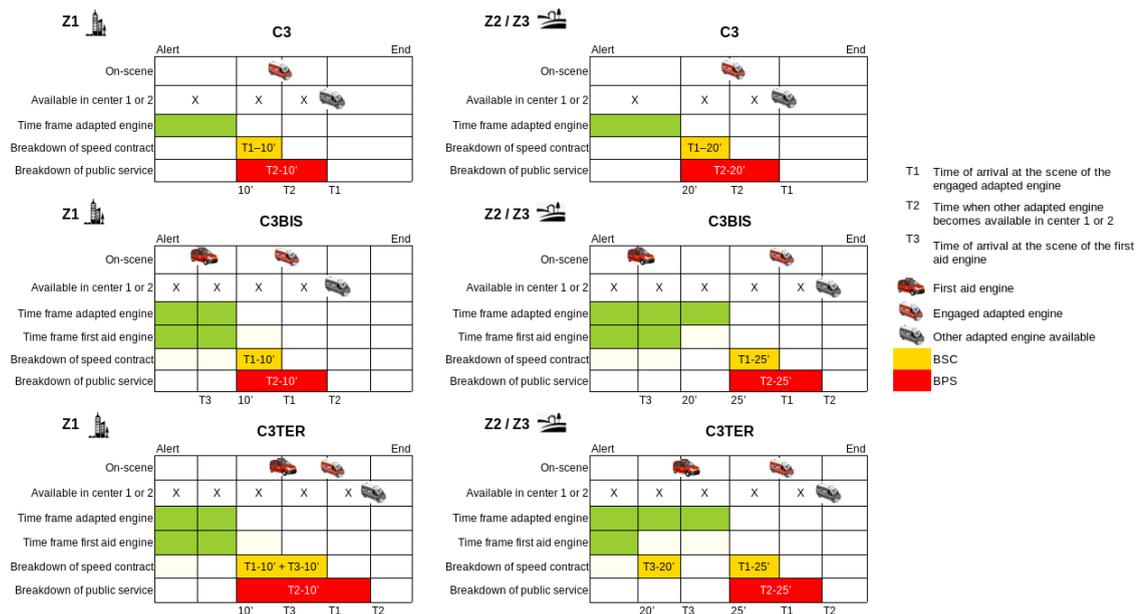


Figure 3.5: Breakdown calculation for cases C3, C3BIS, and C3TER. On the left, the definitions for the urban zone (Z1) and on the right, for the semi-urban (Z2) and rural (Z3) zones. These are the cases when another adapted engine becomes available in centers 1 or 2 of the defense list and before the end of the intervention.



Figure 3.6: Breakdown calculation for cases C1, C1BIS, and C1TER, which are the most critical cases. On the left, the definitions for the urban zone (Z1) and on the right, for the semi-urban (Z2) and rural (Z3) zones. These are cases when there is no other adapted engine available in centers 1 or 2 of the defense list even after the end of the intervention.

able does so before the intervention ends. And when there is no other adapted engine available and no first aid engine arriving before the engaged adapted engine to extend its arrival time, it occurs cases C1, C1BIS, and C1TER (Fig. 3.6).

### 3.4/ CONCLUSION

In this chapter, we have presented the internal data provided by SDIS25 such as the centers, the shapefiles, the deployment plan, the list of interventions, the registered victims, the departures of the engines, the departures of the agents, the composition of the engines, etc. In addition, we have described data from external sources that we have used to expand and enrich our database such as meteorological, traffic, air pollution, epidemiological, hydric, web trends, etc. Finally, based on the knowledge of the internal data, we have explained the operational process carried out by the SDIS25 to respond to an emergency. From the moment an operator answers the call, alerts a center and the latter sends initially an adapted armament to the emergency scene, helps victims, ends its mission and returns to its center for a new intervention. Likewise, we detailed the service breakage calculation modality, designed by the SDIS25, where a break represents the inability to ensure relief to the population. In this case, 2 types of breakdowns and 12 possible cases were identified for the calculation of the breakdown time.





CONTRIBUTION: PREDICTION OF FIRE  
DEPARTMENT INTERVENTIONS



# 4

## PREDICTING THE NUMBER OF INTERVENTIONS

In the previous Chapter, we have reviewed the datasets used and the operational flow of the fire department. In this Chapter, we start to study the predictions of interventions, more precisely, the prediction of the number of interventions in the next hour. For this, we will use the datasets described in Chapter 3 and four ML techniques.

### 4.1/ INTRODUCTION

Most of the problems faced on a day-to-day basis by fire brigades are related to the increase of the number of interventions over time and lack of resources due to insufficient budget. An example of this increase in operational load can be seen in Table 4.1, where the data, provided by SDIS25 and from 2006 to 2017, show the constant growth of the number of interventions over the years (Total Interv.) and the average (Average), standard deviation (Std. Dev.), and maximum number of interventions per hour. These statistics show that the 2017 hourly operational load is more than double the load handled in 2006. This increase and the shortage of personnel and equipment affects the response time to accidents, causing interruptions in the service and deteriorating the quality of response.

Table 4.1: Analysis of the growth of interventions from 2006 to 2017. Data from SDIS25.

Year	Total Interv.	Average	Std. Dev.	Max. Interv.
2006	17,375	1.98	2.04	30
2007	19,368	2.21	2.06	13
2008	18,037	2.05	1.95	16
2009	28,719	3.27	3.34	84
2010	29,656	3.38	3.05	93
2011	33,715	3.84	3.66	48
2012	29,070	3.31	2.50	26
2013	29,830	3.40	2.48	30
2014	30,689	3.50	2.55	22
2015	33,586	3.83	2.68	21
2016	34,434	3.93	3.13	85
2017	37,674	4.30	2.94	22

The advantage of several fire departments is that over the years they have been storing detailed information about the dates and times of the start and end of interventions, transfers of victims to the hospital, departures of their engines and agents, emergency call, etc. This information has been used primarily for monitoring their processes.

Considering this, the objective of the present study was to predict the number of interventions in the next hour by using data recorded from 2006 to 2018 by the fire department, i.e., SDIS25, and external sources (Météo-France, Bison-Futé, Hydro, Sentinelles, etc). This would allow establishing strategies in the organization of resources and reduction of response times.

To identify the best prediction model, initially, we ran experiments comparing two machine learning methods, XGBoost [81] and LSTM. On the one hand, XGBoost is a boosting technique that penalizes model complexity and speeds up processing time due to its fast construction of trees. In [105], the authors proposed an approach based on XGBoost to predict urban fire accidents using ten million samples as data set, an algorithm based on association rules to select features, as well as the Box-Cox transformation to clean outliers. On the other hand, LSTM is a highlighting variation of the Recurrent Neural Network (RNN) and introduced by [17], which has shown a remarkable performance in sequential data applications along with overcoming the vanishing gradient problem presented in the RNN [58, 100].

Then, since we observed that models based on XGBoost showed better performance, we explored some other boosting techniques such as AdaBoost and Gradient Boosting, from Scikit-Learn library, to find the technique that best models our data. The goal of boosting techniques is to solve for net error from the prior tree while adding trees sequentially. The first successful boosting algorithm was AdaBoost, used in researches such as the detection of fire smoke [118] and the forecast of accidents on road traffic by a trichotomy AdaBoost algorithm [143]. Another boosting method that stands out is Gradient Boosting, which tweaks residual errors generated by a previous weak learner in each iteration. In [144], a gradient boosting algorithm was employed to measure the weights of the features of a traffic accident which will then be converted into a grey image and will be the input of a CNN to finally predict the severity of a traffic accident. What is more, in [141], the performance of XGBoost, AdaBoost, Gradient Boosting, and Random Forest for forecasting the future driving risk of crash-involved drivers were compared.

### 4.1.1/ OBJECTIVES

To summarize, the objectives of this work are described in the following:

- a) *Creation of a database*, where internal and external variables from various sources were unified. Internal variables such as the datetime of an intervention and the number of interventions per hour. External variables such as meteorological data, hydrological data, traffic indicators, epidemiological statistics, etc.
- b) *Development and comparison of models based on the LSTM neural network and the XGBoost boosting technique* to predict the number of interventions per hour.
- c) *Development and comparison of XGBoost, AdaBoost, and Gradient Boosting* to predict the number of interventions in the next hour.

## 4.2/ MATERIALS AND METHODS

In this section, we describe the collection process of internal and external variables, their preprocessing, and the modeling process with the techniques LSTM, XGBoost, AdaBoost, and Gradient Boosting.

### 4.2.1/ DATA COLLECTION

We considered two kind of sources: internal and external. The internal contains information about all the interventions recorded from 01/01/2006 to 31/12/2018 by SDIS25. The external contains a set of variables from sources such as Météo-France (tabular data), Hydro (the height of rivers), Atmo (air pollution variables), Sentinelles (epidemiologic statistics), Bison-Futé (traffic indicators), described in detail in Chapter 3, section 3.2. We also added data such as:

- Temporal variables: hour, day, day in the week, day in the year, month and year
- Day and night as binary variable.
- Holidays and vacations as binary variables, where 0 indicates that it is not a holiday or vacations and 1 otherwise.
- Festivities such as Ramadan, Eurockéennes, Percée du Vin Jaune and the FIMU. These were included as binary variables with values 1 for the eve, duration and a day after, and 0 for normal days.

- Distance between Earth and Moon to examine its influence in natural disasters. For this, we used the Skyfield library [138].
- Moon phases as an indicator. For this, we used the Skyfield. library [138].

All the information was organized in a dictionary, where the keys are generated by blocks of one hour. Each hour was related to a set of explanatory variables and a target variable. The explanatory variables were all the ones mentioned previously. The target variable was represented by the number of initiated interventions in one hour. This was identified by extracting the datetime of each intervention.

#### 4.2.2/ DATA ENCODING

The data were transformed into our learning format employing two methods from Scikit-learn library [52]. The *StandardScaler* method was applied to numerical variables such as year, hour, precipitation, temperature, pressure, humidity, dew point, wind direction, wind speed, gust speed, chickenpox, influenza, acute diarrhea statistics, rivers height, and distance between Earth and Moon. This method re-scales the distribution of values to zero mean and unit variance. The *OneHotEncoder* method was employed to convert into indicators categorical variables such as Bison Futé's values, barometric trend, day, day of the week, day of the year, month, holidays, vacations, festivities, and moon phases. The original target values were kept (the number of interventions) because the distribution of the interventions count is better represented by discrete values.

The organization of each sample consisted of joining the extracted features with the number of interventions of the previous 169 hours (1 week plus 1h). Eventually, the data set is considered as sequential data and converted to supervised learning, i.e., the target is the number of interventions in the next hour ( $t + 1$ ) of a present sample ( $t$ ).

#### 4.2.3/ MODELING

For the first experiment, where we compared XGBoost and LSTM, the available dataset was from 2006 to 2017 and did not contain air pollution variables from Atmo. The metrics defined to evaluate the results were RMSE and MAE. Furthermore, since the events to be predicted are countable, the accuracy score was also considered with a margin of error zero (ACC0E) that represents the number of exact predictions reached, with a margin of error less or equal to one (ACC1E) and to two (ACC2E). In order to discover unusual years through the analysis of the predictions, each year was predicted, i.e., it was considered as testing set, the remaining years were used as training and validation sets (e.g., to predict 2006, 2007-2017 were used as learning sets; to predict 2007, 2006

and 2008-2017 were used as learning set, etc). Naturally, this was not a real case, but it provided us with information on how well each year can be predicted and why some years presented atypical results. The data set was not cleaned from possible outliers, such as natural disasters (e.g., storms, fires, floods) and strikes that were found in our search analysis. Considering that in real-world applications, the system must perform well in such conditions it is worth maintaining these occurrences and evaluates the performances of the proposed methods.

Our XGBoost model was improved using the GridSearchCV procedure from the Scikit-learn library [52]. The best model used in this research had a `max_depth` of 3, a `learning_rate` of 0.1, the learning task was *Count* and the learning objective was *Poisson* (for data counting problems), and the remaining parameters were kept as default. Our LSTM model was developed with Keras library [74]. It was built with one LSTM layer and 6000 neurons, one dense layer with one neuron as output and a last layer with the LeakyReLU activation function, considering 0.1 in the negative slope coefficient. The time step was one per input. For the training phase, the Stochastic Gradient Descent optimizer was used with a learning rate of 0.01, momentum and decay values of 0.0001, *Poisson* as loss function, a batch size of 64 and 200 epochs with an “EarlyStopping” of 10 epochs to monitor the loss function decrease of the validation set.

For the second experiment, where we compared AdaBoost, Gradient Boosting, and XGBoost, we extended the previous dataset from 2006 to 2018 and added air pollution variables such as PM2.5 fine particles, PM10 fine particles, ozone, and nitrogen oxide from Atmo. Since the dataset was increased, we again search for the best configuration for XGBoost. To evaluate the performance of the three methods, the two last years (2017 and 2018) were selected as the testing set independently, i.e., the data from 2006-2016 and the data from 2006-2017 were used as training sets to predict years 2017 and 2018 (testing sets), respectively. The metrics used to evaluate the performance of the models were the same ones used in the first experiment and one more added called *Tsec*, which is the prediction execution time in seconds. The three techniques were tuned via the GridSearchCV procedure from the Scikit-learn library [52].

## 4.3/ RESULTS

To run both experiments, a machine with a Titan X, Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz with 64GB of RAM and with a GPU of 3072 cores and 12GB of RAM was used.

For the first experiment, Table 4.2 presents the results of the forecast to both LSTM and XGBoost models for all years (2006-2017). Fig. 4.1, Fig. 4.2 and Fig. 4.3 illustrates outcomes of LSTM and XGBoost models on 100 samples trying to predict an unusual

number of interventions occurred in 2010, 2011 and 2016 as a result of natural disasters in the Franche-Comté region. Fig. 4.4 presents the prediction results for 100 samples in the year 2017, which is the year that only considers past years in the training process and presents an uncommon behavior due to ambulance strikes and climate conditions. Lastly, taking into account that LSTM and the XGBoost models predict real values (e.g., 5.67 interventions), results were transformed to the closest integer (e.g., 6 interventions) for being coherent with real life applications.

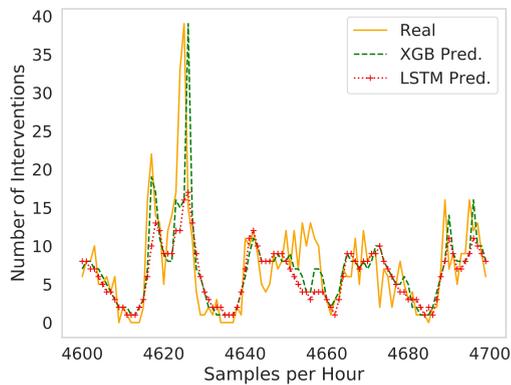


Figure 4.1: Predictions for 2010.

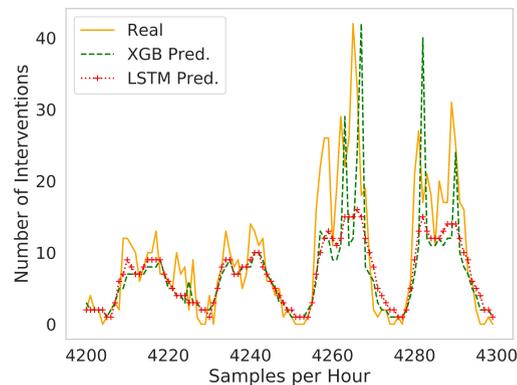


Figure 4.2: Predictions for 2011.

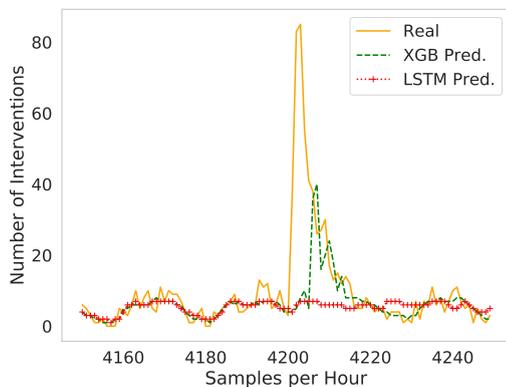


Figure 4.3: Predictions for 2016.

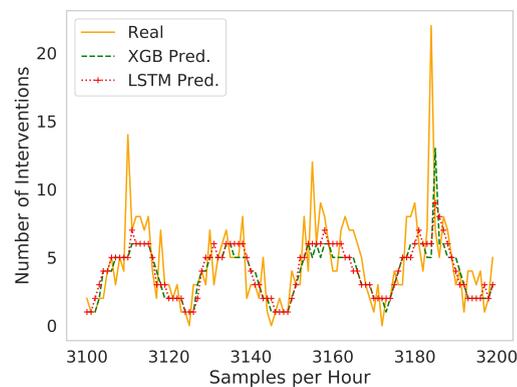


Figure 4.4: Predictions for 2017.

Table 4.2: Prediction results on data 2006-2017

Year	LSTM			XGBoost						
	RMSE	MAE	ACC0E (%)	ACC1E (%)	ACC2E (%)	RMSE	MAE	ACC0E (%)	ACC1E (%)	ACC2E (%)
2006	1.60	1.13	28.28	73.04	90.99	1.61	1.16	25.55	73.27	90.86
2007	1.63	1.19	27.27	70.91	89.06	1.66	1.20	26.19	71.48	88.83
2008	1.59	1.16	26.83	71.68	90.28	1.64	1.22	24.45	69.94	89.55
2009	2.28	1.49	22.72	62.28	83.00	2.39	1.58	21.59	59.04	80.36
2010	2.32	1.49	23.17	61.96	81.92	2.22	1.51	22.65	60.82	81.50
2011	2.49	1.68	21.05	57.54	78.92	2.55	1.69	21.07	58.16	78.93
2012	2.06	1.53	21.30	58.26	81.11	2.08	1.55	21.16	58.03	80.02
2013	2.05	1.53	21.15	58.81	80.58	2.06	1.54	20.91	58.68	80.22
2014	2.04	1.52	21.26	59.10	81.17	2.06	1.52	21.47	59.37	81.00
2015	2.09	1.58	21.14	56.41	79.49	2.09	1.56	21.51	57.70	79.48
2016	2.64	1.71	18.94	53.91	77.51	2.58	1.67	19.16	55.49	78.42
2017	2.26	1.69	19.90	54.63	76.80	2.27	1.68	20.38	55.59	76.94

In Figs. 4.1-4.3, we could see that the XGBoost technique was more robust to outlier data than LSTM. The former was better recognizing peak occurrences during natural disasters.

Taking into consideration that these occurrences are highly likely to happen in the future and fire brigades pursue to nurse its community better, real systems must be prepared to identify these uncommon values.

In Table 4.2, we observed that for the years 2009-2011 and 2016 the RMSE metric presented high values, probably affected by the outlier data. In the case of 2012-2015, the RMSE increment followed a normal pattern compare to the increase of the total number of interventions. Finally, the year 2017 and its predictions in Fig. 4.4, which is the most realistic predictions, showed a lower ACC2E accuracy. This is likely due to the typical growth of the number of interventions over the years and to the exceptional cases, in which for longer periods there was a high operating load. In our analysis of factors, we found that there was an ambulance strike that lasted 29 days between September and October, which meant higher incident attendance for firefighters. We also found online sources related to a 60% increase in the number of operations for the Doubs department caused by a heat wave that occurred in June [197].

For the second experiment, Table 4.3 presents the hyperparameters found and used for each method to achieve its respective better results. Table 4.4 presents the metrics results for each method to both 2017 and 2018 years, where the numbers in bold indicate the overall best result. Furthermore, Figures 4.5 and 4.7 illustrate forecasting results for all three methods, zooming in 175 samples; and Figures 4.6 and 4.8 exhibit in a bar plot the comparison to the exact number of predictions with errors from 0 to 14.

As can be seen in Table 4.3, unlike the other three methods, the best AdaBoost model used fewer estimators and a greater tree depth, which allowed to reach a greater number of exact predictions (ACC0E). However, for both years, AdaBoost model was the method that consumed the largest time in prediction with 0.91 seconds and higher MAE and RMSE metrics as shown in Table 4.4, which implies that high numbers of interventions were not well recognized by the method, this is shown in Figures 4.5 and 4.7.

Table 4.3: Hyperparameters

Hyperparameter	AdaBoost	Gradient Boosting	XGBoost
Estimators	50	100	200
Maximum tree depth	7	5	5
Learning rate	0.05	0.04	0.02
Instance sampling	1.0	0.8	1.0
Features sampling	0.8	1.0	0.9
Loss/Objective	linear	least squares regression	count:poisson

Table 4.4: Prediction results for 2017 and 2018

Year	Technique	TSec	MAE	RMSE	ACC0E (%)	ACC1E (%)	ACC2E (%)
2017	AdaBoost	0.91	1.71	2.35	20.88	55.79	75.38
	GradientB.	0.16	1.69	2.29	20.55	55.66	76.85
	XGBoost	0.24	<b>1.68</b>	<b>2.28</b>	20.76	<b>56.47</b>	<b>76.91</b>
2018	AdaBoost	0.91	1.85	2.60	19.20	53.04	74.02
	GradientB.	0.06	1.81	2.50	18.78	52.68	74.98
	XGBoost	0.24	<b>1.80</b>	<b>2.50</b>	19.02	<b>53.08</b>	<b>75.34</b>

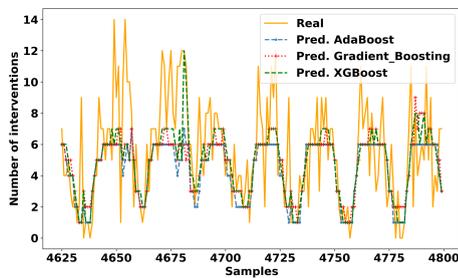


Figure 4.5: Predictions for 1h - 2017

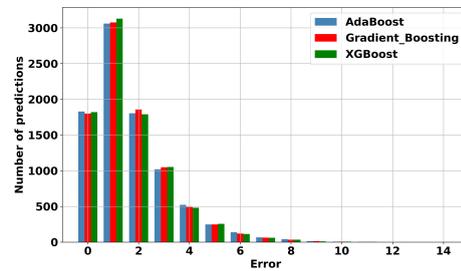


Figure 4.6: Exact predictions for 1h - 2017

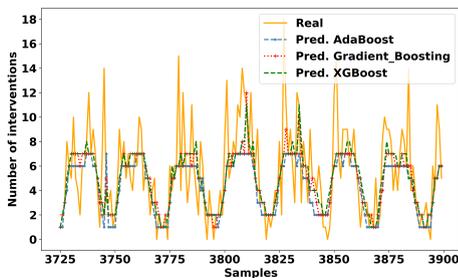


Figure 4.7: Predictions for 1h - 2018

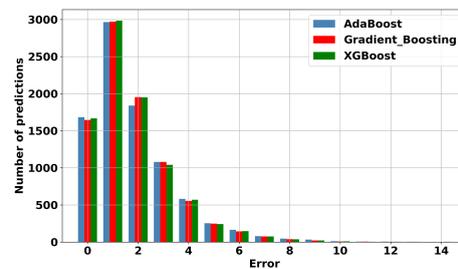


Figure 4.8: Exact predictions for 1h - 2018

Gradient Boosting took a sample of 80% of the instances (Instance sampling = 0.8) and analyzed all the features (Feature sampling = 1.0). As shown in Figures 4.5 and 4.7, the best Gradient Boosting model reached more peaks than AdaBoost. What is more, from the three methods, it was the one that consumed less time (TSec) when performing the prediction for both years. Also, in Figures 4.6 and 4.8, Gradient Boosting presented a larger number of predictions with a margin of error two (ACC2E), i.e., for 2017, it reached 1856 exact predictions, while AdaBoost and XGBoost models obtained 1804 and 1790 respectively; whereas for 2018, AdaBoost achieved 1839 and XGBoost 1950 exact predictions, being surpassed by Gradient Boosting with a value of 1953 exact estimates.

However, from the three methods, XGBoost presented a better performance in recognizing the peaks of interventions over time. This is clearly visualized in the predictions for 2018, in Fig. 4.7, where it reached a peak of 12 interventions. Also, for 2017, XGBoost obtained almost the same results that in experiment 1 but with a different configuration (a low learning rate and a high number of estimators). For both years, the metric ACC2E of the best XGBoost model stood out among the other models.

Furthermore, it can be noticed in Table 4.4 that forecasting results for 2018 decreased performance comparing to 2017. Such result can be explained by the fact that in 2017 the total number of interventions was 37 674 with 4.30 interventions per hour on average and standard deviation of 2.94, while for 2018 the total number of interventions increased to 40 957 with 4.68 interventions per hour on average and standard deviation of 3.23.

Finally, during the development of this experiment, we noticed that the meteorological

and temporal variables showed a greater impact on the models, especially when there were peaks in the occurrence of interventions due to natural phenomena. In addition, it is clear that the temporal variables such as the identification of the day or months with more holidays allow us to recognize that there will be a prolonged increase in the operational load.

#### 4.4/ DISCUSSION AND RELATED WORK

When we reviewed the literature at the time of these experiments, we discovered that there was little research in the specific field of predicting the number, type, and location of interventions for fire departments and based on ML techniques. Some related works were the prediction of the number of interventions using multi-layer perceptron [127], the prediction of the number and nature of firefighting operations [123], the prediction of the short-term traffic speed and flow [103, 97, 109, 143, 144], the detection of fire smoke [118], the prediction of urban fire accidents [105], the prediction of driving risk of crash-involved drivers [141], a survey on the analysis of eight LSTM variants on three tasks: speech recognition, handwriting recognition, and polyphonic music modeling [100], and a hybrid deep learning framework with XGBoost for accident prediction [105].

Furthermore, in 2019, the study [122], also developed by the author of this manuscript, built models for the prediction of interventions and for two types of time horizons (1h and 3h), the technique used was LSTM and achieved an accuracy above 70%. This allowed us to further research and compare other techniques such as the experiments conducted in this chapter. With the comparison of LSTM and XGBoost, we could evaluate the performance of two differently based techniques. LSTM represented the best version of a recurrent neural network, while XGBoost represented the best implementation of a scalable, distributed gradient-boosted decision tree. Thus, we identified that XGBoost recognized better the patterns of our data and therefore, we explored more the boosting methods, testing AdaBoost and Gradient Boosting. As presented in Table 4.4 and considering that we added more data (air pollution variables and interventions from 2018), it can be seen that XGBoost continue to outperform the other techniques with good outlier recognition.

Finally, the present work allowed the identification of the techniques with the best performance, the recognition of variables with the greatest impact, and the generation of three published articles [151, 150, 146]. What is more, in recent years, the company *SAD Marketing* developed the *PredictOps* system [208], which aims to predict the future volume of intervention by geographical area, allowing the optimization of the operational response.

## 4.5/ CONCLUSION

The purpose of this Chapter was to explore the efficacy of different machine learning techniques and future implementation of the one who better fits our data in a real data-driven system. For this, we compared 4 techniques: LSTM, XGBoost, AdaBoost, and Gradient Boosting. In addition, we built a data dictionary with explanatory variables from various sources. The main or internal source was the interventions recorded by the SDIS25, which represents the operational load. And external sources such as Meteo-France (meteorological data), Bison Futé (vehicular traffic data), Atmo (statistics of environmental pollution particles), Hydro (statistics of the height of water in the rivers of the region), Sentinelles (epidemiological statistics), etc.

The evaluation of the performance of the techniques and the impact of the variables allowed an expanded view of the effects of machine learning techniques in solving our problem. Our results were published in [151, 150, 146]. What is more, these studies were the precedent for the construction of PredictOps [208], a system designed for predicting the number of interventions by area, deployed in production by the company SAD Marketing.

## FORECASTING AMBULANCES' TURNAROUND TIME IN HOSPITALS

In the previous chapter, we have developed models based on machine learning for the prediction of the number of interventions, considering variables from various sources (e.g., temporal variables, meteorological records, traffic indicators, environmental pollution variables, etc). In this chapter, we will develop methodologies for the prediction of the ambulance's turnaround times during the first COVID-19 quarantine, in France.

### 5.1/ INTRODUCTION

It is well known that the Emergency Department (ED) is one of the most crowded departments in hospitals [56, 76, 37, 33, 68], where people seek immediate attention given the severity of their problem and long lines are generated contributing to patient dissatisfaction. What is more, EDs' overcrowding can have a knock-on effect on other external services, such as ambulances. Theoretically, when an ambulance arrives at the hospital with patient(s), the crew transfers the patient's care to the ED staff, completes the necessary reports, and cleans and restocks the ambulance (e.g., replaces stretcher linens) to return on service. This total time an ambulance spends at the hospital for handing over patient(s) is referred to as *turnaround interval* [18], which we interchangeably refer to as the turnaround time (TT) of ambulances throughout this chapter.

However, with EDs' overcrowding, there is a problem referred to as ambulance offload delay (AOD), which occurs when ambulances' patients cannot be transferred for immediate care to hospitals' ED [56, 116]. On the one hand, AOD risks patients' life due to delays to receive adequate treatment and/or diagnosis, for example. Besides, AOD affects the emergency medical system (EMS) as their ambulance and staff will be in use for more time. That is, AOD directly increases the ambulances' TT, which poses a population at risk if other major incidents occur and they cannot attend to them [30, 76, 33, 88].

Fire departments are a key component of civil security that ensures the well-being of the population. In some countries such as France, they are also responsible for emergency medical services and have been facing a continuous increase in the number of interventions over the years, which represents a need for the acquisition of more resources and the optimal reorganization of them [150, 151, 146]. In this study, we analyzed the case of SDIS25. At the time of this study, SDIS25 had 71 centers deployed throughout the Doubs region. The region was divided in 573 cities and 440 districts. Its main cities were: Besançon, the capital, where the nearest hospital was Besançon Regional University Hospital Center (CHRU), Montbéliard in the north, near the North Franche-Comté Hospital (HNFC), and Pontarlier in the south, near the Intercommunal Hospital Center of Haute-Comté (CHIH).

As mentioned in Chapter 3, SDIS25 is equipped with specialized engines for each type of intervention and the firefighters have various skills in operating these engines. If there are not enough resources available for a long period to attend to an intervention, either because of the lack of firefighters, the lack of engines or both, a service breakdown occurs, i.e., the inability to assist within the time limits, which puts the safety of a certain area or population at risk. For instance, over the years, it has been observed that most of the breakdowns have occurred in July, since more people go on vacation during this time, including firefighters, i.e., there could be an increase in the number of interventions and with reduced personnel more breakdowns would be generated [152]. If we add to this the impact of a natural phenomenon, epidemic or pandemic, centers would be more affected, as well as their resilience if resources are scarce.

On 12th January 2020, the World Health Organization (WHO) announced [218] a novel Coronavirus [174, 171, 165, 159, 179, 185], which was officially named as *COVID-19* (coronavirus disease 2019) [217] on the 11th of February. Further, with the spread of COVID-19 across the world, it was declared as a pandemic [219] on 11th March 2020. In general, multiple measures have been reported for dealing with this novel Coronavirus across the world [154, 175, 145, 157]. In France, as stated in [172], strengthened surveillance of COVID-19 cases was implemented on 10th January 2020. In their paper, the authors carefully described the real-time surveillance system for the first three cases, which were detected on 24th January 2020. For the scope of this study, the first official cases in the Doubs region have been reported on 18th March 2020 as shown in daily statistics published by Santé Publique France in [211]. Nevertheless, the SDIS25 early started to attend interventions in which patients had symptoms of the disease by 29th February 2020.

Although numerous measures were taken to secure peoples' health and well-being, the COVID-19 pandemic has proven to be quite a challenge, particularly to the public health sector [148, 156, 157]. For instance, in several parts of the world, numerous news re-

ported an increment to the ambulances' waiting time in hospitals. In the Texas county US, there are reports of ambulances waiting about 4-12 hours as hospitals are crowded with coronavirus cases [204]. There is also a report in France, in which ambulances had to wait about 3 hours in a hospital [205]. This increment could lead sometimes to the worst-case scenario, i.e, patients' death, as every minute count in such a situation. This is a case, which was reported in Australia, where two patients have died for long waiting time in ambulances outside hospitals [200].

### 5.1.1/ DESCRIPTION OF THE OPERATIONAL PROCESS

As described in Chapter 3, section 3.3, the operational process flow of SDIS25 considers several recorded times. Here, we will briefly explain that process focusing on the turnaround time.

1. First, an emergency call is received and the required armament is gathered to go to the scene.
2. Once arrived at the scene, if necessary, the victim is taken to the hospital.
3. At the hospital, the firefighters wait to transfer the victim to the emergency room and to do the corresponding administrative process (TT).
4. Finally, they return to their center and are available again to attend other interventions.

Each performed step requires a record of the firefighter's status, which is done manually by pressing a button. The firefighter crew chief makes a record to indicate that they are going to the hospital, another record to indicate when they arrive at the hospital (start of TT), and another one to indicate that they are leaving it (end of TT). This is illustrated in Fig. 3.1. However, there is the possibility of human error, where the firefighter may have forgotten to record the status and registered it a long time later, or on the contrary, where the firefighter may have accidentally recorded too soon.

In addition, an intervention can have several ambulances departing at different times from the scene, depending on the number of victims, or either, sometimes in the same hour we can have several simultaneous interventions but of different types. If firefighters spend more time in hospitals, i.e., high TT, there will be fewer resources available at the centers to respond immediately to an incident.

### 5.1.2/ OBJECTIVES

With these elements in mind, the present work proposes a novel methodology based on Machine Learning to make predictions for the TT of each ambulance in a given time and hospital, aiming to provide a decision support tool for SDIS25 and EMSs, in general. In this way, EMSs can activate various proactive mediations according to the time their personal and resources will be in use, aiming to mitigate service breakdown and, consequently, being able to save more lives. In other words, for the short-term, such predictions could allow better allocation of the remaining and available resources if known a priori the time each ambulance will spend in hospitals. For the medium- to long-term analysis, such a system can be improved to re-calculate a better hospital option considering the predicted TT. To summarize, our objectives in this work are described in the following.

- a) *Analysis of the COVID-19 impact on ambulances' TT and the breakdowns generated in the fire service.* We provide an in-depth analysis of the COVID-19 impact on the average turnaround time (AvTT) per day in each hospital during the first semester of 2020 in the Doubs region. Further, we describe the already existing breakdown problems in the fire service due to long TTs in hospitals. Finally, we show how this problem worsens with the arrival of COVID-19, demonstrating the need for a system to prevent breakdowns in the fire service in the face of a pandemic.
- b) *Creation of a regularly spaced time series model that represents the average hourly turnaround time at each hospital.* The model is based on the history of TTs reported by SDIS25 ambulances since 2015 for CHRU and CHIH hospitals, and since 2017 for HNFC, in order to recognize hourly, daily, and weekly trends. To this was added internal explanatory variables such as the number of interventions by firefighters (a greater increase in interventions in a certain period of time may indicate a greater number of victims attending the hospital) and the suspected cases of COVID-19 registered by the SDIS25. External variables such as the number of COVID-19 cases officially reported in the region; keywords most searched in Google to retrieve trends; traffic predictions, since a very congested day can generate accidents and victims to transport; and meteorological data were also considered, since many floods usually occur in the region, which generates material damage and victims. Four state-of-the-art ML models were considered in this task, i.e., a decision-tree based model (LGBM), a feedforward type neural network (MLP), a recurrent neural network (LSTM), and a decomposable time series model (Prophet).

- c) *Creation of a irregularly spaced time series model to predict the turnaround time for each intervention and its ambulances.* One or more ambulances belong to an intervention, if the intervention has a long duration, the ambulances could go to the hospital at different times of the day. It is here where the first model described in (b) will provide us with predicted information on the AvTT in a given hour as an explanatory variable. Along with the type of intervention (to recognize the degree of severity of the incident) and other variables mentioned previously we propose to forecast the TT for each ambulance. Because in the previous task (b) the LGBM model consistently and considerably outperformed the other ML models, for this task of predicting each ambulance's TT (not average TT per hour), we performed experiments with only LGBM.

## 5.2/ MATERIALS AND METHODS

In this section, we describe the collection process of internal and external variables. We analyze the recorded ambulances' TT through the years and the impact of COVID-19 on ambulances' TT and, consequently, the impact of COVID-19 and high TTs on the fire service. Further, we present our proposed methodology for forecasting the TT of each ambulance and AvTT per hour, and the metrics used to evaluate our models.

### 5.2.1/ DATA COLLECTION

The main source of data comes from the Departmental Fire and Rescue Service of the Doubs department, France. It contains a history of ambulances arrivals collected from January 2015 to June 2020 for CHRU and CHIH hospitals, and from February 2017 to June 2020 for HNFC. For each hospital, two types of datasets were created:

- Arrivals dataset (Arr-DS). The samples represent the TT of each ambulance in a hospital at any time of the day. The features are: year, month, day, day of the week, day of the year, and hour. Also, an indicator to recognize if it is the beginning or end of the week, month, and year. In addition, we added the type of intervention, which is described in 3.1 as three variables: the type of operation, the subtype of operation, and the departure's motive. Finally, the TT in minutes (min) recorded was included. In some records, we found TTs less than 5 minutes, which is very rare, and it is possible that they were caused by a bad manual recording. Therefore, if the TT is less than 5 minutes, we clipped it at 5. Similarly, there were five unusual samples with TT between 4-7 hours, which were clipped at 90 minutes, since most TTs (99.7%) had less than or equal to 90 minutes.

- Average turnaround time dataset (AvTT-DS). This dataset has been organized by hour, where each hour represents the average TT of all ambulances that arrived at the hospital in that hour. For example, if between 15h and 16h there were 3 ambulances with TTs: 20 min, 15 min, and 30 min, the mean was 21.67 min for 15h. In addition, each sample includes the following features: year, month, day, day of the week, day of the year, hour, an indicator to recognize if it is the beginning or end of the week, month, and year. And, since 90% of all hospital arrivals have had an AvTT between 10 and 60 minutes, we set the average minimum and maximum with these values. The hours in which there were no arrivals and, therefore, there was no recorded TT and AvTT, were completed by linear interpolation since the TT over the years has been constantly increasing.

In order to discover possible influential external variables, we built another dataset (External-DS) by hour from “01/01/2015 00:00:00” to “30/06/2020 23:00:00”, where we incorporated variables from Google Trends, Bison-Futé, Météo-France, and Data Gouv, as described in 3.2. Furthermore, we included other variables from the primary source, they were the total number of interventions recorded in a given hour over the 6 years and the number of cases attended with a suspect of COVID-19 per hour, from February 2020 to June 2020. This last variable was completed with zeros before February 2020.

### 5.2.2/ DATA ANALYSIS

The dataset at our disposal has 78,777 interventions where firefighters transported victims to one of the three aforementioned hospitals. The frequency on the number of times each hospital received a victim is 55.44% (CHRU), 18.27% (CHIH), and 26.29% (HNFC), respectively. In order to understand how the ambulances' TT are distributed in our dataset, Fig. 5.1 illustrates, for each hospital, a histogram with bins of 1 minute and the cumulative number of TT in hours (y-axis) for each day of the week and hour in the day (x-axis).

In the first column of Fig. 5.1, CHRU, CHIH, and HNFC have right-skewed distributions with mean and standard deviation (std) values as  $18.43 \pm 10.68$ ,  $14.91 \pm 8.66$ , and  $22.61 \pm 11.26$ , respectively. In the second column of Fig. 5.1, one can notice a similar pattern for the cumulative sum of TT in hours that firefighters' ambulances spend in the three hospitals with different peak values, which depends on the frequency of times each hospital received victims. This pattern is also noticed in the works [53, 76]. In our case, from 8h in the morning on, the TT starts to increase and remains high up to 19h when it starts to decrease. Also, between 0-6h, the highest cumulative TT is during the weekend, i.e., Friday, Saturday, and Sunday. This is because at weekends people tend to go out more and until the wee hours of the morning, which might lead to more accidents, more patient visits to hospitals during on-call hours with reduced staff, and results in a slight

increase in ambulances' TT.

In addition, as stated in the introduction (Subsection 5.1.2), this work aims to study the COVID-19 impact on ambulances' TT in the three aforementioned hospitals. First, Table 5.1 exhibits for each semester (Sem.) of the analyzed years (2015-2020) and hospital, the following statistics: the total number of arrivals (Arr.) and the mean $\pm$ std TT values in minutes.

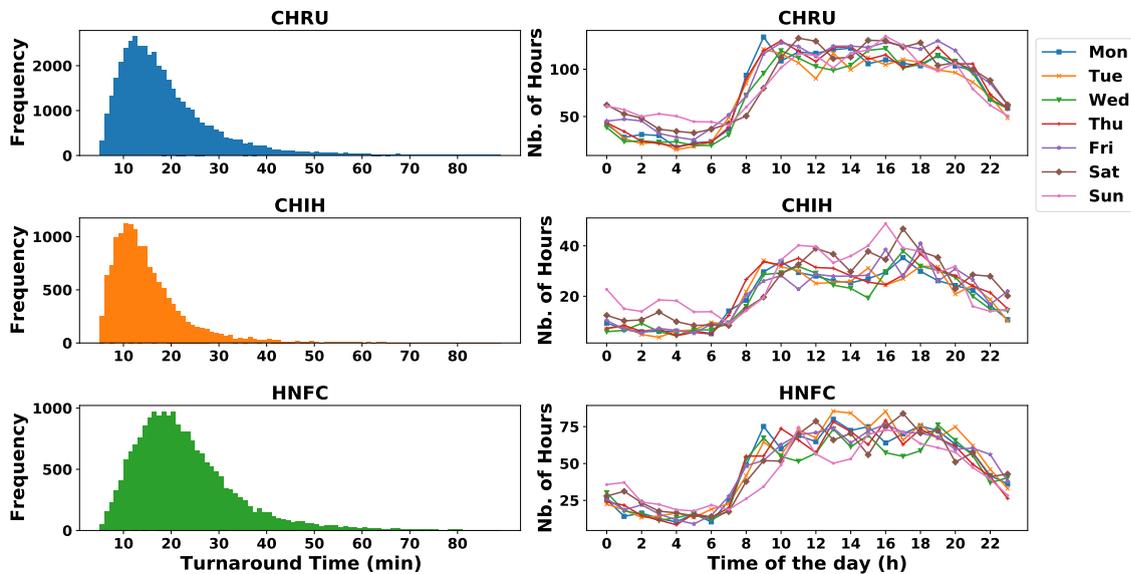


Figure 5.1: Statistical analysis for the TT of ambulances: each row illustrates a histogram with bins of 1 minute to the TT frequency distribution (left side) and the cumulative sum of TT in hours for each day of the week and hour in the day (right side), considering each hospital. On the left side, the frequencies for the three hospitals follow a positively skewed distribution, where more than 80% of cases were less than one hour. On the right side, we can deduce that between 8h and 22h the accumulated time follows the intense human activity while awake, which is translated as more workload and longer TTs.

As one can notice in Table 5.1, CHIH has fewer arrivals and a shorter AvTT per semester whereas CHRU presents more arrivals than the other two hospitals since it is located in the capital of the territory, and a lower AvTT than HNFC. Besides, when looking at the number of arrivals for the first semester, it can be detected that there was a higher workload in the firemen department for the years 2018 and 2019 compared to the other years. In the case of 2020, the reduced workload was due to a lockdown period [215], which decreased the movement of people and reduced the number of firefighters' interventions, for example, fewer traffic accidents. The second semester of each year normally presents a higher AvTT, which could be due to two holiday periods (Jul-Aug and Dec), where people travel more due to vacations. Although the AvTT normally increased for each hospital throughout the years according to its workload, the AvTT for 2020 is higher than all years and for the 3 hospitals, even with a reduced workload compared to 2018 and 2019. For instance, hospital CHRU presented about 9.13% more AvTT in the first semester of 2020

than in the first semester of 2019, which is 9 times higher comparing the increment for the same period for the years 2019 and 2018 (1%). Similarly, hospital HNFC presented about 6.54% more AvTT in the first semester of 2020 than in the first semester of 2019, while the years 2017-2019 presented similar AvTTs. This proves the impact of the pandemic on the increase in ambulances' TT.

Table 5.1: Data analysis, for each hospital, on the number of arrivals (Arr.) and Mean $\pm$ std TT values in minutes per semester during 2015-2020.

Sem.	Year	CHRU		CHIH		HNFC	
		Arr.	Mean $\pm$ std TT	Arr.	Mean $\pm$ std TT	Arr.	Mean $\pm$ std TT
1st	2015	3,114	17.72 $\pm$ 10.24	1,230	14.50 $\pm$ 7.84	-	-
	2016	3,125	17.68 $\pm$ 10.36	1,176	14.33 $\pm$ 7.64	-	-
	2017	3,676	17.55 $\pm$ 10.31	1,288	13.99 $\pm$ 7.43	2,272	22.97 $\pm$ 11.12
	2018	5,233	18.75 $\pm$ 11.37	1,548	15.05 $\pm$ 9.41	3,824	22.91 $\pm$ 12.18
	2019	4,672	18.94 $\pm$ 11.66	1,368	15.63 $\pm$ 9.80	2,994	22.92 $\pm$ 11.90
	2020	3,899	20.67 $\pm$ 12.57	1,184	15.96 $\pm$ 10.51	2,136	24.42 $\pm$ 13.19
2nd	2015	2,893	17.59 $\pm$ 10.28	1,174	14.69 $\pm$ 8.91	-	-
	2016	3,320	17.77 $\pm$ 10.32	1,222	13.88 $\pm$ 7.32	-	-
	2017	4,829	18.93 $\pm$ 11.76	1,556	15.04 $\pm$ 9.41	3,454	22.07 $\pm$ 10.90
	2018	5,182	18.56 $\pm$ 11.35	1,440	15.50 $\pm$ 9.32	3,600	22.04 $\pm$ 11.55
	2019	3,730	20.08 $\pm$ 13.03	1,208	16.52 $\pm$ 11.23	2,430	23.25 $\pm$ 12.37

Furthermore, we now try to comprehend how the on-going pandemic has made an impact on the ambulances' AvTT per day on hospitals for 2020 only. For this purpose, the plot on the top of Fig. 5.2 illustrates for each day (x-axis) the AvTT in minutes (y-axis) for each hospital. Additionally, at the bottom of this figure, the plot illustrates the official COVID-19 statistics from two data sources, i.e., Data Gouv and the ones reported by the SDIS25, to search for evident patterns. More specifically, this plot illustrates the current number of hospitalized individuals (hosp.), the number of patients in reanimation or critical care (rean.), the cumulative number of individuals who returned home (ret. home), and the cumulative number of individuals who died (dead) regarding the Data Gouv source, and the number of cases per day attended by the SDIS25 with a suspicion of COVID-19 (Susp. COVID-19). For both data sources, there is an indication of the first day with COVID-19 cases.

In Fig. 5.2, while the first day with suspicious cases of COVID-19 is 29/02/2020 reported by the SDIS25, the date with official cases is almost three weeks later, on 18/03/2020. However, one can notice that as soon as the SDIS25 starts to take a significant number of people with the symptoms of the disease to hospitals in March (after 10/03/2020), the AvTT per day starts to increase for CHRU and HNFC. This increment is more remarkable, for the three hospitals, after the first official cases have been reported. Furthermore, the peak period of hospitalized cases occurred approximately from the middle to the end of April. In such conditions, there were more hospitalized patients in which some of them started to need more and even intensive care as the number of cases in reanimation indicates. When comparing the AvTT distributions of the 3 hospitals and the number of discharged cases (Ret. home), we find an inverse correlation, given that as hospital re-

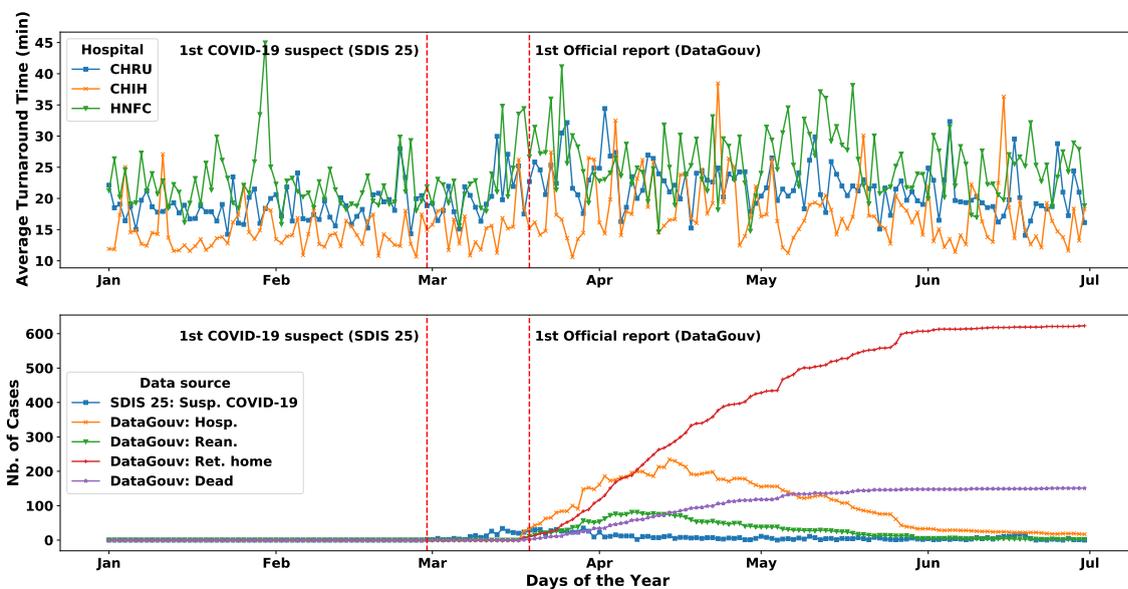


Figure 5.2: Statistical analysis for the AvTT per day considering the COVID-19 impact. The plot on the top illustrates statistics on the average turnaround time in minutes per day for 2020 and each hospital. The plot on the bottom illustrates the COVID-19 related statistics provided by Data Gouv and the SDIS25.

sources are freed up, AvTTs decrease. The opposite case can be found when comparing the AvTT of the 3 hospitals and the number of hospitalized cases (Hosp.), where the 4 distributions follow a similar pattern, indicating that there is a positive correlation between the variables, i.e., as the number of hospitalized patients increased, the hospitals became saturated and their response time to receive new patients was affected. This generated larger TTs for the ambulances, leaving them unavailable to attend other interventions and limiting the resources of the fire brigade. In short, a chain effect on the availability of resources.

To complement Fig. 5.2, Table 5.2 exhibits the AvTT per hospital and month for 2020 highlighting in italic font the highest value per hospital. Comparing the highest AvTT and the lowest one per hospital, considerable increments were noticed. For instance, hospital CHRU presented about 24.96% more AvTT in April than in January, hospital CHIH present about 42.68% more AvTT in April than in February, and hospital HNFC presented about 29.13% more AvTT in May than in February. In general, for each hospital, the period pre-pandemic (January and February) presented low AvTT while months March, April, and May (during-peak-pandemic) had peak-values, which is in accordance with the number of hospitalized cases and reanimation ones in Fig. 5.2. Subsequently, in June (post-peak-pandemic) and for each hospital, the AvTT had an intermediate value between the two aforementioned periods. This can be due to the increasing number of patients that returned home and did not need special care as the 'ret. home' curve in Fig. 5.2 indicates. Proportionally, the number of hospitalized and reanimation cases started to decrease,

which alleviates the healthcare system.

Table 5.2: Data analysis, for each hospital, on the mean $\pm$ std TT values in minutes per month during 2020.

	January	February	March	April	May	June
<b>CHRU</b>	18.34 $\pm$ 7.75	18.93 $\pm$ 8.39	21.88 $\pm$ 10.62	22.90 $\pm$ 9.12	21.55 $\pm$ 8.89	20.38 $\pm$ 9.45
<b>CHIH</b>	14.92 $\pm$ 6.81	14.15 $\pm$ 4.95	16.38 $\pm$ 6.64	20.19 $\pm$ 10.55	17.44 $\pm$ 6.20	16.32 $\pm$ 7.86
<b>HNFC</b>	23.29 $\pm$ 10.63	21.11 $\pm$ 7.22	25.15 $\pm$ 11.27	24.24 $\pm$ 9.02	27.26 $\pm$ 9.92	24.38 $\pm$ 9.77

With these elements in mind, it is now evident that the COVID-19 pandemic has affected the ambulances' TT in hospitals of Doubs-France. This increment might affect the SDIS25 service to the population as their staff and resources would be in use for more time. That is, as a chain-like effect, this would increment the number of breakdowns for the fire brigade when needing to prepare for future interventions, as presented in the work [152]. In the next Subsection 5.2.3, we present and discuss an analysis of the service breakdown during 2020 to validate the negative impact of the COVID-19 pandemic to SDIS25.

### 5.2.3/ SERVICE BREAKDOWN ANALYSIS

Besides an in-depth analysis of the COVID-19 impact on ambulances' TT in hospitals of Doubs France, we also make an study on the chain-like effect that high TT had on the fire service. Following the methodology described in Chapter 3, Fig. 3.2, implemented in Chapter 9, and published in [152], we applied the breakage calculation for the months of March to June 2020, considering only the public service breakdowns of Rescue People (SAP). Public service breakdowns occur when there are no adapted engines, firemen, or both in centers. This means that no adapted armament is available for a certain period of time to respond to an incident.

Fig. 5.3 shows the average breakage time in seconds per day for the months mentioned above comparing the years 2019 and 2020. One can notice small peaks during the months of March and June, which are periods before and after the peak of the pandemic, respectively. However, in the first week of April, there is a peak break of almost 2 hours, which corresponds to the increased TT for ambulances in hospitals presented in Fig. 5.2. What is more, April and May have more days with longer breakdown times. As previously analyzed, during these 2 months there were more cases of COVID-19 in the Doubs region, which reveals that there was certainly a chain reaction effect that left the fire service vulnerable, and as a consequence, a certain sector of the population as well.

In more detail, Figures 5.4 and 5.5 illustrate the causes of the breakdowns during the first semester of the years 2019 and 2020, respectively. One can notice that for January and February of both years, the results are close. However, in March 2020 there is notable increment in the number of breakdowns due to lack of engines, which is the starting month with increasing COVID-19 cases. On the other hand, April and May show

a lower number of breakdowns, which is because there were fewer interventions during the lockdown period as people spent more time in their homes. However, according to Fig. 5.3, the breakdown time was longer, as the available adapted engine spent more time in hospitals. Also, from March to June 2020, it can be seen that there were few or no breakdowns due to the lack of firefighters. The reason behind this is that firefighters and especially volunteers were at home and showed more availability, and there were not as many simultaneous interventions as in 2019.

For this reason, this study aims at developing ML models for forecasting the TT of ambulances. Such knowledge is paramount for EMSs, as they could better prepare themselves

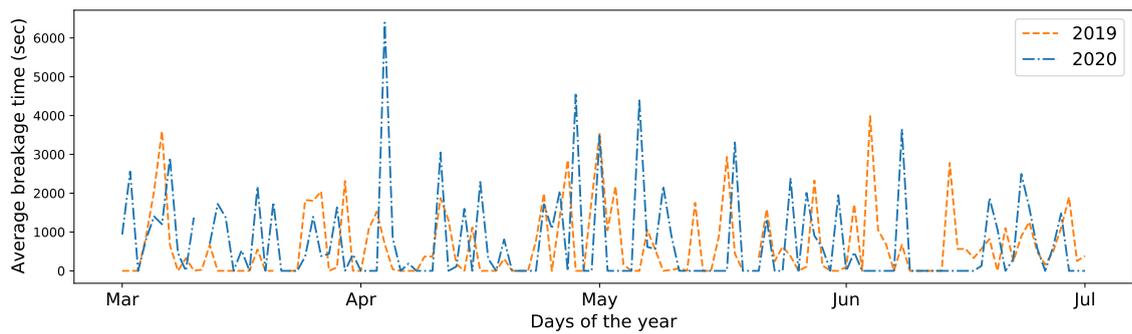


Figure 5.3: Average breakage time in seconds per day for 2020, considering service public breakdowns of the type “rescue people”.

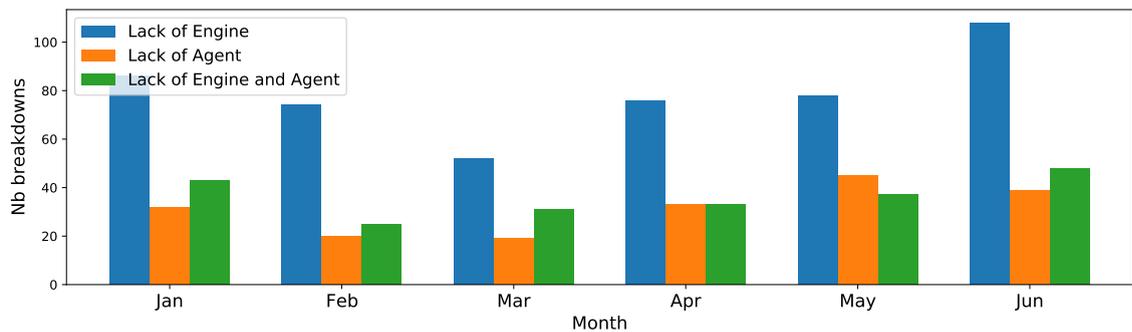


Figure 5.4: Causes of public service breakdowns by month in 2019.

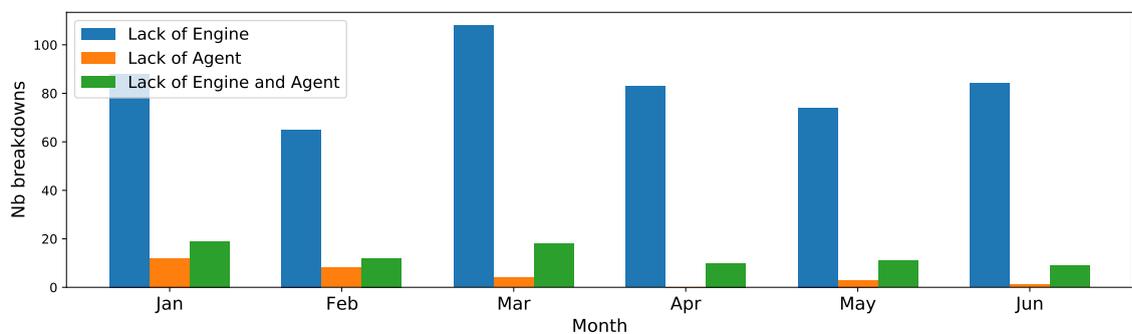


Figure 5.5: Causes of public service breakdowns by month in 2020.

for future interventions and avoid breakdowns on the service as previously analyzed. Such data-driven systems should be of high confidence in order to adequately assist in decision-making solutions in real-life. Moreover, these systems should also be robust enough to abnormal situations such as natural disasters or even the current COVID-19 pandemic.

## 5.2.4/ PROPOSED METHODOLOGY

### 5.2.4.1/ OVERVIEW

The present research proposes a new methodology based on ML techniques for predicting the TT of each ambulance related to a hospital. The approach is composed of two time series models (i.e., regular and irregular). Fig. 5.6 describes the interaction of both models that are created in two stages as explained in the following:

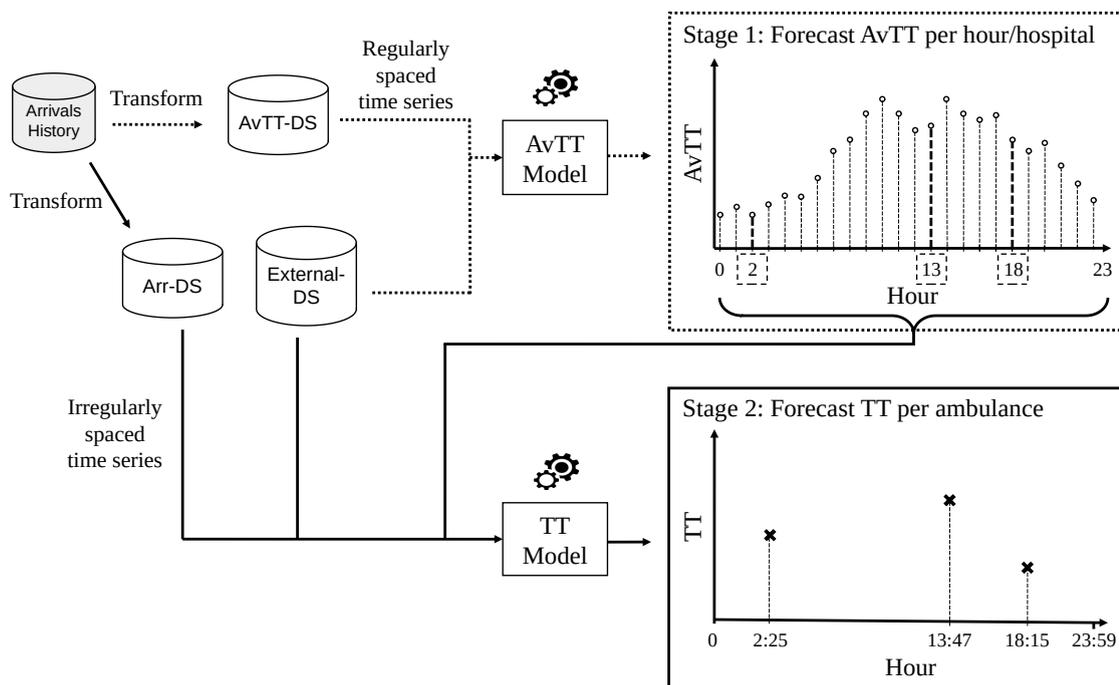


Figure 5.6: The proposed methodology is based on ML techniques to predict the TT of an ambulance in a given hour and hospital. The first stage (dashed line flow) is based on a multivariate model of regularly spaced time series to predict the AvTT of the next hour in each hospital. The second stage (solid line flow) is a multivariate irregularly spaced time series model, which considers temporal variables of each ambulance departure, type of intervention, external variables and the previously predicted AvTT as inputs. For example, to predict the TT of ambulances departing to a hospital at 2:25h, 13:47h and 18:15h, the predicted AvTTs for 2h, 13h and 18h (highlighted) are used, respectively.

*Stage 1: Predicting the average turnaround time per hour at each hospital.* To predict the TT for an ambulance, a valuable input would be the approximate waiting time that

patients may be having in a hospital. However, we do not have internal hospital data to make this prediction. Also, we may consider as entry the AvTT in a given hour for ambulances from different public and private organizations. However, we do not have a record of all of them or their flow. But, what we do have is the history of ambulances arrivals over the years of the fire department of Doubs. So, from this history we generated a regularly spaced time series per hour, which is the AvTT-DS. In this way, it is possible to capture trends and seasonality over time, for example: over the years the AvTT per hour has been increasing, during the day the AvTT is higher than at night, etc. Then, we developed a predictive model of AvTT for the next hour, considering external variables from External-DS, that can influence long waiting periods.

*Stage 2: Predicting the turnaround time for an ambulance.* This stage involves the construction of an irregularly spaced time series model, using the Arr-DS generated from the arrival history, it contains temporal characteristics and the type of intervention. Likewise, we included the External-DS and the AvTT predicted in the first stage. The objective is to forecast the TT of a specific ambulance, that is, at the moment an ambulance warns that it will go to the hospital, we make the prediction to know its approximate TT in a certain hospital. In this way, firefighters will be able to establish better strategies with their resources.

To evaluate the proposed methodology, the months April, May, and June 2020 were selected as the *testing set* for each hospital, since these are the months with a higher number of COVID-19 cases (peak-pandemic period), as previously analyzed in Subsection 5.2.2. For CHRU and CHIH the training set starts from January 2015 until March 2020, and for HNFC the training set is from February 2017 to March 2020.

During the hyperparameter search process, which is developed independently for each model (AvTT and TT), each iteration tests a different configuration throughout the training cycle. Since we use time series models, in which there is a dependency in prior time steps that allows recognizing the increase in TT over hours and days, the cross-validation-based training process for both models was a rolling-origin evaluation [115, 22] (a.k.a. forward chaining). This is a more realistic approach to re-train time series models with data that becomes available each time (e.g., hour or day) for further predictions. More specifically, our models are first fitted with the training set (with data until March 2020) for each hospital, making predictions *one-step-ahead* for all the hours of a day with the AvTT model, and per ambulance departure with the TT model for a day. This way, for all days in the testing set (April, May, and June 2020), the training set is expanded to include all the known values of each day and the process is repeated, i.e., the training set is updated day by day. Besides, all the predictions of each model (i.e., the three months selected as the testing set) in an iteration are stored, and at the end, we compute the models' metrics. For the hyperparameter tuning process, we used RMSE, described

in Chapter 2, Subsection 2.3.1, as the guiding metric. Thus, a new hyperparameter configuration is generated and the search process is repeated during a given number of iterations. The pre-processing and modeling of each model, AvTT and TT, are described in the following two subsections.

#### 5.2.4.2/ AVTT MODEL: PREPROCESSING AND MODELING

Hour ID	Temporal			Google trends			Traffic		Weather			DataGouv			Windowing			Target			
	year	month	...	COVID-19	grippe	...	bisonDepart	bisonReturn	temperature	pressure	...	hosp.	rea.	...	Avg1	...	AVTT-24h	...	AVTT-1h	AVTT	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
100	2015	1	...	0	52	...	1	3	0	99390	...	0	0	...	15.4	...	15	...	10	25	
101	2015	1	...	0	42	...	1	2	1	99323	...	0	0	...	12.4	...	20	...	15	18	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
47452	2020	3	...	12	93	...	4	2	12	99380	...	48	21	...	20.2	...	18	...	25	30	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 5.7: Illustrated example of the input data structure used for the AvTT model. In gray the hour identifier (Hour ID), in yellow the explanatory variables such as time, traffic, weather, statistics of COVID-19 cases from Data Gouv, trends in Google, number of assisted interventions, AvTT of the previous 24 hours, etc. And in orange, the AvTT to predict.

The dataset for the regularly spaced time series model, that will forecast the AvTT per hour, was organized as follows:

- We took the AvTT-DS and for each sample we added 3 moving averages of AvTT as features, using a window size of 2 hours back.
- Also, the AvTT of the last 24 hours were added as features.
- Finally, we complete the dataset with the features of the External-DS, according to the previous hour, since in real life these are data that we can obtain previously.

The structure can be seen in Fig. 5.7, where each line represents a sample by hour with its identifier (Hour ID), and the columns illustrate the predictors and the target, in yellow and orange, respectively. Next, all features were standardized using Scikit-Learn's [52] StandardScaler function, except by the target, which is the AvTT per hour in minutes.

In the search for the best model that gives us more accurate results, we compared the performance of various intelligent approaches such as traditional neural networks *MLP* and *LSTM*, decision trees like *LGBM*, and a framework specifically oriented to time series called *Prophet*. Each technique was briefly described in Chapter 2, Subsection 2.1.4.3.

To navigate the hyperparameter space and pick the most optimal set, it was used the HyperOpt library [61] with 50 iterations for each technique described previously. This library

is based on Bayesian optimization and the selected logic was the algorithm Tree Parzen Estimator suggest (tpe.suggest), which instead of modeling the probability of an observation given a new configuration, models two density functions according to a defined percentile to finally estimate the expected improvement of a given run. The RMSE metric was used to guide the search for the best configuration per technique.

### 5.2.4.3/ TT MODEL: PREPROCESSING AND MODELING

Departure ID	Type of intervention			AvTT			Temporal			Google trends			Traffic		Weather			DataGouv			...	Target	
	Level1	Level2	Level3	AVTT Predicted	AVTT-48h	...	AVTT-1h	year	month	...	COVID-19	grippe	...	bisonDepart	bisonReturn	temperature	pressure	...	hosp.	rea.	...	...	TT
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	1	2	4	25	15	...	10	2015	1	...	0	52	...	1	3	0	99370	...	0	0	...	...	22
201	1	3	4	15	20	...	15	2015	1	...	0	35	...	1	2	1	99325	...	0	0	...	...	14
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
78700	2	3	3	30	18	...	25	2020	3	...	12	100	...	4	2	9	99390	...	52	32	...	...	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 5.8: Illustrated example of the input data structure used for the TT model. In gray the depart identifier, in yellow the explanatory variables such as the type of intervention, the predicted AvTT for the present hour and the AvTT of the last 48h, time variables, Google trends, traffic, weather, etc. And in orange, the TT of the ambulance to predict.

The dataset for the irregularly spaced time series model, that will predict the TT for each ambulance, was constructed as follows:

- We took the Arr-DS and added the features of the External-DS, according to a previous hour, since these features come from sources that export their data online and by hour.
- Since the AvTT model runs earlier, it will give us the average prediction for the current hour, in which an ambulance is on the scene and reports that it will go to the hospital. In this way, the predicted AvTT will be used as a predictor when we forecast the real turnaround time of a specific ambulance. In addition, the AvTT per hour of the last 48 hours were added in order to improve the model's ability to recognize daily seasonality.

The structure can be seen in Fig. 5.8, where each line represents an ambulance departure to the hospital with its identifier (Departure ID), and the columns in yellow and orange exemplify the predictors and the target, respectively. Next, all features were standardized using Scikit-Learn's StandardScaler function, except for the three variables that describe the type of intervention, which are categorical and were encoded using One-Hot-Encoding (OHE) function from Scikit-Learn library too. The target, which is the TT of an ambulance, was kept in its original format (minutes).

To build the time series models, it was chosen the LGBM technique, because of its robustness and execution speed. To find the best set of hyperparameters, we performed 100 iterations using `tpe.suggest` from HyperOpt technique. The loss function used was RMSE.

### 5.2.5/ BASELINE MODELS

Two straightforward baseline models are used to compare the effectiveness of the proposed methodology. The first baseline is to compare with the multivariate regularly spaced time series model from Subsection 5.2.4.2. The second baseline is to compare with the multivariate irregularly spaced time series model from Subsection 5.2.4.3.

#### 5.2.5.1/ BASELINE: AVTT PER HOUR

The baseline model to predict the AvTT per hour (BSAvTT) uses the Arr-DS, where the TT of ambulances until the previous day are averaged by hour, and these values are used as a prediction for each hour during the current day. At the end of each day, if there were ambulances transporting victims to the hospital, the average for each hour is updated.

#### 5.2.5.2/ BASELINE: TT FOR AN AMBULANCE

The baseline model for predicting the TT of an ambulance at a given hour and hospital (BSTT) considers adding the External-DS features, according to a previous hour, to each ambulance in the Arr-DS. In this solution, we do not consider as features any AvTT per hour of previous or predicted hours.

Similar to the standardization described in Subsection 5.2.4.3, predictors were transformed into features using the `StandardScaler` and `One-Hot-Encoder` from Scikit-Learn's library. The target kept its original value in minutes. The technique used for the modeling was LGBM and for the optimization of hyperparameters was HYPEROPT with its `tpe.suggest` algorithm, 100 iterations, and RMSE as the loss function.

## 5.3/ RESULTS

In this section, it is first analyzed the results for the time series model to predict AvTT per hour and hospital (Subsection 5.3.1). Further, we analyze results for the main goal of this work, i.e., to predict the TT for each ambulance (Subsection 5.3.2) transporting victims to hospitals.

The metrics used to evaluate our models are RMSE and MAE, described in Chapter 2, Subsection 2.3.1. And the metric Accuracy with margin of error  $\pm 10$  (ACC10), which is the ratio of number of correct predictions, with a maximum margin of error of  $\pm 10$  minutes, to the total number of input samples. Let  $f(y_i, \hat{y}_i)$  be a function that  $f(y_i, \hat{y}_i) = 1$  if  $\hat{y}_i \in \{y_i \pm 1, y_i \pm 2, \dots, y_i \pm 10\}$  and 0 otherwise. ACC10 is calculated as:  $ACC10 = \frac{1}{n} \sum_{i=1}^n f(y_i, \hat{y}_i) \cdot 100(\%)$ . For example, if  $y_i = 35$  and  $\hat{y}_i = 28$  or  $\hat{y}_i = 41$ ,  $f(y_i, \hat{y}_i) = 1$  in both cases.

### 5.3.1/ FORECASTING THE AVERAGE TURNAROUND TIME OF AMBULANCES PER HOUR

To predict the AvTT of ambulances per hour and hospital during April, May, and June 2020, respectively, the baseline BSAvTT and four ML-based models were evaluated, namely LGBM, MLP, LSTM, and Prophet. Table 5.3 presents the metrics, discussed previously, for each model and hospital, where the best results are highlighted in bold. Both MAE and RMSE metrics express model prediction error in units of the variable of interest AvTT (minutes).

Table 5.3: Prediction results for the AvTT of ambulances per hour and hospital by ML technique.

Hospital	Metric	BSAvTT	LGBM	MLP	LSTM	Prophet
CHRU	RMSE	9.70	<b>7.36</b>	7.78	8.77	7.70
	MAE	6.82	<b>4.94</b>	5.49	6.05	5.34
	ACC10	83.29%	<b>90.16%</b>	88.05%	84.02%	88.46%
CHIH	RMSE	9.08	<b>3.79</b>	4.27	4.85	4.00
	MAE	5.60	<b>1.76</b>	2.60	2.75	2.15
	ACC10	86.81%	<b>97.02%</b>	96.79%	95.28%	96.98%
HNFC	RMSE	10.08	<b>5.97</b>	6.77	7.88	6.66
	MAE	7.58	<b>3.52</b>	4.45	5.53	4.55
	ACC10	77.24%	<b>93.09%</b>	91.30%	86.22%	91.48%

Table 5.4 indicates the range of each hyperparameter we considered in the Bayesian optimization, as well as the best configurations used to train and evaluate the models. The search space for each technique was established with previous empirical experiments that helped to limit the search and define the most influential hyperparameters. The latter are shown in the table for each technique while those that do not appear keep their default values. In the case of LGBM, the maximum number of boosted trees was 1000 with a maximum depth of 12, and subsets of samples and features were greater than 50% of the complete dataset. In the case of neural networks such as MLP, 2 dense layers were established, where the number of neurons varied for each layer. Unlike LSTM, where the number of layers and neurons were initially defined, and the variations occurred at the batch size and learning rate level. In the case of Prophet, the default number of `n_changepoints` is 25, and it was modified to vary up to 100 to recognize the changes in the trends of the TTs over hours, days, weeks, months and years. Similarly, varying the `seasonality_prior_scale` allows us to test different sizes of fluctuations over time.

Table 5.4: Search space for hyperparameters by technique and the best configuration obtained for predicting AvTT per hour for each hospital.

Technique	Search space	Best configuration		
		CHRU	CHIH	HNFC
LGBM	max_depth: [1-12]	6	12	5
	n_estimators: [50-1000]	143	808	768
	num_leaves: [31-100]	40	61	45
	learning_rate: [0.001-1]	0.8498	0.0103	0.0094
	subsample: [0.5-1]	0.99	0.92	0.5
	colsample_bytree: [0.5-1]	0.94	0.6	0.63
MLP	Dense layers: 2	2	2	2
	nb_neurons: [100-500]	(100, 302)	(125, 325)	(362, 562)
	alpha: [0.00001-0.01]	0.000029	0.003674	0.000034
	learning_rate_init: [0.0001-0.1]	0.0355	0.003611	0.000543
	max_iter: [50-200]	154	86	177
	tol: [0.00001-0.01]	0.003731	0.002888	0.000661
	momentum: [0.00001-0.01]	0.0073	0.006057	0.00479
	Early stopping: 20	20	20	20
LSTM	LSTM layers and neurons: 1, (110)	1, (110)	1, (110)	1, (110)
	Dense layers and neurons: 2, (128, 1)	2, (128, 1)	2, (128, 1)	2, (128, 1)
	Activation function: ReLU	ReLU	ReLU	ReLU
	Dropout: 0.5	0.5	0.5	0.5
	Loss function: 'mse'	'mse'	'mse'	'mse'
	Optimizer: Adam	Adam	Adam	Adam
	Early stopping: 15	15	15	15
	Max. epochs: 100	100	100	100
	Batch size: [40-250]	93	142	112
	Learning rate: [0.005-0.01]	0.00841	0.00595	0.00894
	Prophet	n_changepoints: [20-100]	45	75
seasonality_prior_scale: [0-50]		36.82	18.17	12.13
holidays_prior_scale: [0-50]		23.47	22.42	34.77

Moreover, to highlight the effectiveness of the techniques in some peak values, Fig. 5.9 illustrates for each hospital and two days (48h), the comparison of each ML model to forecasting the AvTT of ambulances per hour in different days. Lastly, Fig. 5.10 illustrates for each hospital, the time series results for the period with more breakdowns in service (last weeks of April and the first weeks of May 2020), according to Fig. 5.3, during the COVID-19 peak-period. This figure considers only the LGBM model, according to the best results of Table 5.3, and the baseline BSAvTT.

One can notice in Table 5.3 that all ML-based methods consistently and considerably outperform the baseline BSAvTT model. Among the four ML-based models, LGBM presented the best performance for all metrics and hospitals. Similar metric results were achieved by MLP and Prophet. Additionally, LGBM provided faster execution time than neural network-based methods (MLP and LSTM) and Prophet. For hospital CHIH, LGBM provided lower estimation error with  $RMS E = 3.79$  and  $MAE = 1.76$  minutes, which naturally led to more correct predictions, considering a margin of error of  $\pm 10$  minutes, and  $ACC10 = 97.02\%$ . These results demonstrate that it is possible to forecast the AvTT of ambulances in hospitals per hour with good accuracy for practical purposes, i.e., 97% of the time, the error is less than or equal to 10 minutes. Similarly, although not as good as for hospital CHIH, hospitals CHRU and HNFC present ACC10 metric higher than 90% with RMSE about 6 to 7 minutes.

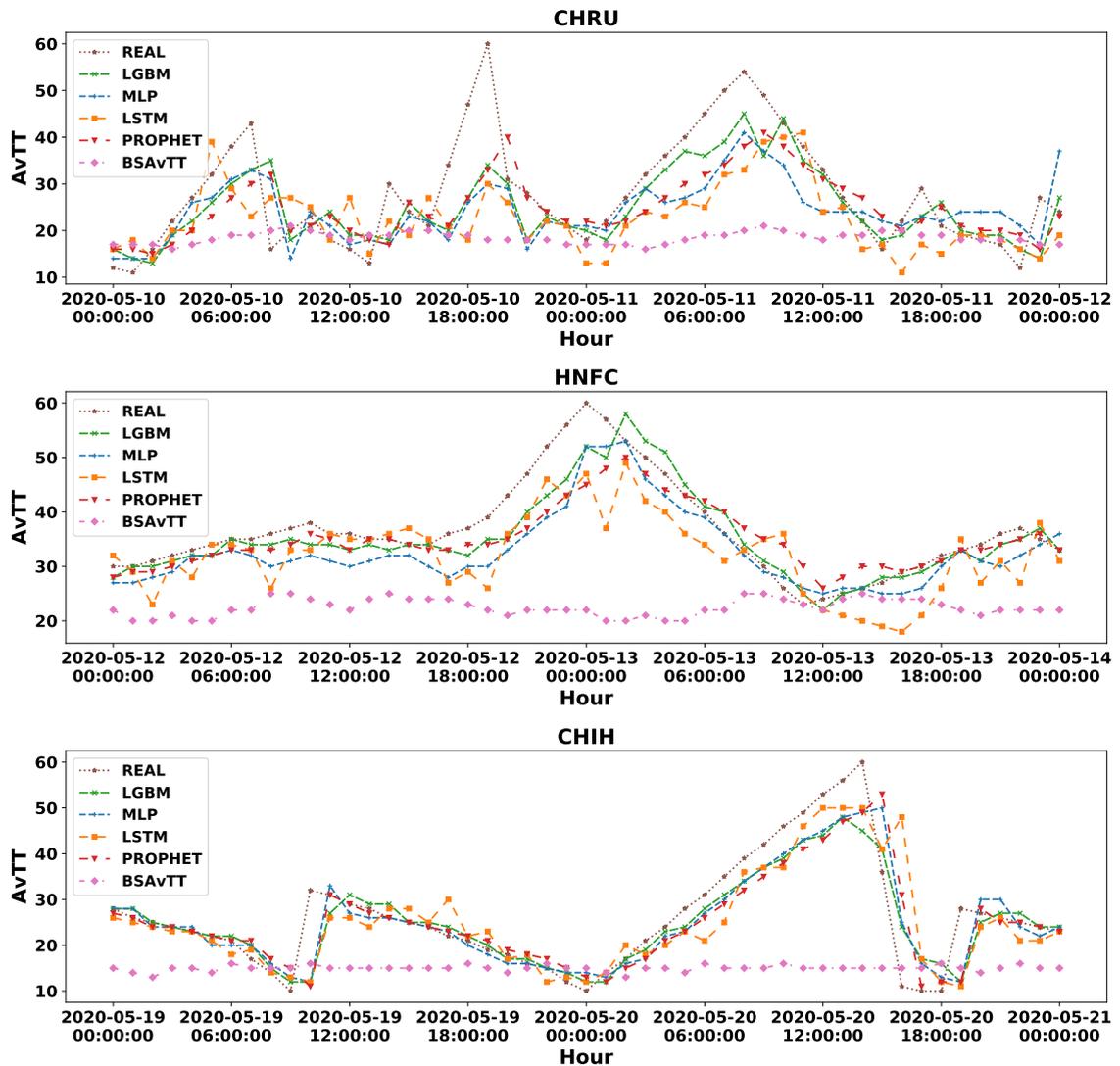


Figure 5.9: Comparison between the real and the predicted AvTT for ambulances per hour in each hospital, namely CHRU, CHIH, and HNFC, respectively, by each ML technique and the baseline BSAvTT.

One can notice in Fig. 5.9 that ML-based models follow the AvTT hourly trend, as well as some peak values. In Fig. 5.10, while LGBM presents accurate prediction results for the AvTT of ambulances for all three hospitals, BSAvTT presents poor predictions following a similar pattern for them through the days. Due to COVID-19, the TTs of ambulances increased during some periods (cf. Fig. 5.2), and therefore, using only the average per hour resulted in poor performance.

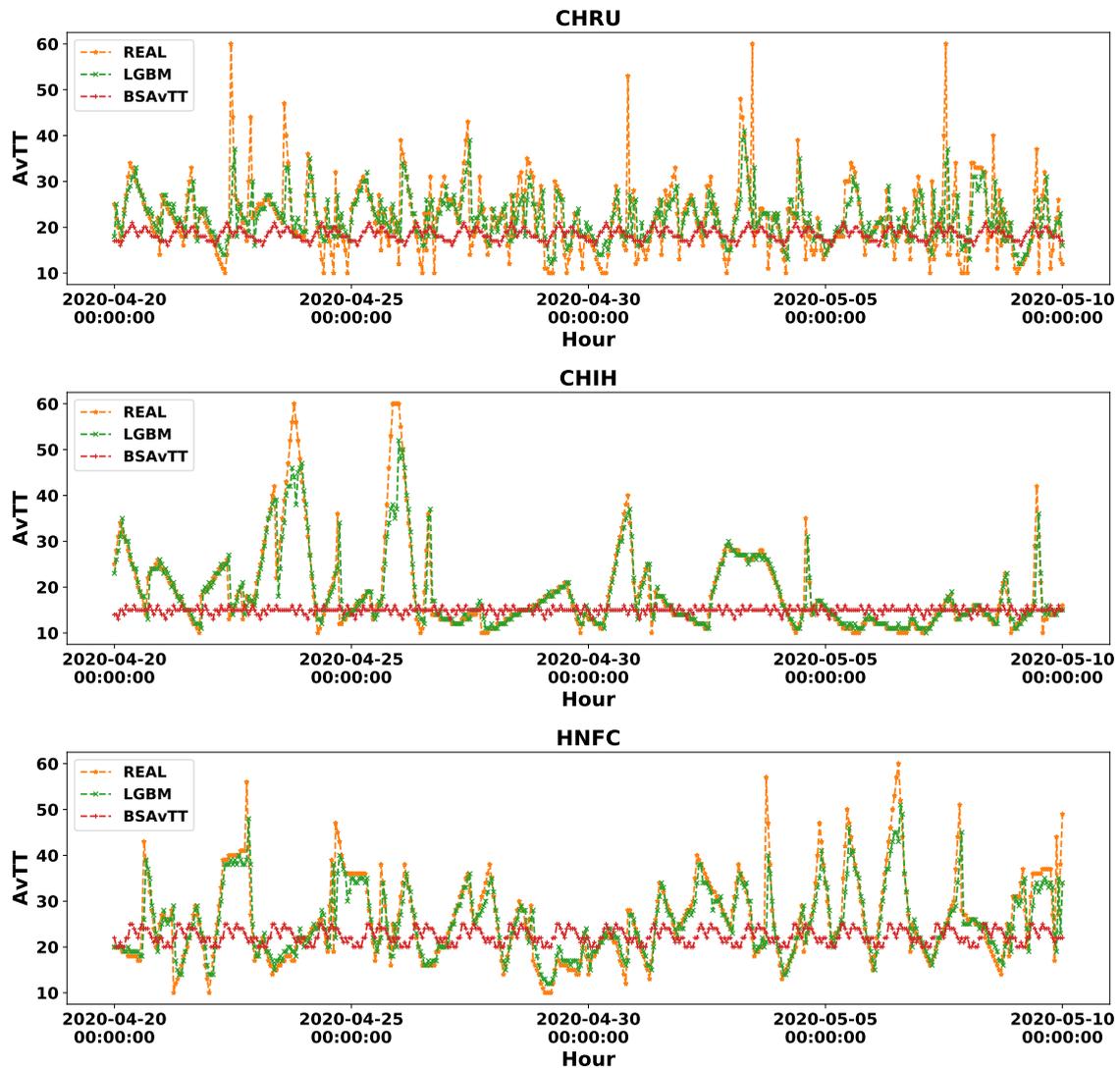


Figure 5.10: Comparison between the real hourly AvTT, the predictions of the best model obtained with the LGBM technique and the predictions of the baseline model BSAvTT, for each hospital (CHRU, CHIH and HNFC). It is observed that the AvTT predicted by LGBM better recognizes the patterns of the mean times and their variation through time, while the BSAvTT maintains a more constant pattern.

### 5.3.2/ FORECASTING THE TURNAROUND TIME FOR EACH AMBULANCE

To predict the TT of each ambulance in a given time and hospital during April, May, and June 2020, respectively, the baseline BSTT and our proposed methodology were evaluated. In both cases, LGBM is the modeling technique. In our proposal, there are additional predictors such as the AvTT predicted by the first (and best) regularly spaced multivariate time series model, i.e., also using LGBM, as well as past AvTTs for recognizing average daily trends. Table 5.5 presents the results for each model and hospital, where the best one are highlighted in bold. Both MAE and RMSE metrics express model

prediction error in units of the variable of interest TT (minutes). Table 5.6 indicates the range of each hyperparameter we considered in the Bayesian optimization, as well as the best configurations used to train and evaluate the models. The description of the LGBM search space follows the one made in Subsection 5.3.1.

Table 5.5: Prediction results for the TT using the proposed methodology and the BSTT model.

Hospital	Metric	BSTT	Proposed methodology
CHRU	RMSE	13.42	<b>12.92</b>
	MAE	8.74	<b>8.31</b>
	ACC10	73.96	<b>74.42</b>
CHIH	RMSE	13.99	<b>10.53</b>
	MAE	7.70	<b>5.37</b>
	ACC10	82.97	<b>86.63</b>
HNFC	RMSE	13.31	<b>11.70</b>
	MAE	9.58	<b>7.74</b>
	ACC10	68.02	<b>76.67</b>

Table 5.6: Search space for hyperparameters in LGBM and the best configuration obtained according to the method applied for predicting the TTs of each ambulance.

Method	LGBM search space	Best configuration		
		CHRU	CHIH	HNFC
BSTT	max_depth: [1-12]	2	1	1
	n_estimators: [50-1000]	978	820	678
	num_leaves: [31-100]	98	59	99
	learning_rate: [0.001-1]	0.0034	0.0099	0.0157
	subsample: [0.5-1]	0.78	0.68	0.55
	colsample_bytree: [0.5-1]	0.86	0.89	0.92
Proposed methodology	max_depth: [1-12]	6	2	9
	n_estimators: [50-1000]	150	129	401
	num_leaves: [31-100]	94	46	75
	learning_rate: [0.001-1]	0.1888	0.0956	0.0299
	subsample: [0.5-1]	0.72	0.65	0.79
	colsample_bytree: [0.5-1]	0.73	0.76	0.5

Moreover, Fig. 5.11 illustrates the comparison of each model to forecasting the ambulance's TT in a given time and hospital, in different periods during the COVID-19 outbreak. Notice that there might be several or no ambulance in a given hour in the hospital. Also, this figure plots the ambulances' TT according to the moment in which they reported that they were going to the hospital.

As one can notice in Table 5.5, the proposed methodology consistently outperforms the straightforward BSTT model. While for hospital CHRU the difference is small and favoring our methodology, for hospital CHIH and HNFC, our proposal considerably outperforms the baseline model. For instance, our proposed methodology provided a lower estimation error with  $RMSE = 10.53$  and  $MAE = 5.37$  minutes for hospital CHIH, which naturally led to more correct predictions by achieving  $ACC10 = 86.63\%$ . Similarly, hospitals CHRU and HNFC present ACC10 metric around 75% with RMSE about 12 to 13 minutes. These results demonstrate that it is possible to forecast the TT of ambulances in a given time and hospital with good accuracy for practical purposes, i.e., 86% of the time, the error is less than or equal to 10 minutes for CHIH, for example. These results are also reflected in

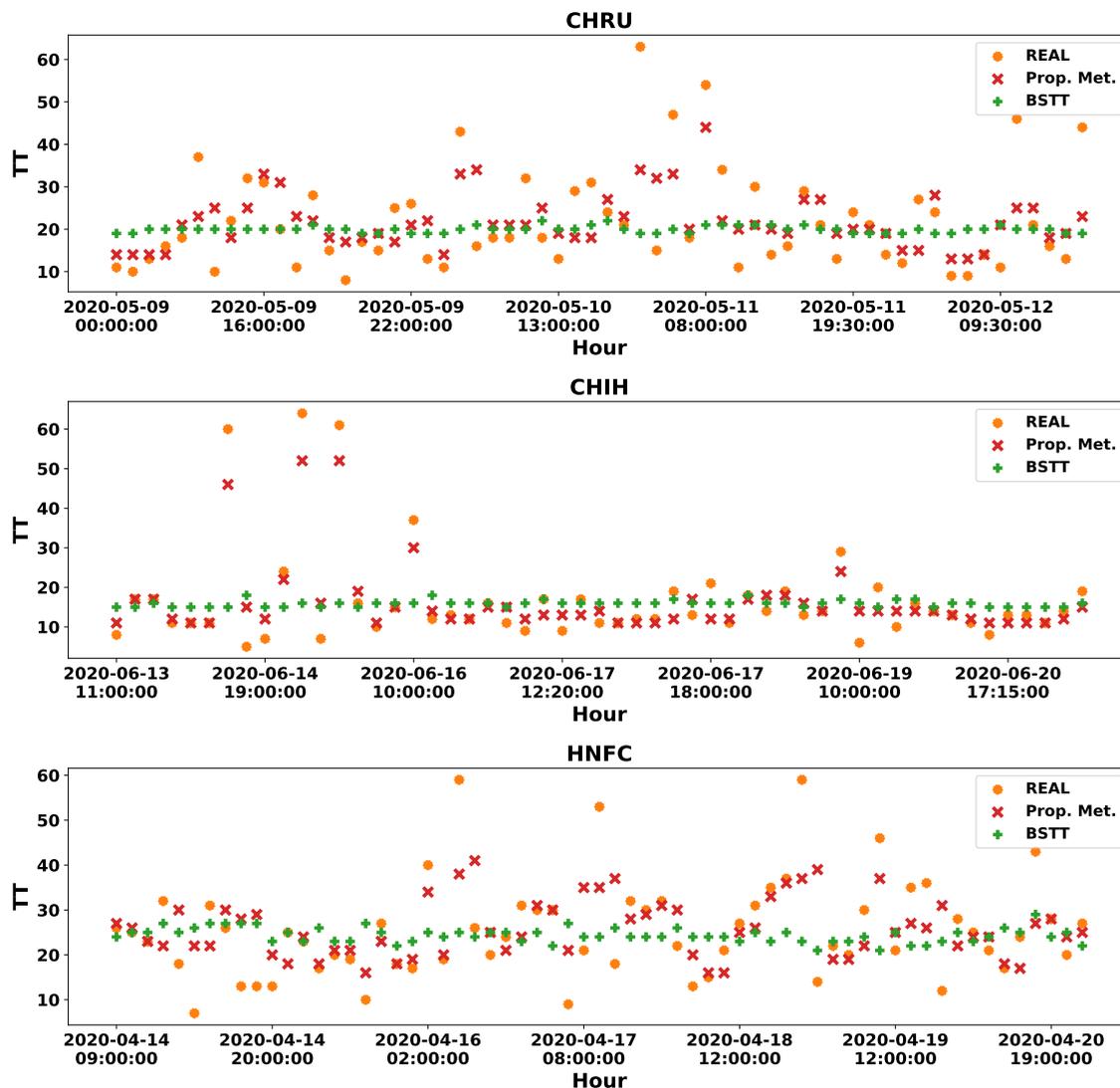


Figure 5.11: Comparison between the real TT of each ambulance for each hospital (CHRU, CHIH and HNFC), the predictions of the proposed methodology and the predictions of the baseline model BSTT.

Fig. 5.11, where our proposed methodology recognizes high- and low-peak TTs of ambulances. On the other hand, BSTT tends to underestimate the TTs for many ambulances by forecasting values around a mean value.

## 5.4/ DISCUSSION AND RELATED WORK

Extended waiting times to transfer patient(s) from ambulances to EDs (i.e., AOD) and eventually, high TTs for ambulances, may lead to numerous consequences for patients' care, financial losses, and unavailability of providing adequate emergency medical services [116, 37, 88, 30, 33]. For example, in [30], the authors conducted a prospective

longitudinal study to analyze the impact of ED overcrowding on firefighters' AOD. The study identified 21,240 cases where their ambulances were out of service due to AOD, which may have a significant effect on their ability to respond to other calls. As noticed in the survey work on AOD from Li et al. [116], the impact of AOD on EMS resource availability has received less attention from the research community. In [18], the authors coined the term *turnaround interval* for the total time an ambulance spends at the hospital, i.e., transferring patient(s)' care, completing paperwork, and reestablishing the equipment for the next call. The authors conducted a prospective study by analyzing firefighters' ambulances' activity on transferring 122 patients to a hospital. In [62], the authors designed a discrete event simulation model to evaluate the change on AOD by having dedicated ED nurses for ambulances' handover. Although the authors identified that this practice may reduce the ambulances' TT, this would also lead to low staff utilization. In [53], the authors investigated the relationship between ambulances' TT with patients' acuity, destination hospital, and time of the day, using one-year data from 61,094 patients. In [76], the authors have analyzed the impact of ED overcrowding and ambulances' turnaround interval.

Notice that most aforementioned works have been performed to highlight the importance of the topic and to identify the relationship between ED overcrowding and ambulances' TT. At the time of this study and to the authors' knowledge, there was no research trying to forecast the TT of ambulances in hospitals, as we presented in this research, but rather for the prediction of patients' waiting time in ED [48, 60, 163]. On the other hand, within the context of COVID-19, multiple works have investigated machine-learning-based solutions to help to fight the outbreak, e.g., forecasting ED volume [167], forecasting the number of confirmed COVID-19 cases [177, 170, 180, 155], and survey works on ML forecasting models for COVID-19 [179, 185].

In this study, we analyzed ambulances' TT from January 2015 to June 2020 on two hospitals (CHRU and CHIH), and from February 2017 to June 2020 on one hospital (HNFC), for the SDIS25 fire department in Doubs, France. We included in our analysis (Subsection 5.2.2) the impact of COVID-19 in the TTs of ambulances in a larger longitudinal study. Further, we analyzed the negative impact due to COVID-19, which increased the TTs of firefighters' ambulances and, consequently, generated more breakdowns in services (Subsection 5.2.3).

The study in [53] identified a high correlation between patient acuity and TT. In our case, there are three variables related to the type of intervention, in which the latter is the severity level of the victim. Taking this into consideration and for the first time, this research proposed a ML-based methodology (cf. Figure 5.6) to forecast the TT for each ambulance in a given time and hospital, using as a reference the forecasting of AvTT per hour for ambulances in hospitals, its type of intervention, and external variables. Indeed, forecasting

the TT of an ambulance in a given time and hospital could provide valuable information for SDIS25, and in general, for EMSs. For instance, EMSs could activate proactive decision-making with the available resources and personnel in order to be able to save more lives. Further, if there are policies on ambulance diversion, EMSs may consider diverging their ambulances to other EDs whose TT is smaller to provide adequate care to the ambulances' patient(s), as well as, returning on service in less time. Besides, the predicted AvTT for ambulances in the next hour(s) could provide approximate information for ambulances' TT, which hospitals and EMSs services may consider as a reference value, and use as a predictor in a second model as we propose.

These data-driven systems should be of high confidence to assist adequately as a decision-support tool in real-life. Moreover, these systems should also be robust enough to abnormal situations such as natural disasters and pandemics. For instance, in adverse cases such as the current COVID-19 pandemic, the need for such a kind of information (ambulances' TT and AvTT) becomes vital since healthcare systems may saturate. While our work was motivated to include the impact of the novel COVID-19 pandemic, our solutions are not limited to it. Indeed, these forecasting models could help other private or public EMSs to forecast the TT of their ambulances and AvTT of hospitals, as well as to future abnormal situations.

To evaluate our proposed methodology, experiments were performed for the period of April, May, and June of 2020, which considers the peak-period of the COVID-19 first wave in the Doubs Region (cf. Figure 5.2) and a high number of breakdowns in the SDIS25 service (cf. Figures 5.3 and 5.5). As shown in the results, it is possible to accurately forecast the TT of each ambulance in a given time and hospital, as well as, to forecast the AvTT of ambulances per hour and hospital. On the one hand, our proposed methodology achieves ACC10 metrics ranging from  $\sim 75\%$  (CHRU) to  $\sim 87\%$  (CHIH), considering a margin of error of  $\pm 10$  minutes, which are promising accuracies for practical purposes to forecasting the TT of an ambulance. In addition, the multivariate regularly spaced time series model trained with LGBM achieves ACC10 higher than 90% for all three hospitals for predicting AvTT.

In our experiments, it is remarkable the improvement of the results with the proposed methodology, for such complex and important tasks, comparing to straightforward prediction models such as the baselines BSAvTT and BSTT. On the one hand, BSAvTT is a straightforward solution that considers only the historical data of ambulances' TT and averages the TT per hour for prediction. Even though this is an intuitive solution that hospital staff and/or EMS could think of applying, we demonstrate that training multivariate time series models based on ML result in much higher performance. For instance, external variables such as meteorological (bad weather may results in more traffic accidents, floods, etc), temporal (day of the week/year, holiday or not), trends (google search

for disease-related keywords), are variables that may lead to reasons of overcrowding in hospitals' EDs. On the other hand, BSTT is a straightforward model that one might consider applying by using only the historical data of ambulances' TT, external variables (traffic, weather, etc.), and variables related to the incident (type, hour, etc.). However, we demonstrate that adding the predicted AvTT of ambulances for the hour an ambulance is going to the hospital and past AvTTs, leads to much higher performance for this complex task.

Finally, the present work has some key limitations that are described in the following. We analyzed and build our models using the data of SDIS25. Although it may represent a sufficient amount of samples, other emergency medical services in France also transport victims to hospitals' EDs. In addition, there was no data from hospitals such as a history of all ambulances' TT, patient flow and the number of doctors on duty, for example. These variables could be of high importance as they are internal to hospitals, which would help ML models to better forecast the TT of each ambulance and AvTT for ambulances. Furthermore, the present work was published in the Applied Soft Computing journal [178].

## 5.5/ CONCLUSION

In this Chapter, we have investigated the impact of COVID-19 on ambulances' TT and developed a methodology to predict their turnaround time. We analyzed and noticed (objective a) that SDIS25 ambulances' TT increased during the outbreak of COVID-19, which led to more breakdowns in service. This increase was due to the shortage of ambulances that would normally return to service in less time. In this way, aiming to provide a decision support tool for SDIS25 (and EMSs in general), this study proposed a novel ML-based methodology to forecast the TT of each ambulance in a given time and hospital.

More precisely, considering that ambulances could arrive at hospitals anytime, we defined our problem as irregularly spaced time series [115], i.e., the spacing of observation times is not constant (e.g., per day). Thus, we first proposed to reconstruct a regularly spaced time series (i.e., AvTT per hour), which gave us a reference on the average time that several ambulances could have been waiting (objective b), and thus, recognize a daily or weekly seasonality. Since not every hour nor every day SDIS25 ambulances went to hospitals, a linear interpolation was applied to complete the dataset. This is because SDIS25 is not the only institution to transport victims to hospitals. Finally, besides the prediction of AvTT per hour/hospital, we could refined the predictions for each ambulance (objective c), i.e., an irregularly spaced time series problem.

Four state-of-the-art ML models were considered in this study, namely, Light Gradient Boosted Machine, Multilayer Perceptron, Long Short-Term Memory, and Prophet. As shown in the results 5.3, the proposed methodology provided remarkable results for practical purposes. The AvTT accuracies obtained for the three hospitals were 90.16%, 97.02%, and 93.09%. And the TT accuracies were 74.42%, 86.63%, and 76.67%, all with an error margin of  $\pm 10$  minutes. These results were published in [178].

# PREDICTING AMBULANCE RESPONSE TIME

In the previous chapter, we have developed methodologies for the prediction of the ambulance's turnaround times during the first COVID-19 quarantine, in France. In this chapter, we will develop ML models to predict ambulance response time (ART), i.e., predict the minutes elapsed between the reception of the alert and the arrival of the ambulance at the scene. In addition, we will evaluate and recognize the most influential variables in the construction of our models.

## 6.1/ INTRODUCTION

Ambulance response time (ART) is a key component for evaluating pre-hospital emergency medical services operations. ART refers to the period between the EMS notification and the moment an ambulance arrives at the emergency scene [111, 121], and it is normally divided into two periods: the pre-travel delay, from the notification to the ambulance dispatch, and the travel time, from the ambulance dispatch to arrival on-scene. In many urgent situations (e.g., cardiovascular emergencies, trauma, or respiratory distress), the victims need first-aid treatment within adequate time to increase survival rate [57, 121, 158, 111, 79, 131] and, hence, improving ART is vital.

As noticed in [150, 146], the SDIS25 and fire departments in general, have been facing a continuous increase in the number of interventions over the years, which may have adverse consequences on ARTs. For instance, the pre-travel delay affects directly ARTs if there is a lack of human and material resources when a call is received. This means, if there is a lack of firefighters, ambulances, or both, ART may be higher than allowed and, hence, a breakdown in the SDIS25 service occurs [152]. This inability to assist within the time limits impacts negatively both EMS and victims because the safety of a certain area or population will be at risk. Thus, there is a need for an intelligent ART prediction system,

which can assist SDIS25 (and EMS, in general) in the dispatching of ambulances.

Indeed, predicting ART is useful for many reasons. It can help in choosing the best center to provide the ambulance. As mentioned in 3.3, SDIS25 has a deployment plan, i.e., each city in the department is associated with an ordered list of centers with the needed engine to respond, so that the first centers are the most likely to provide a rapid and adequate response. This structure is mainly defined by the administrative policies of the organization, which considers, for example, the operational load (number of interventions) that the city represents, and according to this, the necessary armament that its nearest center should have, as well as the shortest distances and times between the centers and the cities. However, this structure varies very little over time, for example, when there is a creation or territorial modification of a city. Although it takes into account the actual travel distance (considering street structures, highways, etc), it does not take into account the real-time state of road traffic, weather conditions, etc. Predicting ART would therefore make it possible to move from static center scheduling to dynamic scheduling. It would also make it possible to estimate the pre-travel delay partially and to see in advance whether, at a given moment, a center is at risk of running out of ambulances. In other words, it enables the anticipation of breakdowns and the redeployment of resources.

### 6.1.1/ DESCRIPTION OF THE OPERATIONAL PROCESS

The focus of this study is on interventions with *victims* that were further transported to hospitals. In these interventions, there was a need for an *emergency and victim assistance vehicle*. VSAVs are equipped with adequate material and personnel for first-aid treatment in urgent situations. In this chapter, we interchangeably use the term 'ambulance' when referring to VSAV.

As described in Chapter 3, section 3.3, the process of an intervention starts with an emergency call received by an operator of the SDIS25 call center. Next, the adequate crew/engine is notified, i.e., an alert is received by the center (start of the ART). Once the sufficient armament is gathered, the ambulance goes to the emergency scene. Upon arriving on-scene, the crew uses a mechanical system to report their arrival (end of the ART). This is illustrated in Fig. 3.1.

The operation process to decide the adequate SDIS25 center to attend the intervention depends on the exact *location* of the intervention. As stated in 3.3, there is a *commune*, a *quartier*, and a *zone* that jointly define a list of priority centers, which are responsible for the call. The reason for such a list is because a single center may not have sufficient resources at the time of the alert to attend an intervention. In this case, if the first center of the list does not have sufficient resources, another center(s) would be in charge of the call. Also, many situations may generate several victims (e.g., traffic accidents, floods).

In these cases, a single intervention can require more than one ambulance, which can come from different centers depending on the availability of resources. This means different ARTs for the same intervention and, therefore, we focus on each ambulance in our analysis and predictions.

In addition, although in some countries the reason of the emergency may require a recommended ART [26], for SDIS25, ART depends on the zone as detailed in [152]. There are three zones: Z1 refers to urban areas, Z2 refers to semi-urban areas, and Z3 refers to rural ones. Therefore, SDIS25 ambulances should arrive on-scene with  $ART \leq 10$  minutes (min) on Z1 and with  $ART \leq 25$  min on Z2 and Z3 3.1, i.e., including the pre-travel delay (gathering armament) and travel-time. If these time limits are not reached, a breakdown in SDIS25 services is generated [152]. The victim state may also be impacted negatively with high ARTs [79, 111]. Lastly, SDIS25 may also help other EMS outside the Doubs region, and in this case, there is no pre-defined ART limit by SDIS25.

### 6.1.2/ OBJECTIVES

With these elements in mind, the present study defines the following objectives:

- a) *Develop ML-based models to predict ART.* We will compare the performance of two state-of-the-art ML techniques based on decision trees (XGBoost and LGBM), which are known for their high performance (and speed) with tabular data; a traditional and well-known neural network (MLP), and a classical statistical method (LASSO) that can perform both variable selection and regularization. This will help us to recognize the model that best fits our data.
- b) *Recognize the most influential variables* when building accurate ML-based models to predict ART. This would allow other EMS to collect these variables and recreate our methodology or develop their own taking into account their policies.

## 6.2/ MATERIALS AND METHODS

In this section, we describe the collection process of internal and external variables, we analyze the recorded ARTs through the years and by zone, and we present our proposed methodology for forecasting the response time of each ambulance and the variables with the greatest impact.

### 6.2.1/ DATA COLLECTION

We used retrospective data of EMS operations recorded by SDIS25. All interventions with victim that were attended by SDIS25 centers with a VSAV and further transported to hospitals were eligible for inclusion. These data covered the period of January 2006 to June 2020. The main attributes of this dataset, extracted from 3.1 are described in the following:

- *ID* is a unique identifier for each intervention;
- *SDate* is the "Starting Date" of the intervention, which represents the time SDIS25 took charge of the intervention after processing the call;
- *ADate* is the "Arrival Date" of an ambulance at the emergency scene;
- *Center* is the SDIS25 center from which the ambulance left;
- *Location* is the precise location (latitude, longitude) of the intervention;
- *Zone* is either urban (Z1), semi-urban (Z2), or rural (Z3);
- *Commune* is the city where the intervention took place;
- *Quartier* is the district where the interventions took place.

Each ambulance represents one sample, i.e., a single intervention may have received one or more ambulances. The ART variable was calculated as  $ART = ADate - SDate$ . We excluded outlying observations with ART of less than 1 minute and with ART of more than 45 minutes, which represented less than 1.4% of the original number of samples.

Using *SDate*, we have added temporal information such as: year, month, day, weekday, hour, and categorical indicators to denote holidays, end/start of the month, and end/start of the year. Besides, with the exact coordinates from both *Center* and emergency's *Location*, we calculated the great-circle distance [216], using the GeoPy library [203], to add it as a feature. *Great Circle Distance* is the shortest distance between two points on the surface of a sphere. We used the great-circle distance since it is faster to be calculated than the Geodesic distance and more accurate than the Euclidean distance. Besides, we have added the number of interventions in the past hour and the number of active interventions in the current hour. As also remarked in the literature [57, 85], the number of interventions on previous hours might impact ART. In addition, external data, extracted from 3.2, that may affect ART were gathered from the following sources: Bison-Futé [196] (indicator of traffic level), Météo-France [209] (meteorological information), and OSRM API [51], which gives us the driving distance in kilometers (km) on the fastest route and its travel time duration in minutes for each ambulance. To calculate these distances and times, we used the coordinates from both *Center* and emergency's *Location*.

## 6.2.2/ DATA ANALYSIS

After removing outlying observations, the dataset at our disposal has 186,130 dispatched ambulances from SDIS25 centers that attended 182,700 EMS interventions. The frequency on the number of dispatched ambulances per zone is 39.62% (Z1), 33.38% (Z2), 26.71% (Z3), and 0.29% (outside the Doubs region), respectively. Figure 6.1 illustrates the distribution of our variable of interest, namely ART, via three histograms with bins of 1 minute for each zone within the Doubs region. One can notice that the ART distributions follow a typical right-skewed distribution also observed in other works/countries [57, 29, 32]. The mean and standard deviation (std) values for zones Z1, Z2, and Z3 are  $8.79 \pm 5.66$  min,  $11.43 \pm 6.15$  min, and  $15.38 \pm 6.41$  min, respectively. SDIS25 had about 79.52% of the time  $ART \leq 10$  min on zone Z1, and had about 95.76% and 92.50% of the time  $ART \leq 25$  min on zones Z2 and Z3, respectively.

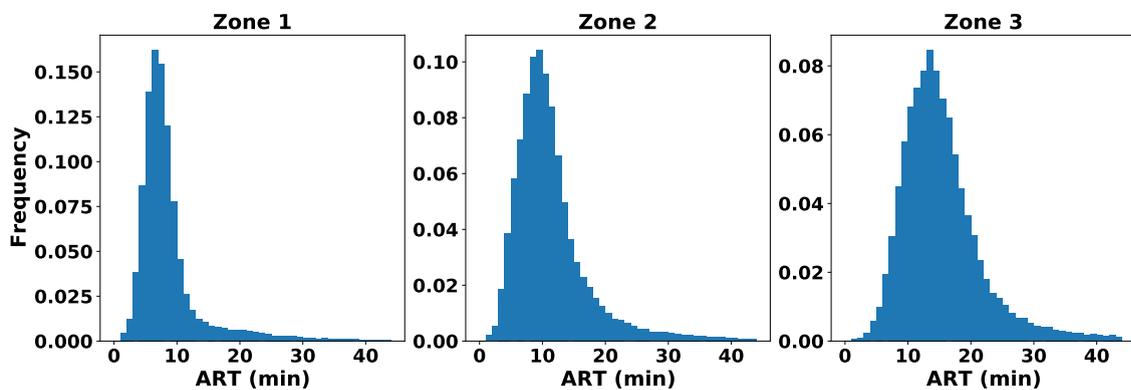


Figure 6.1: Distribution of the ART variable for zones Z1, Z2, and Z3, respectively.

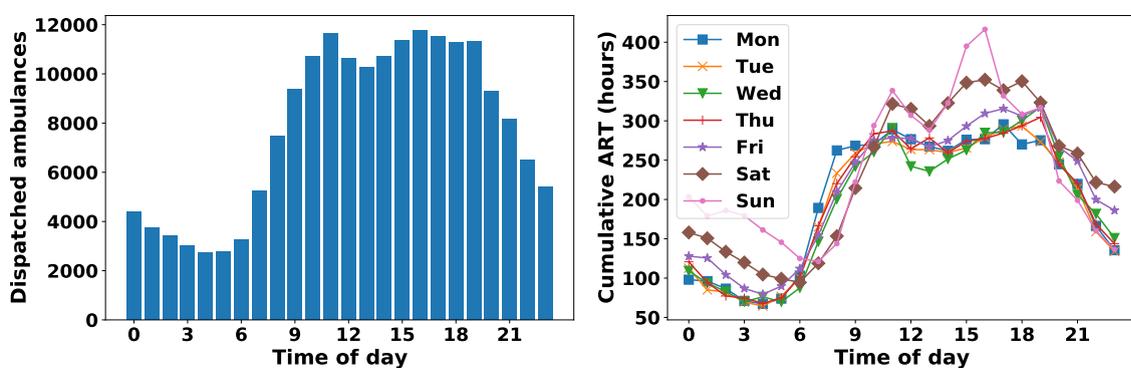


Figure 6.2: Histogram of the total number of dispatched ambulances per hour in the day (left-hand plot) and cumulative ART in hours per day of the week and hour in the day (right-hand plot).

Figure 6.2 illustrates the total number of dispatched ambulances per hour (left-hand plot) and the cumulative ART in hours per day of the week and hour in the day (right-hand plot). One can notice that the total number of dispatched ambulances is notably related

to the hour in the day, i.e., there were more interventions in working periods rather than between 0h to 6h. This behavior is also noticed in the works [32, 46]. Besides, as one can notice with the right-hand plot of Figure 6.2, from 8h in the morning on, the cumulative ART starts to increase and remains high up to 19h when it starts to decrease. While this high cumulative ART can be linked with the high hourly demand, ambulances dispatched during working periods are also more likely to traffic congestion and, naturally, to undergo through longer travel time. Secondly, due to the number of interventions in a given hour, SDIS25 centers may have taken more time to dispatch ambulances if their resources were in use in other incidents. A slightly different profile can be seen on weekends, with noticeable higher cumulative ARTs in the late night (0-6h) and during some hours of the day too.

Summary statistics per year and per zone are shown in Table 6.1. The metrics in this table includes the total number of dispatched ambulances (Nb. Amb.), and descriptive statistics such as mean and standard deviation (std) values for the ART variable. We recall that for the year 2020, these statistics are up to June 2020 only. As also noticed in [146, 150], the number of interventions increases throughout the years. The year 2010 presented high values in comparison with all other years, e.g., for Z1, the average ART was above the 10 min recommendation.

Table 6.1: Mean and std values for the ART variable and the total number of dispatched ambulances (Nb. Amb.) per year in zones Z1, Z2, and Z3, respectively. For 2020, we only consider cases of the first semester.

Year	Z1			Z2			Z3		
	Nb. Amb.	Mean	Std	Nb. Amb.	Mean	Std	Nb. Amb.	Mean	Std
2006	197	9.23	4.41	367	11.25	5.50	354	14.27	5.40
2007	236	7.39	3.05	671	10.79	5.04	595	14.35	5.52
2008	799	8.69	6.04	1,055	11.19	5.32	911	14.53	6.02
2009	1,363	8.76	6.05	2,087	11.08	5.67	1,872	14.94	6.46
2010	2,643	10.08	7.23	2,797	12.48	6.85	2,483	16.01	7.22
2011	5,971	8.26	5.61	4,276	11.24	6.13	3,295	14.50	6.25
2012	6,078	8.66	5.89	4,661	11.18	6.39	3,602	14.86	6.24
2013	6,780	8.82	5.72	5,048	11.03	6.11	3,972	15.07	6.30
2014	6,847	8.37	5.23	5,481	10.80	5.86	4,240	14.91	6.34
2015	7,226	8.46	5.50	5,596	10.86	5.78	4,643	15.02	6.12
2016	7,510	8.50	5.35	6,179	11.19	5.92	4,861	15.32	6.35
2017	8,650	8.76	5.32	7,251	11.49	6.01	5,523	15.51	6.36
2018	9,051	8.90	5.46	7,641	11.64	6.11	5,956	15.59	6.23
2019	7,030	9.42	6.02	6,238	12.29	6.66	5,016	16.60	6.88
2020	3,397	9.73	5.87	2,843	12.59	6.56	2,449	16.46	6.44

### 6.2.3/ PREPROCESSING AND MODELING

In our experiments, each sample corresponds to one ambulance dispatch, in which we included temporal features (e.g., hour, day), weather data (e.g., pressure, temperature), traffic data, the emergency's location (latitude and longitude in radians), and computable features (e.g., distance, travel time). The scalar target variable is the ART in minutes,

which is the time measured from the EMS notification to the ambulance's arrival on-scene. All numerical features (e.g., temperature) were standardized using the *StandardScaler* function from the Scikit-Learn library. Categorical features (e.g., center, zone, hour) were encoded using mean encoding, i.e., the mean value of the ART variable with respect to each feature. The target variable, namely ART, was kept in its original format (minutes) since no remarkable improvement was achieved with scaling.

Four state-of-the-art ML techniques have been used in our experiments to predict ART of each depart. We compared the performance of two techniques based on decision trees (XGBoost and LGBM), which are known for their high performance (and speed) with tabular data; a traditional and well-known neural network (MLP), and a classical statistical method (LASSO), that can perform both variable selection and regularization.

To evaluate our models, we used the metrics RMSE, MAE, and ACC5. ACC5 is the accuracy score with a margin of error  $\pm 5$ min and RMSE is the metric to guide the searching of the best configuration (hyperparameter tuning process) with Bayesian Optimization (HyperOpt library) and 100 iterations.

All models were trained continuously, i.e., initially a new configuration is generated by HyperOpt and the tpe.suggest algorithm. Next, we will train a model with years 2016-2019 to predict January 2020 (testing set). Then, to predict February 2020, we will add to the training set January 2020 and retrain the model with the same configuration. This process will be repeated for the months from March to June. All monthly predictions are stored to calculate RMSE and MAE metrics. Finally, HyperOpt receives the RMSE calculated, generates another configuration of hyperparameters, and a new iteration starts.

## 6.3/ RESULTS

The search space and the best configurations obtained for the 4 techniques are presented in Table 6.2, the metrics resulting from the best models are described in Table 6.3, and the predictions for 80 samples and by method can be visualized in Fig. 6.3.

As we can see when analyzing the performance of the models in Table. 6.3, the best LASSO model obtained the highest RMSE 5.65, for more complex ML-based models, RMSE was less than 5.6, achieving the lowest RMSE 5.54 with XGBoost and LGBM. In comparison with the results of existing literature, similar RMSE and MAE results were achieved in [132] to predict ART.

Fig. 6.3 shows ART prediction results for 80 ambulances dispatched in 2020 out of 8,709. From here, we can see that the performances of XGBoost and LGBM are quite similar. However, as found in the literature and verified with our experiments, LGBM has a lower training time cost than XGBoost [102]. Besides, we observed that in Table 6.2, XGBoost

Technique	Search Space	Best Configuration
XGBoost	max_depth: [1, 10]	9
	n_estimators: [50, 500]	465
	learning_rate: [0.001, 0.5]	0.0265
	min_child_weight: [1, 10]	5
	max_delta_step: [1, 11]	4
	gamma: [0.5, 5]	3
	subsample: [0.5, 1]	0.8
	colsample_bytree: [0.5, 1]	0.5
	alpha: [0, 5]	2
LGBM	max_depth: [1, 10]	7
	n_estimators: [50, 500]	355
	learning_rate: [1e-4, 0.5]	0.0188
	subsample: [0.5, 1]	0.54066
	colsample_bytree: [0.5, 1]	0.5160
	num_leaves: [31, 400]	400
	reg_alpha: [0, 5]	4
MLP	Dense layers: [1, 7]	7
	Number of neurons: [2 <sup>8</sup> , 2 <sup>13</sup> ]	2 <sup>10</sup>
	Batch size: [32, 168]	140
	Learning rate: [1e-5, 0.01]	0.00265
	Optimizer: Adam	Adam
	Epochs: 100	100
	Early stopping: 10	10
LASSO	alpha: [0.01, 2]	0.0205

Table 6.2: Search space for hyperparameters by ML technique and the best configuration obtained.

Table 6.3: ART prediction scores

Method	RMSE	MAE	ACC5 (%)
XGBoost	<b>5.54</b>	3.44	83.94
LGBM	<b>5.54</b>	3.41	84.13
MLP / hour	5.60	3.56	82.80
LASSO / hour	5.65	3.48	83.25

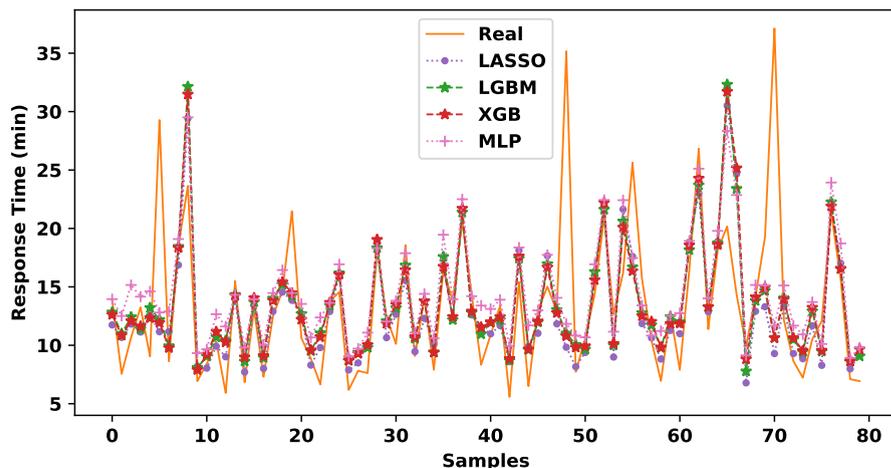


Figure 6.3: Illustration of predictions made by LASSO, XGBoost, LGBM, and MLP for 80 samples.

built more trees ( $n\_estimators = 465$ ) and those were deeper ( $max\_depth = 9$ ) compared to LGBM ( $n\_estimators = 355$  and  $max\_depth = 7$ ), this is mainly because in XGBoost, trees grow depth-wise while in LGBM, trees grow leaf-wise.

Lastly, the importance of the features, taking into account LASSO coefficients and decision trees' importance scores were: averaged ART per categorical features (*e.g.*, center, commune, hour); OSRM API-based features (*i.e.*, estimated driving distance and estimated travel time); the great-circle distance between the center and the emergency scene; the number of interventions in the previous hour, and the number of interventions still active. Immediately thereafter, it appeared the weather data, which were added as "real-time" features, *i.e.*, using the date of the intervention to retrieve the features. Penultimate, the traffic data, which are indicators provided by [196] at the beginning of each year and, which might have shown more influence if they had been retrieved in real-time. Finally, it appeared some temporal variables such as weekend indicators, start/end of the month, and the day of the year.

## 6.4/ DISCUSSION AND RELATED WORK

When we reviewed the literature at the time of these experiments, we found that the main focus was on the analysis of ART [57, 29, 40] and its association with trauma [26, 121] and cardiac arrest [111, 131, 158], for example. To reduce ART, some works propose reallocation of ambulances [149, 79], operation demand forecasting [166, 146, 79, 123, 150], travel time prediction [46], simulation models [54, 32], and EMS response time predictions [46, 132]. The work in [132] propose a real-time system for predicting ARTs for the San Francisco fire department, which closely relates to this study. The authors processed about 4.5 million EMS calls utilizing original location data to predict ART using four ML models, namely linear regression, linear regression with elastic net regularization, decision tree regression, and random forest. In our case, we included weather data, which could help to recognize high ARTs due to bad weather conditions. Furthermore, in our analysis, we compared four different types of techniques, namely, XGBoost, LGBM, LASSO, and MLP, being LGBM and XGBoost the most outstanding in performance. In addition, we identified the features with the most impact, which were mainly the averaged ART per center, commune, and hour; the estimated driving distance and travel time between the center and the emergency scene; the workload described by the number of interventions in the previous hour; and the weather features.

We also identified some limitations in this work. We analyzed ARTs using the data and operation procedures of only one EMS in France, namely SDIS 25. Although it may represent a sufficient amount of samples, other public and private organizations are also responsible for EMS calls, *e.g.*, the SAMU (Urgent Medical Aid Service in English) analyzed in [54]. Besides, there is the possibility of human error when using the mechanical system to report, *i.e.*, record the arrival on-scene time. For instance, the crew may have forgotten to record status on arrival and may have registered later, or conversely, where

the crew may have accidentally recorded before arriving at the location. Also, it is noteworthy to mention that the arrival on-scene does not mean arriving at the victim's side, e.g., in some cases the real location of a victim is at the  $n$ -th stage of a building as investigated in [40].

Finally, learning and extracting meaningful patterns from data, e.g., through ML, play a key role in understanding several behaviors. However, on the one hand, storing and/or sharing original personal data with trusted organizations may still lead to data breaches [182] and/or misuse of data, which compromises users' privacy. On the other hand, training ML models with original data can also leak private information. For instance, in [107] the authors evaluate how some models can memorize sensitive information from the training data, and in [106], the authors investigate how ML models are susceptible to membership inference attacks. To address these problems, some works [153, 113, 146, 123, 98, 161, 23, 20] propose to train ML models with sanitized data, which is also known as input perturbation [41] in the privacy-preserving ML literature. Due to the sensitive data exposed in the present investigation, the study was complemented with geo-indistinguishability data anonymization technique, implemented in collaboration with Héber Hwang Arcolezi. This technique is a state-of-the-art formal notion based on differential privacy, it obfuscates the real location by adding controlled random noise. In this way, EMSs could use or share sanitized datasets to avoid possible data leakages, membership inference attacks, or data reconstructions. This sanitization process is out of the scope of this chapter but the collaborative work was published in the *Mathematical and Computational Applications* journal [176].

## 6.5/ CONCLUSION

Since ART is a fundamental indicator of the effectiveness of EMS systems [158, 121, 111, 79, 26, 131] and an intelligent decision-support system could help to minimize these operational times, in this Chapter, we have evaluated four ML models (LASSO, XGBoost, LGBM, and MLP) to predict the ambulance response time (objective a) and identified the most influential variables in the prediction (objective b).

First, we analyzed historical records of ARTs to find correlations between features and explain the trends through the 15 years of collected data. Then, we sought to predict the response time that each center equipped with ambulances had to an intervention. We determined that XGBoost and LGBM presented the best performances. Finally, we identified the impact variables from the highest to lowest, being the most influential, the average ART by commune and the distance between the center and the intervention's location; and the least influential, month and year.

In this way, we could convert a static resource deployment plan into a dynamic one, in which the selection of the center to attend the intervention would be based on the shortest response time of the closest centers, taking into account the location of the emergency, traffic, and weather conditions. Our results were published in [176], where we considered that the ML models could be subject to attacks and compromise the victims' privacy.



# PREDICTING VICTIM'S MORTALITY AND THEIR NEED FOR TRANSPORTATION

In the previous chapter, we studied the performance of ML models to predict ambulance response time and identified the most influential variables in the construction of our models. In this chapter, we propose a novel methodology based on ML techniques to predict the victims' mortality and their need for transportation to health facilities, using data gathered from the start of the emergency call until a center is notified.

## 7.1/ INTRODUCTION

As mentioned in Chapter 6, shorter ambulance response times are potential contributors to higher survival rates [79, 121] since every second is a matter of life. At the time of this study, several decision-support systems based on machine learning (ML) techniques have been proposed for application in emergency medicine [139, 188, 187]. Indeed, in the context of this paper, for EMS, there are many interests in using ML methods for tasks such as: identifying possible medical conditions before arrival on emergency departments [160], to predict ambulances' demand to allow their reallocation [79], to predict ambulance response time [176, 132], to predict the ambulances' turnaround time in hospitals [178], to predict clinical outcomes [134], to early identify clinical conditions on emergency calls [120], to recognize and predict service disruptions [152], and so on. The response time also depends on how and by whom the call is processed in the EMS center [87]. For this reason, there is a need to optimize these services and take advantage of plenty of data gathered throughout the years in hospitals and EMS.

Therefore, in this study, we analyze the calls and interventions collected by SDIS25 from 2015 to 2020. With this information, we will build intelligent models for the prediction of

victim's mortality, i.e., to identify whether the victim is at serious risk and most likely to die if not attended as soon as possible, or if it is a mild case. Next, we will build another ML model to help us predict whether or not the engaged engine will perform a transfer of the victim at the hospital, private clinics, etc, i.e., to identify if the engaged resources will be busy more time, and consequently, fewer or zero resources will be available at the center, which could generate breakdowns in SDIS25 service. To determine the best ML models, we will compare 4 methods presented in the following. Given the easy and fast use of techniques based on decision trees with tabular data, we selected 2 remarkable representatives: LGBM and XGBoost. Also, given the large amount of data we have and the recognized potential of neural networks, we tested 2 types: CNN and TabNET.

### 7.1.1/ DESCRIPTION OF THE OPERATIONAL PROCESS

The focus of this study is on the calls that became interventions handled by the SDIS25, because we recall that some calls are referred to the SAMU or to other organization.

As described in Chapter 3, section 3.3, and Fig. 3.1, the service is activated through some emergency phone number (e.g., 18 or 112). Next, the call is treated by an operator, which collects the necessary information about the emergency (e.g., victims, location) and notifies the center(s) to deal with the intervention during the call or immediately at the end of the call. Once the adapted armament is ready, the ambulance departs to the emergency scene. After providing first-aid treatment to the victim(s) and depending on the victim's status, the victim(s) is transported to some health facility (HF). Lastly, the ambulance and its crew return to the SDIS25 center and are available again to attend other interventions. Fig. 3.1 shows in green our interval of interested, which is from the EMS call center's phone starts to ring (*Phone rings*) until some center(s) is notified to handle the intervention or the call ends (*End call*). During this study, the interval will be named as *call period* and we will use it to predict whether the victim will die and if the ambulance will need to transport the victim to some HF.

### 7.1.2/ OBJECTIVES

From 2016 to 2020, the SDIS25 has presented an average increase of about 15% in the number of EMS calls (i.e., emergency interventions), compared to 2015, reaching the highest peak in 2018 with an increase of 25%. And just as calls increased, the number of fatalities also increased, on average by 7%, with a peak of 17% in 2018. Similarly, the number of travels to HFs grew on average by 15%, with a peak in 2018 of 26%. This implies that the resources of SDIS25 were engaged for longer, so there were fewer resources in the centers. And with the present scarcity of resources in its service, the SDIS25 is vulnerable to a major crisis (e.g., pandemics, natural disasters), which puts the

population at risk. For this reason, an intelligent system is needed to allow them to predict the urgency of the intervention (victim's mortality) and plan the distribution of resources that will be engaged (victim's transportation).

This way, within the *call period* and with all the accessible information about the call processing time; operators' and victims' personal data, and the intervention itself, the objective of this study is to contribute with a novel ML-based methodology to predict both aforementioned targets, which can be used as a decision-support system by SDIS25 (and EMS in general). Indeed, when a center has received the alert to go to an intervention, the system could launch its predictions to allow EMS to better determine their resources and provide faster response time (and possibly avoid victim's mortality). It would also allow to know that the engaged armament will be in use for less time (by not going to a hospital, for example). In summary, this work defines the following objectives:

- a) **Prediction of victims' mortality:** Predict if the victim will die. This would allow to increase the level of urgency of the call and to improve the selection of adapted engaged resources.
- b) **Prediction of victims' transportation:** Predict the need to transport the victim from emergency scenes to HFs. This would allow knowing if the EMS resources will be engaged for a longer period of time, and as a result, to make a better decision to deploy its resources.
- c) **Variables' impact.** Recognize the variables with the greatest impact for building predictive models. This would allow replication of the process in another EMS.

## 7.2/ MATERIALS AND METHODS

### 7.2.1/ DATA COLLECTION

We used retrospective data recorded by SDIS25 from January 2015 to December 2020. All calls made to SDIS25 lines (e.g., 18 and 112) were considered. Calls that had an intervention identifier indicating that they were attended by SDIS25 centers were eligible for inclusion in the case there were victims. In total, after removing outlying observations, the primary dataset at our disposal comprises 177883 emergency calls with the following explanatory variables:

- Victim: age, sex, and center that assisted the victim; the city where the intervention occurred; distance between the center and the city; victim mortality and victim transportation to HFs. In this chapter, we interchangeably use the term *city* when referring to *commune*.

- Operator: age, sex, grade, and seniority (experience time).
- Call/Intervention: hour, day, day of the week, month, year, delay time to answer the phone, call duration, delay time to diffuse the alert, and type of intervention. The latter is described by 3 variables: type of operation (SAP, INC, etc), subtype of operation (an emergency, fire on the public road, fire in an individual room, etc), and the motive for departure (external hemorrhage, respiratory distress, etc). We also have the reason for departure. The difference between the *motive* and *reason* is that the first is what is recorded by the operator based on the call received (partial), and the second is what is recorded by the firefighters upon return from the mission (confirmed). This last one will not be considered during the modeling since the developed system will make predictions before the armament departs to the mission.

From here, we extracted 2 variables as our predictive targets: the victim's mortality and their need for transport to a hospital. Both will be developed in the following sections.

7.2.2/ DATA ANALYSIS

Table 7.1 exhibits for each year the number (Nb.) of EMS calls, mortality, and transportation to HFs, and the augmentation (Growth in %) throughout years in comparison with 2015. The last row named 'Total' indicates the sum for Nb. and the average for Growth. One can notice that the total number of mortality and transportation to HFs represent 1.94% and 78.72% of the total number of EMS calls, respectively. Also, over the years, the number of EMS calls increased by 14.67% on average, which led to an average increase of 7.17% and 15.27% on the number of fatalities and transportation to HFs, respectively.

Table 7.1: Summary statistics on the number (Nb.) of EMS calls, mortality, and transportation to HFs and the augmentation (Growth) in relation to the baseline year 2015. Total indicates the sum for Nb. and the average for Growth.

Year	EMS Calls		Mortality		Transport	
	Nb.	Growth (%)	Nb.	Growth (%)	Nb.	Growth (%)
2015	26417	—	544	—	20700	—
2016	28529	7.99	532	-2.21	22479	8.59
2017	31311	18.53	562	3.31	24990	20.72
2018	33150	25.49	638	17.28	26271	26.91
2019	29914	13.24	562	3.31	23370	12.90
2020	28562	8.12	621	14.15	22214	7.31
<b>Total</b>	177883	14.67	3459	7.17	140024	15.27

Fig. 7.1 presents statistics on call processing times by time intervals in seconds and their ratio of deaths as a percentage. One can observe that the ratio of deaths increases in the 3 time periods as time increases. This is more noticeable with 'Call duration' that reaches

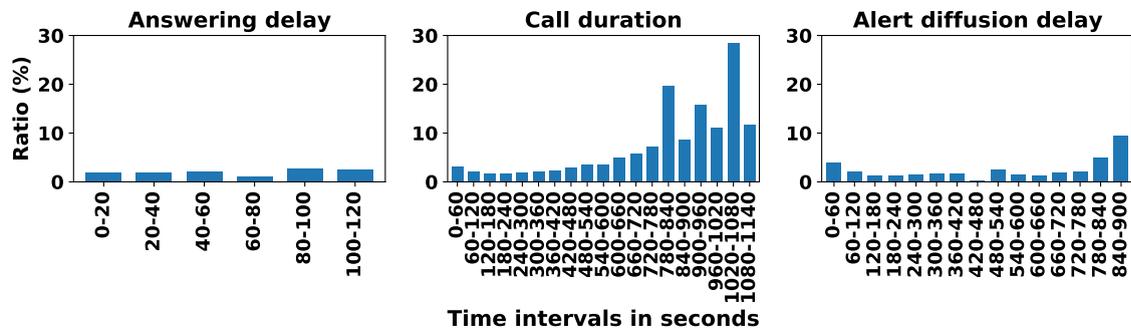


Figure 7.1: Ratio of victims' death ( $Nb. Deaths/Nb. Calls \times 100$ ) according to call center operators' delay time to answer the phone, the total call duration time, and the delay time taken to broadcast the alert, respectively, by time intervals in seconds.

a rate of 28.57% of deaths when the calls lasted between 17 and 18 minutes (interval 1020-1080). Although, there were few calls (14), more cases of deaths appeared (4). Yet, we still kept these cases since they could influence the prediction of victim's mortality.

Furthermore, on analyzing the attributes *motive* and *reason* for departure, we remark that about 75% of the times both values were equal. This means, most of the time, the information was correctly acquired during the call, which helps SDIS25 to correctly define the armament needed to attend the victim, and could contribute to the accuracy of our ML models launched before the departure of the armament.

### 7.2.3/ METHODOLOGY

This article proposes a new methodology based on ML techniques to predict the mortality of a victim and their need to be transported or not to an HF. For these tasks, we will compare the performance of four ML techniques, namely, XGBoost, LGBM, CNN, and TabNET.

To increase the impact of the predictors, we added 6 other variables by combining the existing ones. The first 2 were:

- Age group (AGG\_GROUP\_AGE). The age of the victims was grouped into 8 categories.
- Time between rings the phone until the alert is broadcast (AGG\_INT\_TM\_TOTAL). It is the sum of delay time to answer the phone, call duration, and delay time to diffuse the alert, see Fig 3.1.

Then we sort our dataset in ascending order by datetime and divide it into a training set (calls from 2015-2019) to build our models and testing set (calls from 2020) to validate the performance of the models. Finally, we add 4 more predictors:

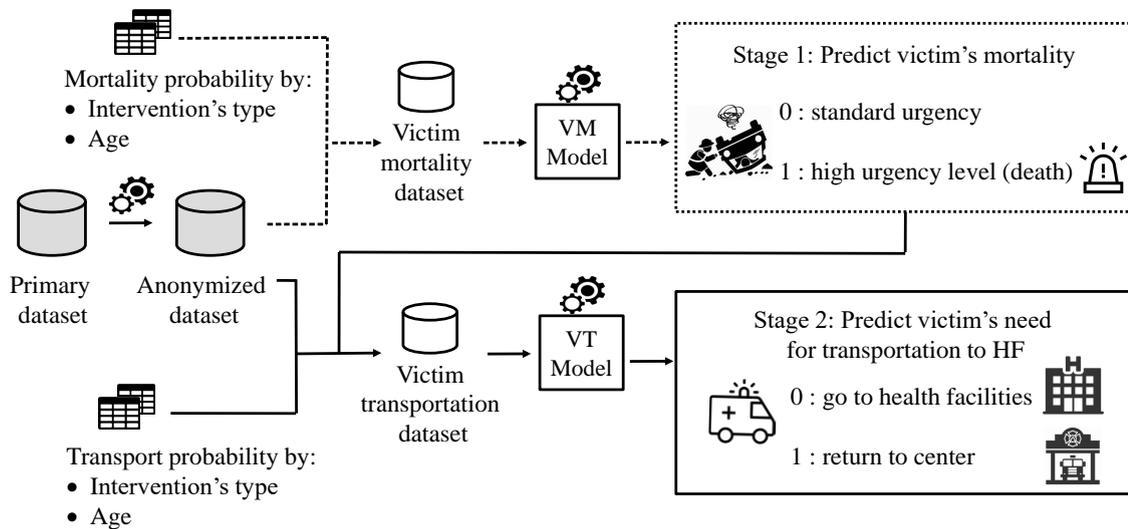


Figure 7.2: Proposed ML-based methodology developed in 2 stages. Stage 1 predicts the victim's mortality, which will be used as a feature in stage 2 when predicting the need to transport the victim to an HF.

- Probability of mortality by motive (AGG.MORT.PROBA.MOT). We took the training set and grouped the calls in which the victims died by intervention type (type, subtype, and motive of departure). The number of cases by intervention type is divided by the total number of deaths. This results in a probability, where the probabilities  $> 0$  are stored in a temporary table. The testing set samples are completed by looking for their probability in the temporary table. If the intervention's type is not found, the sample will receive a probability of zero, and the table will be updated for the next sample. Otherwise, the sample would take the current probability, and again the table would be updated. This follows the actual process, where the death of the victim is confirmed only when the firefighters return from their mission.
- Probability of mortality by age (AGG.MORT.PROBA.AGE). The training set data that resulted in deaths are grouped by motive of departure and AGG.GROUP.AGE. In this way, the number of cases per category is divided between the total deaths to generate a probability and create a temporary table with the values  $> 0$ . The assignment of probabilities to the testing set follows the same process of the variable described previously.
- Transport probability by motive (AGG.TRAN.PROBA.MOT). Its process is similar to the variable AGG.MORT.PROBA.MOT, with the difference that we exchanged deaths for non-transportation to HFs.
- Probability of transport by age (AGG.TRAN.PROBA.AGE). It follows the same process of AGG.MORT.PROBA.AGE, but with the "non-transportation to HFs" target.

The categorical variables such as: the grade and sex of the operator, the sex and city

of the victim, the center that attended the victim, the intervention's type, the weekday and the month were converted using the LabelEncoder method of Scikit-Learn, and the remaining variables that are numerical maintained their original scale.

Fig. 7.2 presents the interaction of our 2 final models developed in 2 stages as explained in the following:

**Stage 1: Prediction of victims' mortality.** In this stage, the objective is to predict whether or not the victim is at great risk of dying if they are not treated quickly. This is represented by the binary target variable VICTIM.MORTALITY (0: alive, 1: dead). The used dataset is composed of all the explanatory variables of the primary dataset and the new ones (AGG.GROUP\_AGE, AGG.INT\_TM\_TOTAL, AGG.MORT\_PROBA\_MOT, AGG.MORT\_PROBA\_AGE). Since there is an imbalance in the number of samples between classes 0 and 1 (see Subsection 7.2.2), we sought to compensate for the balance by using the Adaptive Synthetic sampling approach (ADASYN) and sample duplication, both only in the training set. In our preliminary experiments, duplication of samples in class 1 presented better results than ADASYN and, thus, duplication was used to construct the final models.

The specifications for the training process by technique is described in the following:

- LGBM: objective="binary", is\_unbalance=True, importance\_type="gain", boosting\_type="gbdt", and the search space was n\_estimators [50-1000], learning\_rate [0.001-0.8], max\_depth [1-10], and colsample\_by\_tree [0.5-1].
- XGBoost: objective="binary:logistic", boosting\_type="gbtree", and the tuned hyperparameters were n\_estimators [50-1000], learning\_rate [0.001-0.5], max\_depth [1-20], colsample\_by\_tree [0.2-1], and scale\_pos\_weight [20-60].
- CNN: we extracted 10% of the training samples for the validation set. In addition, 4 types of architectures were built using Keras library, their structure are presented in Table 7.2. The main hyperparameters were loss="BinaryCrossentropy", optimizer=Adam, epochs=1000, monitor="val\_loss", mode="min", patience=15, restore\_best\_weights=True. And the tuned hyperparameters were batch size [40-200], learning rate [0.0001-0.2], and the architecture's type [1,2,3,4].
- TabNET: we extracted 10% of the training test to be used as a validation set during the training process. The main specifications were optimizer="Adam", eval\_metric="logloss", max\_epochs=1000, and patience=15. The hyperparameters tuned were n\_d [3-10], n\_a [1-10], n\_steps [2-10], n\_independent [1-5], n\_shared [1-5], gamma [1-2], lr [0.001-0.5], and batch size [800-2000].

To evaluate the performances of our models, we followed the definitions in 2.3.2 to calculate the metrics: Accuracy (ACC), Balanced Accuracy (BACC), Area Under the Receiver

Table 7.2: Defined architectures for CNN with the number of neurons ( $nn$ ), pool\_size ( $s$ ), and dropout rate ( $r$ ) used.

Archi. 1	Archi. 2	Archi. 3	Archi. 4
Input	Input	Input	Input
BatchNormalization	BatchNormalization	BatchNormalization	BatchNormalization
Conv1D (nn=128)	Conv1D (nn=512)	Conv1D (nn=32)	Conv1D (nn=32)
MaxPooling1D (s=2)	MaxPooling1D (s=2)	MaxPooling (s=2)	MaxPooling (s=2)
Flatten	Flatten	Dropout (r=0.2)	Dropout (r=0.2)
Dense (nn=2)	Dense (nn=2)	Conv1D (nn=64)	Conv1D (nn=64)
Activation (sigmoid)	Activation (sigmoid)	MaxPooling (s=2)	MaxPooling (s=2)
		Flatten	Dropout (r=0.5)
		Dense (nn=2)	Conv1D (nn=128)
		Activation (sigmoid)	MaxPooling (s=2)
			Flatten
			Dense (nn=2)
			Activation (sigmoid)

Operating Characteristic Curve (AUC), Macro F1-Score (F1, which is the mean of all F1-score per class), and Recall and Precision for class 1. Each result was multiplied by 100 to show it in percentage.

Finally, to search hyperparameters for each technique, we used Bayesian optimization through the Hyperopt library with the Tree Parzen Estimator Suggest (tpe.suggest) and 500 iterations. The objective function employed to select the best victim mortality model (*VM model*) is a combination of ACC, BACC, F1, and Recall of class 1 in the form of  $\text{maximize}(ACC^2 \times Recall^2 \times BACC \times MF1)$ . This is because we wanted to keep the models with high ACC and Recall without losing the balance between classes due to the class imbalance problem.

**Stage 2: Prediction of victims' transportation.** This stage predicts whether the victim will need or not to be transported to an HF. The target is represented by a binary variable VICTIM\_TRANSPORTATION (0: needed, **1: did not need**). Given that most cases require (see Subsection 7.2.2) a transfer to an HF (negative class, 0), it would be more hard but advantageous to detect those calls that need the resources for less time (positive class, 1), thus, recognize that the armament will return faster to the center and there will be more resources available ready to attend other interventions. The dataset used in this stage included as an explanatory variable the predicted victim mortality since if the prediction indicates a death, there will be a greater probability of going to an HF to save the victim's life. Therefore, the engaged armament will be unavailable for a longer time. Other predictors considered are all those defined in stage 1, and AGG\_TRAN\_PROBA\_MOT and AGG\_TRAN\_PROBA\_AGE. Unlike the previous phase, no duplication or generation of artificial samples was applied since no improvement was observed in the preliminary tests. Finally, we used the same search space, model selection process, and metrics applied in stage 1 to choose the best victim transportation model (*VT Model*).

### 7.3/ RESULTS

Table 7.3 exhibits the metrics results considering the best model retrieved by ML technique, for each target, and for all samples from 2020. Fig. 7.3 illustrates the Pearson correlation coefficient ranging from -1 (negative correlation) to +1 (positive correlation). Fig. 7.4 shows the ROC curve of the best models by technique and target. And, Fig. 7.5 illustrates the 20 features with the highest impact by target and best XGBoost model. These features were identified by the XGBoost *Gain* feature importance algorithm, which is based on the relative contribution of each feature to improve the accuracy in the division of a branch.

Table 7.3: Results of the best models by ML technique for 2020.

ML Technique	Victim's Mortality (%)						Victim's Transportation (%)					
	ACC	BACC	AUC	F1	Recall	Precision	ACC	BACC	AUC	F1	Recall	Precision
LGBM	96.07	90.83	95.60	73	85	34	73.53	70.99	73.78	67	66	44
XGBoost	<b>96.44</b>	<b>90.86</b>	<b>96.04</b>	<b>75</b>	<b>85</b>	<b>36</b>	<b>73.62</b>	<b>72.63</b>	<b>78.91</b>	<b>68</b>	<b>71</b>	<b>44</b>
CNN	95.97	89.44	94.63	73	83	33	70.42	70.30	75.76	65	70	40
TabNET	95.39	90.24	94.97	71	85	30	70.98	70.26	74.17	65	69	41

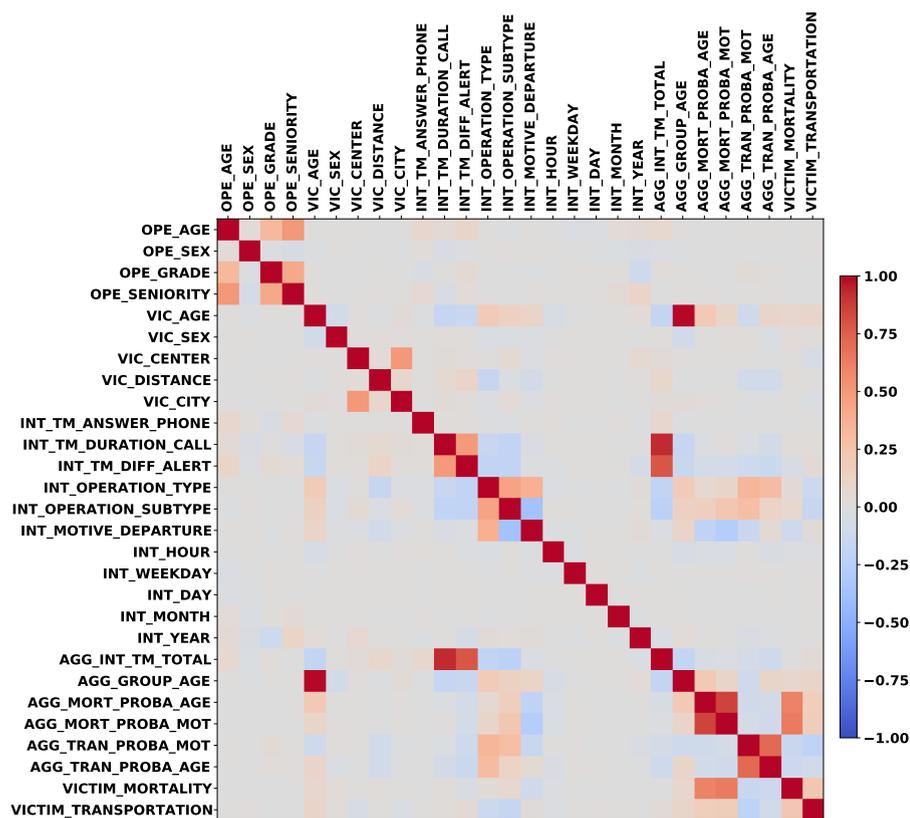


Figure 7.3: Pearson correlation between all predictors and targets.

From Table 7.3, one can notice that for ML models trained with original data, XGBoost consistently outperformed all the other models in both binary classification tasks, with results highlighted in bold. In general, one can notice that the developed models reached

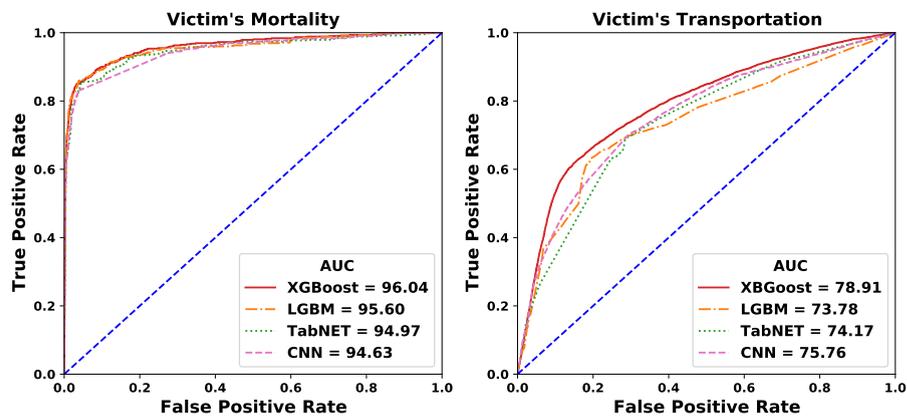


Figure 7.4: Model performances. ROC curve of the best classifiers by technique.

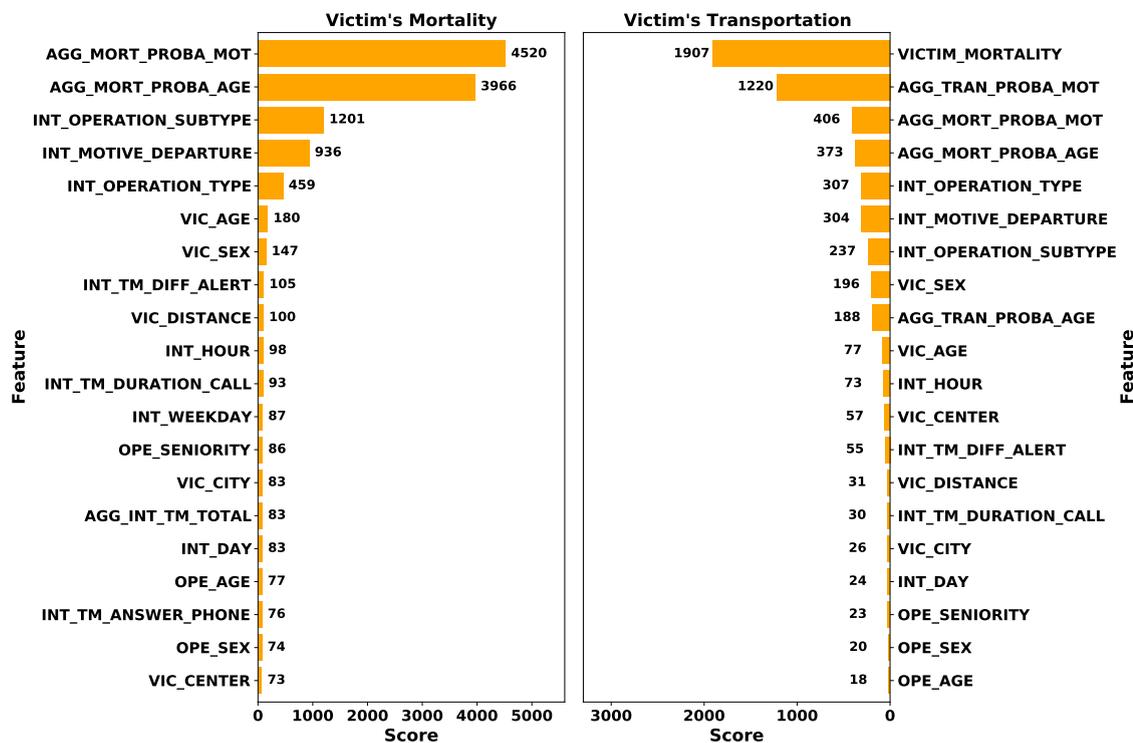


Figure 7.5: Feature importance from the best XGBoost models, considering the type *Gain* as score and the first 20 variables.

high scores when predicting victims' mortality, with ACC, BACC, and AUC scores higher than 88% getting to 96%. Also, as an imbalanced classification problem, for mortality, all models achieved good F1 scores of about 70% ~ 75%, and precision and recall for the minority class (victim will die – 1) of about 30% ~ 36% and 80% ~ 85%, respectively. On the other hand, the results for classifying if the ambulance will return to the EMS center (positive class – 1) or will transport the victim to HFs (negative class – 0) were not as good as for the victims' mortality target. In this case, all models achieved intermediate scores of about 70% ~ 78% for ACC, BACC, and AUC. Besides, the F1 metric decreased

to about 65% ~ 68% with lower recall values ranging from 66% ~ 71% and higher precision scores 40% ~ 44% in comparison with predicting mortality.

What is more, Fig. 7.4 allow us to visualize the model performances for both targets, where the fraction of positive cases correctly predicted (TPR) keeps high and the fraction of negative cases incorrectly predicted is low. From here, we can also identify that the performance of the predictive mortality model is more efficient than the transport model, which suggests that the predictions of the first model (victim's mortality) are very good but it is necessary to add other variables (e.g., telephone call recordings or transcripts, victim's economic status, social status, health history, etc.) to improve the efficiency of the second model (victim's transportation) that directly impacts the consumption of operational resources.

From Fig. 7.3, one can notice that few of the initial variables (VIC\_SEX, INT\_OPERATION\_TYPE, INT\_OPERATION\_SUBTYPE, and INT\_MOTIVE\_DEPARTURE) have a slight linear correlation with both objectives. However, by combining some of these and generating the aggregates (AGG prefix), we obtain variables with higher correlation (AGG\_MORT\_PROBA\_AGE, AGG\_MORT\_PROBA\_MOT, AGG\_TRAN\_PROBA\_AGE, and AGG\_TRAN\_PROBA\_MOT).

From Fig. 7.5, we can confirm that the aggregate variables generated from probabilities and the type of intervention (type, subtype, and motive) have a great impact on the creation of both models. In the case of victim's mortality, age and sex are more important than the alert diffusion time and duration of the call. In addition to that, we discovered that the hour and day of the week also have an impact on the model. In the case of non-transportation of the victim to an HF, the greatest impact is generated by the mortality of the victim and, to a lesser degree, the sex and age of the victim, the hour of the call, and the center from which the ambulance departed. Moreover, operators' personal data showed lower importance in both models.

## 7.4/ DISCUSSION AND RELATED WORK

At the time of this study and when reviewing the related works, we found that the authors in [79] identified that a shorter response time is associated with a higher survival rate. Thus, some EMS have been exploiting their data collected for years with ML techniques, with the firm intention of reinforcing the immediate response to an emergency and strategically deploying their resources [187, 139, 188, 160, 120]. Following with these general objectives but with a different approach, the work of the present chapter proposes a new methodology to predict the victim's mortality with a first model. Since this prediction is done before dispatching ambulances, an EMS could activate proactive decisions to mobilize their armament and provide quicker response times. In addition, we also propose

to use the prediction of the victim's mortality as an input to a second model (cf. Fig. 7.2), which predicts if the ambulance will return to the EMS center and, as consequence, the crew will be engaged for less time and more resources will be available.

To evaluate our methodology, we compared the performance of 4 recognized techniques (XGBoost, LGBM, TabNet, and CNN), where the best accuracy was obtained by XGBoost for both objectives. We identify that although the victim's mortality prediction model is efficient, more features are still needed to improve the predictive capacity of the second model. In addition, we determined that for the prediction of victim's mortality the variables AGG.MORT.PROBA.MOT, AGG.MORT.PROBA.AGE, INT.OPERATION.SUBTYPE, INT.MOTIVE.DEPARTURE, INT.OPERATION.TYPE, VIC.AGE, VIC.SEX were the most influential; and in the case of the prediction of victim's transportation, the most influential were VICTIM.MORTALITY, AGG.TRAN.PROBA.MOT, AGG.MORT.PROBA.MOT, AGG.MORT.PROBA.AGE, INT.OPERATION.TYPE, INT.MOTIVE.DEPARTURE, and INT.OPERATION.SUBTYPE.

Finally, although the collection of medical data allows investigations to propose improved ML-based decision-support tools, there is a problem with the disclosure of personal and sensitive information. In our case, there are two entities we are concerned with, namely, call center *operators* and *victims* with regard to privacy. What is more, with the raw dataset containing direct identifiers (e.g., names), one straightforward question as "*Is there any operator linked with an increased ratio of victims' death?*" can be easily computed, which compromises the operators' privacy and can lead to social and/or economical damages. Similarly, one can easily access the reason for the intervention (e.g., cardiac arrest) and use this information to compromise the victims' privacy through discrimination in health insurance, for example. Besides, even by excluding direct identifiers, both victims' and operators' identities are still at risk of being retrieved. For this reason, the present study was complemented with the application and analysis of state-of-the-art privacy models, namely,  $K$ -anonymity [28] and Differential Privacy [36, 69], implemented in collaboration with Héber Hwang Arcolezi (member of the AND team). This sanitization process is out of the scope of this chapter but the collaborative work was published in the IEEE Transactions on Industrial Informatics journal [190].

## 7.5/ CONCLUSION

The present study developed a new methodology to predict the mortality of victims and their need for transport to health centers, based on machine learning techniques. For this purpose, we used as input retrospective data recorded from 2015 to 2020 by SDIS25. These data consider information on calls, operators, centers, victims, and aggregate variables. To build the models, we applied four ML techniques, namely, XGBoost, LGBM, TabNet, and CNN.

The results demonstrated that it is possible to predict victim mortality with high accuracy (XGBoost, 96.44%). Furthermore, we validated that this resulting variable is a highly influential feature in the prediction of victim transport to an HF. This second predictive model (victim transportation) showed an accuracy of 73.62% also with XGBoost. Although its performance was lower than the first model, it allowed us to identify the variables with the greatest impact so far.

Thus, the developed methodology would allow an EMS to predict if the victim is at serious risk or if it is a mild case, and if the compromised engine will return to its center or perform a victim transport to any HF, i.e., it will be kept longer busy and less resources will be available. All this would help a better deployment of EMS resources.

Finally, our results were published in [190], where we considered that the ML models could be subject to attacks and compromise the operator's and victims' privacy.



# PREDICTING INTERVENTION PEAKS DUE TO RARE EVENTS

In the previous chapter, we developed a methodology based on ML techniques to predict victim's mortality and their need for transportation to health facilities. In the present chapter, we develop and evaluate the performance of several approaches for forecasting the operational overload of firefighters generated in a short period of time and due to rare events.

## 8.1/ INTRODUCTION

The missions and the activity of fire departments change from one country to another: in some countries, fire departments are only in charge of extinguishing fires, while in others they are also in charge of rescuing people, whether it is urgent or not. In regions such as Doubs, in France, fire activities represent only almost 10% of all their interventions, and they may be called out for road accidents, floods, respiratory ailments, gastroenteritis, or even wasp nests. In Western countries with an ageing population, the share of personal assistance is constantly increasing and the repeated economic crises lead to budget cuts.

Thus, in countries where fire brigades activity encompasses rescuing people, the number of interventions has been steadily increasing for several years now, getting worse with the COVID-19 pandemic. In this context, ensuring rapid and efficient interventions becomes a major challenge for many brigades and, therefore, it is essential to plan interventions in advance. Fortunately, the vast majority of brigades have accumulated an important mass of data related to their past interventions. It is, therefore, interesting to try to use these data and the recent advances in Machine Learning to try out intervention prediction.

It is reasonable to think that such predictions are possible, as the reasons for these interventions are to some extent deterministic. Forest fires occur more frequently in dry, hot weather than in wet, rainy weather; floods follow heavy rains; domestic and work acci-

dents occur mostly in the middle of the day, and they are rare at 3am; falls on the ice do not occur in summer and drowning in outdoor pools does not occur in winter.

As can be seen from the above examples, weather conditions have an undeniable impact on human activity and the accidents it causes, and therefore on the activity of firefighters. In addition, most national weather prediction services provide online services or APIs to retrieve various physical quantities useful for weather knowledge. These quantities, which are internationally codified and are mandatory measurements, include temperature, pressure, wind direction and strength, dew point, or hydrometry, for a set of stations spread over the entire territory.

However, while these physical quantities are useful in predicting, to some extent, the weather, they are only partially effective in predicting firefighters interventions, their types, and intensities. Knowing that a thunderstorm event is coming does not accurately predict the extent to which firefighters activity will be affected. More precisely, by coupling the history of weather conditions with the history of interventions, we can see that while some storm-type weather conditions clearly lead to peaks in intervention, others, on the contrary, go somewhat unnoticed. In other words, the simple quantitative value of the physical quantities to be measured does not alone make it possible to accurately predict the occurrence of periods of heavy intervention.

For this reason, in the present study, we will experiment with the textual content of weather bulletins, since they inform about the risk of possible snowfall, storms, floods, etc., and describe useful advice. This textual content of bulletins is valuable and under-explored, because phrases like *“it is advisable to unplug electrical appliances”* and *“it is strongly advised not to walk along the coast”* are rich in information for the prediction of heating-related interventions or emergency rescue due to rare events. It is here, where automatic natural language processing tools would allow us to understand the link between such statements and peak intervention periods.

### 8.1.1/ OBJECTIVES

With these elements in mind, the present work performs a detailed study of various ML models developed for the prediction of intervention peaks due to rare weather events and identifies the remarkable impact of NLP models and weather bulletin texts. The rarity of events such as flood or heating events makes them difficult to predict due to the small amount of data available and, as we will see in the remainder, these events can be the source of extremely high numbers of interventions. The predictions are made for 4 categories of interventions, namely, Emergency Person Rescue, Total Person Rescue, interventions related to Heating, and interventions related to Storm/Flood. The data was collected from 2012-2020. They are composed of interventions of the Fire Department

of Doubs (SDIS25), located in the northeast of France, quantitative weather variables and texts of weather bulletins and their vigilance levels from Météo-France. In this way, SDIS25 or other Emergency Medical Services (EMS), in general, could better recognize periods with high workload generated by rare events, through text processing of weather bulletins, and strategically prepare the appropriate personnel and armament to deal with the crisis so that no service disruptions occur and more lives can be saved. In summary, our objectives in this work are described in the following.

- a) *Analysis and prediction of interventions using basic univariate time series models.* This allows to recognize the seasonality of rescue-type interventions and identify the lack of recognition of incidents triggered by rare events. We compared 3 basic models per category to predict the number of interventions per hour. Each model makes predictions equal to the mean, equal to the last known value, and equal to the mean per hour.
- b) *Analysis and prediction of interventions using multivariate time series models.* This allows to complement and identify the trend and the seasonality of the signal by adding qualitative and quantitative variables (variables of vigilance, weather, and calendar). We developed 7 models per category with the state-of-the-art Extreme Gradient Boosting (XGBoost) technique, in which the features are combined to measure their impact on the prediction.
- c) *Analysis and prediction of intervention peaks using multilabel classification models based on decision trees and tabular data.* The problem is restated as a multilabel classification task for the 4 categories, using the variables of the previous model. Furthermore, we considered the XGBoost and Random Forest (RF) techniques, well-known in the literature for their fast execution and robustness. This allows to determine the influence of the variables according to the new perspective.
- d) *Analysis and prediction of intervention peaks using multilabel classification models based on NLP techniques and text from meteorological bulletins.* We developed and compared models based on ancient (LSTM and CNN) and modern (FlauBERT and CamemBERT transformers) neural networks. This allows to significantly improve the forecast of the peaks compared to the previous models with decision trees and tabular data. Thus, it demonstrates that it is possible to extract much more information from public weather bulletins using NLP techniques.

## 8.2/ MATERIALS AND METHODS

In this section, we describe the collection process of internal variables such as the list of interventions attended by SDIS25, weather bulletins for the region of Doubs and three

neighboring departments (Territory of Belfort, Haute-Saône, and Jura), and qualitative and quantitative features (calendar and meteorological variables). In addition, we will present the structuring of weather bulletins and interventions as a classification problem. Finally, we will briefly describe the NLP techniques applied to the solution of our problem.

### 8.2.1/ DATA ACQUISITION

We collected all the interventions of SDIS25 from 2012 to 2020. The number of interventions were grouped by slots of 1 hour and per type of intervention (Emergency Person Rescue, Total Person Rescue, interventions related to Heating, and interventions related to Storm and Flood). A short statistical description of average (Avg.), standard deviation (Std.), minimum (Min.), and maximum (Max.) of the number of interventions per hour and type is provided in Table 8.1. And, Fig. 8.1 plots the curves for each type and for the first days of 2012.

Table 8.1: Statistical description of the number of interventions per hour

Type of intervention	Avg.	Std.	Min.	Max.
Emergency person rescue	3.56	2.46	0	20
Total person rescue	7.25	4.53	0	30
Heating	0.32	0.62	0	6
Storm/Flood	0.26	1.17	0	82

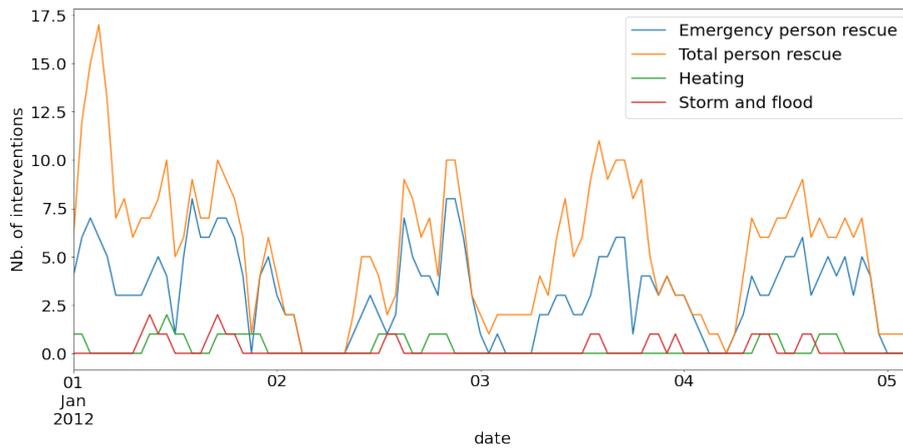


Figure 8.1: Number of interventions per type, early 2012

On the one hand, Table 8.1 shows a high average for both types of people rescue, which demonstrates that these are very common in the work load of the fire department. What is more, in Fig. 8.1, we can observe seasonal patterns, a clearly 24-hour cycle, that can be better recognized with the integration of calendar variables. On the other hand, the Heating type only presents a maximum of 6 simultaneous interventions in an hour, which demonstrates that this type is quite uncommon. The category Storm and Flood presents

```

<Titre name="Conséquences_possibles">
<Paragraphe>
<Intitule>Vent/Orange</Intitule>
<Texte>
* Des coupures d'électricité_et_de_téléphone_peuvent_affecter_les_réseaux_de_distribution
pendant_des_durées_relativement_importantes.
</Texte>
<Texte>
* Les_toitures_et_les_cheminées_peuvent_être_endommagées.
</Texte>
<Texte>
* Des_branches_d'arbre_risquent_de_se_rompre.- Les véhicules peuvent être déportés.
</Texte>
<Texte>
* La circulation routière peut être perturbée, en particulier sur le réseau secondaire
en zone forestière.
</Texte>
<Texte>
* Quelques dégâts peuvent affecter les réseaux de distribution d'électricité_et_de_téléphone.
</Texte>

```

Figure 8.2: Example of possible consequences section (in French)

a low average, but a high maximum number of interventions, i.e., an operational overload at specific times due to rare events such as storms and floods. The rarity of storms or heating events will make their predictions problematic in the absence of additional information, which already shows the interest of considering weather-type variables.

This is the reason why we have retrieved historical meteorological data from the Météo-France site (essential SYNOP data [209]). The three closest meteorological stations selected are those of Nancy-Ochey (latitude 48.581000, longitude 5.959833), Dijon-Longvic (47.267833; 5.088333), and Basel-Mulhouse (47.614333; 7.510000). The data recovered in this way are temperature (degrees Celsius), pressure (Pa), pressure variation (Pa per hour), barometric trend (categorical), humidity (percent), dew point, last hour rainfall (millimeter), last three hours rainfall (millimeter), mean wind speed (10 min., m/s), mean wind direction (10 min., m/s), gusts over a period (m/s), horizontal visibility (m), and current weather (categorical, 100 possible values).

These data were supplemented by (textual) vigilance alert bulletins from Météo-France [210]. They are XML files containing the type of vigilance (heatwave, extreme cold, snow or ice, thunderstorms, strong winds), the beginning and end of the vigilance period, the level of the alert (green, orange or red), a detailed description of the risk (including the locations impacted, the conditions to be expected...), as well as a set of very detailed advice to users. An example of such files is provided in Fig. 8.2. Finally, we added calendar-type variables, namely the time of the niche considered, the day in the week, the day in the month, the month in the year, and the year considered.

### 8.2.2/ DATA PREPROCESSING

Since 2011, 12218 weather bulletins have been produced, including 1054 for the North-East region of France (known as CMIRNE). This region consists of 18 departments,

Table 8.2: Example of a multilabelling of vigilance texts

Text	Emergency Person Rescue	Total Person Rescue	Heating	Storm/ Flood
Les températures sont déjà négatives aujourd'hui mercredi. A 15h les ... une vigilance particulière notamment pour les personnes sensibles ou exposées.	1	0	1	1
Période de grand froid; moins intense qu'en 1985; mais nécessitant toutefois ... températures sous abris observées s'échelonnent entre -1 et -4 degrés.	1	0	1	1

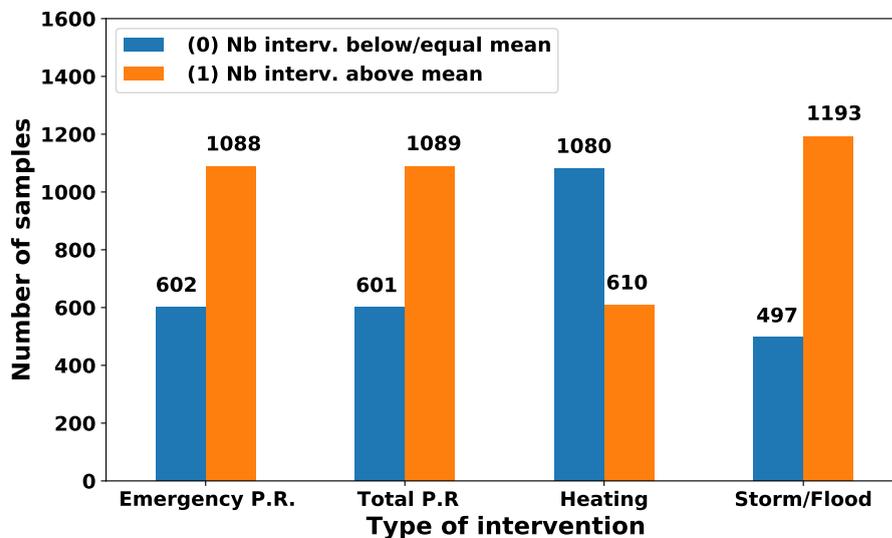


Figure 8.3: Number of samples for each category

but we are only interested in Doubs and its three neighboring departments (Territory of Belfort, Haute-Saône and Jura), in a first time.

The set of texts available may seem small at first glance, but on the one hand, each bulletin covers a number of departments and has several sections such as location, description, qualification of the phenomenon, new facts, current situation, expected evolution, possible consequences, and behavioural advice, and each section is made up of several long and detailed sentences. The result is a corpus of 76333 characters

These bulletins are then cut out by section. Over the entire period of vigilance, the average number of interventions is calculated for each of the four types considered, and a class 0 or 1 is introduced depending on whether this number is above the average number of interventions according to the type, for all period 2012-2020. We thus associate 4 binary labels for each text in each section of each bulletin, as shown in Table 8.2. Through this encoding, we became interested in the question **Q1**: *in the context of this vigilance event, will there be an increase in the number of interventions, greater than the average, related to rescuing people, heating, and storm/flood?*

In Fig. 8.3, one can observe in more detail the number of samples (texts) with value 0,

where the number of interventions was below or equal to the mean; and value 1, in the opposite case. Furthermore, we find an unequal distribution in each binary class of each category, that might be bias the prediction models.

### 8.2.3/ MODELING

In the following, we will describe the implementations used in this study for natural language processing.

1. **LSTM.** Given the sequential nature of texts, RNN and LSTM are fairly common models in NLP literature. However, since RNNs present gradient vanishing problems and LSTM overcome them, we will build models based on LSTM layers.

We used the *Keras* and *TensorFlow* libraries to initially built several architectures with 1, 2, and 3 LSTM layers, different numbers of neurons, and constant values for learning rate and batch size. Then, we evaluated their performances and selected the 3 architectures that gave us the first best results, these are the ones described in Table 8.3. Finally, to obtain the best LSTM model, we intensified the search with the 3 selected architectures, varying the learning rate and batch size parameters. For this, we used the HyperOpt library [61] with 100 iterations and the Tree Parzen Estimator suggest (tpe.suggest) algorithm, which models 2 density functions instead of the probability of an observation to estimate the expected improvement of a new configuration.

Before the texts enter the neural network, the data was preprocessed as follows:

- Words with flexible endings were eliminated from the French texts and their base forms were returned, using the *SpaCy* library [206]. This process is known as *lemmatization*. Unlike a pre-trained model in a large vocabulary, our LSTM models use the texts of the weather bulletins as a base, hence, there is the need to bring the words to their base form to identify a greater risk of a possible event. For example, the words "*inondations*" and "*inondables*" share the same root and refer to "*floods*".
- Using the *nltk* library [42], French stopwords were eliminated, i.e., words such as articles, pronouns, prepositions, auxiliary verbs, among others, which do not have a big impact on our predictions were deleted.
- From the *Keras* library, we used the *Tokenizer* function to create a dictionary of words based on their frequency. Then, we reduced the dictionary to the 1000 most repeated words and for each text we generated vectors with integer values, that are the indexes in the dictionary. Finally, they were padded to the same length and entered the models described in 8.3.

Table 8.3: Defined architectures for LSTM

Archi. 1	Archi. 2	Archi. 3
Input(200)	Input(200)	Input(200)
Embedding(100)	Embedding(100)	Embedding(200)
LSTM(128)	LSTM(1000)	LSTM(128)
Dense(256, ReLU)	Dense(2000, ReLU)	Dense(256, ReLU)
Dropout(0.5)	Dropout(0.5)	Dropout(0.2)
Dense(4, Sigmoid)	Dense(4, Sigmoid)	LSTM(512)
		Dense(1024, ReLU)
		Dropout(0.2)
		Dense(4, Sigmoid)

Table 8.4: Defined architectures for CNN

Archi. 1	Archi. 2	Archi. 3
Input(200)	Input(200)	Input(200)
Embedding(200)	Embedding(200)	Embedding(200)
Conv1D(128, 3, ReLU)	Conv1D(16, 3, ReLU)	Conv1D(256, 3, ReLU)
MaxPooling1D(2)	Dropout(0.2)	Dropout(0.2)
Flatten()	MaxPooling1D(2)	MaxPooling1D(4)
Dense(4, Sigmoid)	Conv1D(32, 3, ReLU)	Conv1D(300, 4, ReLU)
	Dropout(0.5)	Dropout(0.2)
	MaxPooling1D(2)	MaxPooling1D(4)
	Conv1D(64, 3, ReLU)	Conv1D(360, 4, ReLU)
	MaxPooling1D(2)	Dropout(0.5)
	Flatten()	MaxPooling1D(4)
	Dense(4, Sigmoid)	Flatten
		Dense(400, ReLU)
		Dropout(0.2)
		Dense(4, Sigmoid)

- CNN.** CNN has been a game changer in the field of image analysis. They also have shown some interesting results in NLP [162, 86]. Indeed, texts can be represented as an array of vectors, just like images can be represented as an array of pixel values. Here, we deal with one dimensional convolution, but the principles remain the same: we still want to find patterns in the sequence, which become more complex with each added convolutional layer. In order to associate each word with a specific vector, we found in the literature different words embedding techniques that are context-sensitive such as Glove [73] and Word2Vec [65].

In our case, we tested with architectures based on CNN layers. In this way, similar to LSTM, we developed several architectures and selected three, which are described in Table 8.4. These ones will be used to perform an exhaustive search, varying the learning rate and batch size with the HyperOpt library [61]. In addition, the text preprocessing performed before entering the CNN is also the applied for LSTM.

- Transformers.** Transformers models provide a solution by processing the whole sequence all at once and by using the attention mechanism. This mechanism allows capturing different types of relationships between tokens. A Transformer is built based on an encoder and a decoder, each of them consisting of a stack of attention and dense layers. An example of this is BERT, which is a well-known word embedding technique. Word embeddings are able to capture the context, semantic,

and syntactic similarity (gender, synonyms, ...) of a word by reducing the dimension. For instance, BERT will represent the words "remarkable" and "admirable" as relatively closely spaced vectors in the vector space.

In our experiments, we run the base implementation of CamemBERT [133] and FlauBERT [130] from the HuggingFace's transformers library in Pytorch. Both uses the original architecture of BERT base (12 layers, 768 hidden dimensions, 12 attention heads, and 110M parameters).

## 8.3/ RESULTS

According to the four objectives defined in 8.1.1, first, we present the scores of the univariate time series models for the prediction of the number of interventions (objective a). The models perform constant predictions equal to the mean, predictions corresponding to the persistence model (we predict the last known values), and predictions corresponding to the mean per hour. Next, we develop multivariate time series models (objective b), based on one of the most powerful learning machine tools available today: XGBoost [81]. The data used are composed of calendar features (day in the week, month, year...), quantitative meteorological variables, corresponding to the three meteorological stations closest to the Doubs, which makes it possible to measure the extent to which predictions of fire-fighting interventions can be made by taking only numerical data from meteorological information, and only vigilance levels from bulletins as indicators. Then, we reframe the problem as a multilabel classification task (objective c) in order to analyze the impact of the tabular data previously described but with a new perspective that allows us to recognize the periods with intervention peaks and not the number of interventions. For this purpose, we compared the XGBoost classifier and the one from a technique also recognized for its speed and robustness, called Random Forest [24]. Finally, we replaced the tabular data with the meteorological texts of the bulletins and applied NLP techniques to study how valuable these data can be (objective d).

### 8.3.1/ PREDICTION OF INTERVENTIONS USING BASIC UNIVARIATE TIME SERIES MODELS

Scores corresponding to constant predictions are given in this sub-section. The results obtained are shown in the Table 8.5 for the Emergency Person Rescue, Table 8.6 for the Total Person Rescue, Table 8.7 for the Heating related interventions, and Table 8.8 for the Storm/Flood ones. The metrics considered were MAE and RMSE. As can be seen, the average per hour does better than the average alone for personal assistance, which is understandable given the daily seasonality of human activity. As shown in Fig. 8.4, there

are fewer interventions at night than during the day because people are simply sleeping; similarly, because people eat at noon, there is a plateau at that time. This improvement is not found for interventions such as heating or storm, as these are only minimally related to human activity: a storm can cause damage both day and night.

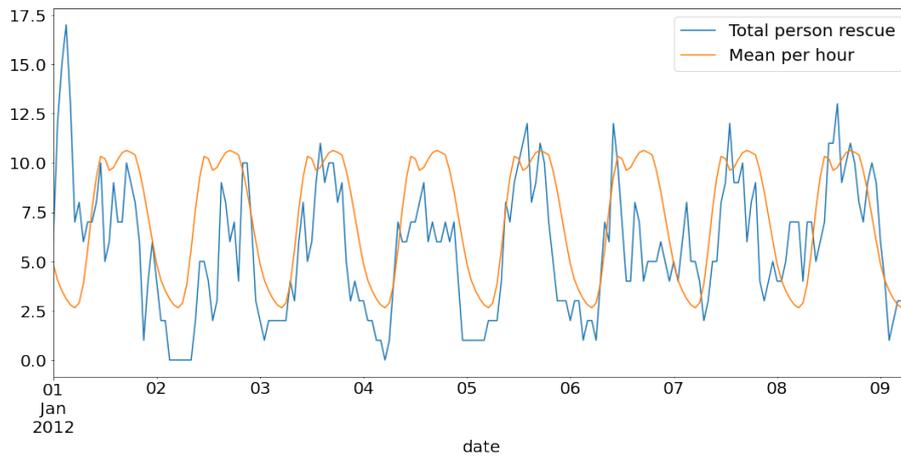


Figure 8.4: Total Person Rescue interventions and its mean per hour series, early 2012

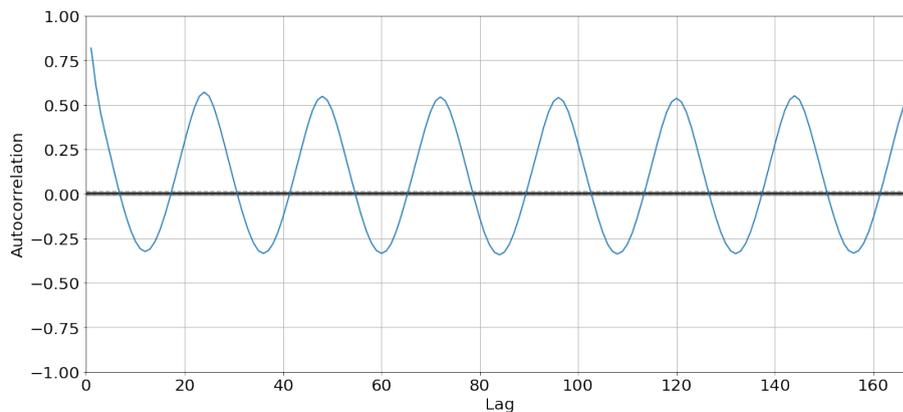


Figure 8.5: Auto-correlation graph for Total Person Rescue

One might a priori be astonished that the simplistic persistence model does better than the average per hour, which seems more evolved. However, this is well explained by the scarcity of interventions of the heating or storm type: if most of the time, there are 0 interventions per hour, then the probability that the slots at time  $t$  and time  $t+1$  both have no intervention is high. Thus, replicating what happened at time  $t$  as a prediction of what will happen at time  $t+1$  is a winning strategy when events are rare. The good success of this model in the case of rescue-type interventions is again explained by the strong daily seasonality of these interventions: between one hour and the next, the number of interventions is close, when it is very different between time  $t$  and  $t+12$  hours, i.e., there is a strong correlation between the time series of the rescue-type and the time series shifted by one hour, presented in Fig. 8.5.

Table 8.5: Emergency Person Rescue prediction scores

<b>Method</b>	<b>MAE</b>	<b>RMSE</b>
Mean	1.9856	2.4554
Persistence	1.3504	1.8292
Mean / hour	1.6485	2.1160

Table 8.6: Total Person Rescue prediction scores

<b>Method</b>	<b>MAE</b>	<b>RMSE</b>
Mean	3.6939	4.5267
Persistence	2.0571	2.7298
Mean / hour	2.5843	3.3901

Table 8.7: Heating prediction scores

<b>Method</b>	<b>MAE</b>	<b>RMSE</b>
Mean	0.4829	0.6240
Persistence	0.1737	0.4490
Mean / hour	0.4641	0.6127

Table 8.8: Storm and Flood prediction scores

<b>Method</b>	<b>MAE</b>	<b>RMSE</b>
Mean	0.4340	1.1742
Persistence	0.1749	0.5901
Mean / hour	0.4225	1.1699

### 8.3.2/ PREDICTION OF INTERVENTIONS USING MULTIVARIATE TIME SERIES MODELS

The first features to be taken into consideration are obviously the calendar data, which fully condition human activity. The time of day captures the daily seasonality, with a drop-in activity at night, a maximum of activity in the late morning and in the afternoon, with a plateau at noon. For various reasons, the day in the week also has its importance (e.g., weekend, a day without school for children). In the same way, the day in the year makes it possible to recover particular periods such as the summer, the winter vacations, even particular days (national holiday, Christmas, and new year's Day). These particularities are also contained, but in a less continuous way, in the month in the year. Finally, the year is important because it allows us to model the general upward trend, for the various reasons mentioned above (population aging, population growth, etc). For accidents related to heating (e.g., chimney fire), storms and floods, or even emergency rescue, it is reasonable to think that meteorology is important and that predictions should be improved by adding variables describing it. For example, between two national holidays, if the first one is rainy when the second one takes place under a radiant sun during a heatwave, this will probably have an impact on firefighters' departs.

Table 8.9: Prediction scores using XGBoost, Emergency Person Rescue case

<b>Features</b>	<b>MAE</b>	<b>RMSE</b>
Calendar	1.523	1.961
Weather	1.825	2.307
Vigilance	2.021	2.493
Weather + Calendar	1.586	2.049
Calendar + Vigilance	1.400	1.881
Weather + Vigilance	2.186	2.809
All	1.940	2.632

Table 8.10: Prediction scores using XGBoost, Total Person Rescue case

<b>Features</b>	<b>MAE</b>	<b>RMSE</b>
Calendar	2.223	2.917
Weather	3.281	4.138
Vigilance	4.117	5.138
Weather + Calendar	2.375	3.118
Calendar + Vigilance	2.279	2.998
Weather + Vigilance	4.293	5.376
All	3.385	4.445

Table 8.11: Prediction scores using XGBoost, Heating case

<b>Features</b>	<b>MAE</b>	<b>RMSE</b>
Calendar	0.413	0.558
Weather	0.502	0.663
Vigilance	0.510	0.659
Weather + Calendar	0.475	0.636
Calendar + Vigilance	0.319	0.495
Weather + Vigilance	0.565	0.825
All	0.533	0.804

Table 8.12: Prediction scores using XGBoost, Storm/Flood case

<b>Features</b>	<b>MAE</b>	<b>RMSE</b>
Calendar	0.325	0.734
Weather	0.386	0.867
Vigilance	1.729	4.959
Weather + Calendar	0.370	0.863
Calendar + Vigilance	0.793	2.453
Weather + Vigilance	1.212	2.565
All	1.161	2.978

The purpose of this section is to see if this meteorological influence can be recovered without using NLP. We will therefore compare the quality of the predictions for the 4 types of intervention, with or without the temporal data (calendar), with or without the color of the weather alert (vigilance), and with or without the quantitative data from the Météo-France site (weather) such as temperature, pressure, wind, etc. To do so, we will randomly

separate our data set between learning (80%) and test (20%), and we will look at the MAE and RMSE scores on the test set, once the learning is completed. This learning will be done with XGBoost (Poisson regression, max depth of 6), with 20% of the learning set used for validation, and an early stopping criterion of 10 steps.

From Table 8.9, the case of emergency person rescue, calendar data is obviously what appears to be most important, but that predictions can be improved by adding the color of the weather alert bulletin. These bulletins are basically too coarse a data set to be useful in predicting these types of interventions on their own. As such, quantitative meteorological data do a little better but are far from what is obtained with calendar data. However, the best result is obtained by mixing calendar data with vigilance bulletins, when the addition of quantitative weather variables systematically lowers the quality of the predictions. We noted that, among baselines (subsection 8.3.1), an autoregressive component is fundamental to this prediction problem, and it would improve the scores of the Table 8.9, i.e., among the interventions between  $t$  and  $t+1$ , there are the new interventions that appear after  $t$ , and those that appeared previously but are still in use.

The same lessons can be learned from total interventions in a more pronounced way, see Table 8.10. This time, the best result is achieved by calendar data alone, and the addition of weather variables only pushes XGBoost to overfit. These total interventions include, in addition to the emergency rescue, accidents on the public highway, and non-emergency rescue: person trapped in an elevator or locked in a balcony, wasp's nest, etc. These interventions are, by nature, much more difficult to predict. And the impact of temperature or atmospheric pressure is certainly much less on this type of intervention than, for example, knowing whether it is the middle of the day or the middle of the night. One can then naturally wonder if adding textual information on the weather could improve such results.

In the case of interventions related to heating, this time the best score is obtained by linking the calendar data to the vigilance bulletins, as shown in Table 8.11. This is understandable, given the nature of the interventions (chimney fire, electrical heating appliance fire...). Here again, quantitative meteorological information does not provide the same benefit as the color of the vigilance bulletin, for the same reasons as previously mentioned. Conversely, for events such as storms and floods, if the calendar alone produces the best results, it is closely followed by the combination of calendar and meteorological data (cf. Table 8.12). Here, in a counter-intuitive way, the vigilance bulletins greatly reduce performance. All this can be explained by noting first of all that floods do not occur at any time of the year in the Doubs, but mainly in winter. They are very localized in time, when the bulletins of vigilance usually extend over a fairly long period. Finally, such events follow a strong fall of water, a fact that is found in weather variables.

Finally, the case of heating-type interventions shows that by adding information about the

weather, predictions can be improved. The case of storms and floods, on the other hand, shows that quantitative information (temperature, pressure...) and qualitative information (risk of storm, flooding...) do not provide enough information on the weather: the latter has an obvious impact on the interventions, but these variables do not improve the predictions. And the Emergency and Total People Rescue confirm that these variables describe the weather forecast too crudely, which explains why we are interested in the textual content of the vigilance bulletins.

### 8.3.3/ PREDICTION OF INTERVENTION PEAKS USING MULTILABEL CLASSIFICATION MODELS BASED ON DECISION TREES AND TABULAR DATA

In this section, the problem is restated as a multilabel classification, where the labels are the 4 categories (Emergency Person Rescue, Total Person Rescue, Heating, and Storm/Flood) and the possible classes are 0 and 1 as it is described in 8.2.2. This is to identify the peak periods of intervention concerning each category. The quantitative and qualitative variables described in subsection 8.3.2 are the inputs for the models developed in this section, where each sample of the dataset represents one hour, an illustration is showed in Figure 8.6. In addition, the created models are the results of various combinations of the variables to observe the influence of them but with a classification approach. These models will be the baselines to be overcome by the NLP techniques in the following subsection.

Hour	Day	Weekday	...	Emergency Person Rescue	Total Person Rescue	Heating	Storm/Flood
...	...	...	...	...	...	...	...
0	4	2	...	1	0	1	0
1	4	2	...	1	0	0	1
...	...	...	...	...	...	...	...
10	22	6	...	0	1	1	0
11	22	6	...	1	1	0	0
...	...	...	...	...	...	...	...

Figure 8.6: Illustration example of the tabular data for multilabel classification, considering calendar variables

Before applying NLP techniques, we sought to develop and compare models with simpler techniques, which do not require a greater consumption of resources such as those based on decision trees. These techniques have demonstrated their remarkable effectiveness in the literature with tabular data. We chose XGBoost as a representative of boosting algorithms, which seek to reduce model variance, and Random Forest, as a representative of bagging algorithms, which seek to reduce bias.

Table 8.13: Hyperparameters search space and the best configuration for XGBoost and Random Forest multilabel models

Method	Search Space	Best configuration						All
		Calendar	Weather	Vigilance	Weather +	Calendar +	Weather +	
XGBoost	n_estimators: [50-200]	195	159	193	132	132	86	112
	learning_rate: [0.001-0.8]	0.49	0.14	0.39	0.17	0.40	0.60	0.69
	max_depth: [1-100]	100	6	2	3	20	1	4
	colsample_bytree: [0.2-1]	0.99	0.5	0.56	0.42	0.87	0.22	0.46
	objective: multi:softmax eval_metric: mlogloss							
Random Forest	n_estimators: [50-500]	52	328	61	152	411	337	87
	max_features: [0.2-1]	0.38	0.74	0.88	0.46	0.53	0.94	0.22
	max_depth: [1-10]	100	1	10	5	20	1	5
	class_weight: [balanced, balanced_subsample]		balanced	balanced	balanced	balanced	balanced	balanced

Since decision trees are robust to work with categorical and continuous values, we kept the original values of our variables. Furthermore, similar to the previous subsection, we randomly split the data into 80% for training and 20% for testing. Since XGBoost and Random Forest do not natively support multitarget classification, we used *MultiOutputClassifier*, from the Scikit-Learn library, to fit one classifier per target. To select the best model, we used Bayesian optimization via the HyperOpt library. In total 100 iterations were performed for each combination of variable type and each technique. To guide the search for the best model, we set as loss function the metric *Micro F1-score*, hereafter called *Micro F1* for short. Generally, this metric is used to assess the quality of multilabel binary problems, where the score closer to 1 means better *Micro-Precision* and *Micro-Recall* (we will abbreviate both to *Precision* and *Recall*, respectively) and closer to 0 means poor model performance. Other metrics such as *Accuracy* and *Balanced Accuracy* are also presented to analyze our resulting models.

Table 8.14 shows the results of the best models obtained for each data input combination and Table 8.13 describes their hyperparameters and the search space used. As we can see in Table 8.14, calendar data together with vigilance alert levels improve the performance of the models. What is more, the best Random Forest model used only calendar data reaching an Micro F1 of 0.81. Also, we noted that when the inputs are weather, vigilance indicators, and weather plus vigilance indicators, the models show an F1-score below 0.64, a poor performance, when in fact these variables should present a greater contribution on the recognition of intervention peaks.

Therefore, these are the baseline results that we need to outperform to be efficient. An NLP tool must have an Micro F1 greater than those values. Thus, we have to see whether information derived from the text of the bulletins allows better prediction of interventions than simple calendar data, vigilance levels, or quantitative information such as temperature or wind speed. For this purpose, we are interested in comparing different NLP models.

Table 8.14: Prediction results of the multilabel models based on XGBoost and Random Forest techniques

Technique	Input	Micro F1	Accuracy	Balanced Accuracy	Precision	Recall
XGBoost	Calendar	0.80	0.48	0.82	0.80	0.80
	Weather	0.63	0.21	0.64	0.65	0.60
	Vigilance	0.62	0.23	0.62	0.61	0.62
	Weather + Calendar	0.71	0.30	0.71	0.73	0.69
	Calendar + Vigilance	0.81	0.48	0.82	0.81	0.80
	Weather + Vigilance	0.64	0.24	0.65	0.65	0.64
	All	0.71	0.35	0.71	0.74	0.68
Random Forest	Calendar	<b>0.81</b>	<b>0.51</b>	<b>0.83</b>	<b>0.82</b>	<b>0.81</b>
	Weather	0.61	0.24	0.61	0.63	0.59
	Vigilance	0.64	0.18	0.58	0.54	0.77
	Weather + Calendar	0.73	0.35	0.72	0.74	0.71
	Calendar + Vigilance	0.81	0.50	0.83	0.81	0.81
	Weather + Vigilance	0.62	0.24	0.61	0.63	0.61
	All	0.72	0.33	0.72	0.74	0.71

### 8.3.4/ PREDICTION OF INTERVENTION PEAKS USING MULTILABEL CLASSIFICATION MODELS BASED ON NLP TECHNIQUES AND TEXT FROM METEOROLOGICAL BULLETINS

In this section, we seek to discover if we have better learning for the 4 outputs when we process the texts of the bulletins. For this task, the dataset used considers the bulletins of the three neighboring departments of Doubs and the bulletins of Doubs. They were structured as shown in Table 8.2 and the task is maintained as a multilabel classification for the recognition of the intervention peak periods.

The initial dataset was split into 80% for the learning phase and 20% for the testing phase. Experimentation was then performed by varying different hyper-parameters for the different models. Results presented in Table 8.16 correspond to the best results obtained for each technique applied. We calculated the same metrics presented in the previous section 8.3.3 to identify possible improvements. On the one hand, for LSTM and CNN, we applied a text preprocessing and used the library Hyperopt to search for the best configuration for our models, where the number of iterations was 100, and the guiding metric was the Micro F1. The best setting for LSTM was the architecture n<sup>o</sup>1, with a learning rate = 6e-4, number of epochs = 200, and the batch size = 59. The best configuration for CNN was the architecture n<sup>o</sup>3 with a learning rate = 9e-3, number of epochs = 105, and the batch size = 95. On the other hand, transformers were used in its base models: CamemBERT (110 Million parameters) and FlauBERT (138 Million parameters) with the following hyperparameters. For CamemBERT: learning rate = 1e-5, number of epochs = 75, and batch size = 48. For FlauBERT: learning rate = 1e-5, number of epochs = 150, and batch size = 128. The search of the best hyperparameters was performed by using a random search with an early stopping strategy of 15 iterations. For more details on the description of the search space and the best configuration, see Table 8.15.

From Table 8.16, we see that LSTM and CNN results are quite similar. However, the two

Table 8.15: Hyperparameters search space and the best configuration for NLP multilabel models

Method	Search Space	Best configuration
CNN	type of architecture: [1,2,3]	3
	learning rate: [0.00001-0.01]	0.009
	batch size: [40-150]	95
	epochs: 500	105
	early stopping: 15	15
	restore best weights: True	True
LSTM	type of architecture: [1,2,3]	1
	learning rate: [0.00001-0.009]	0.0006
	batch size: [40-150]	59
	epochs: 200	200
	early stopping: 20	20
	restore best weights: True	True
FlauBERT	type of architecture: flaubert-base-cased	flaubert-base-cased
	learning rate: [0.0001, 0.00001]	0.00001
	batch size: [16-256]	128
	epochs: [10-200]	150
	early stopping: 15	15
	restore best weights: True	True
CamemBERT	type of architecture: camembert-base	camembert-base
	learning rate: [0.0001, 0.00001]	0.00001
	batch size: [16-256]	48
	epochs: [10-200]	75
	early stopping: 15	15
	restore best weights: True	True

Table 8.16: Prediction results of the multilabel models based on NLP techniques

Technique	Input	Micro F1	Accuracy	Balanced Accuracy	Precision	Recall
CNN	Bulletin Text	0.84	0.56	0.78	0.83	0.85
LSTM		0.85	0.56	0.80	0.84	0.86
FlauBERT		0.87	0.59	0.82	0.86	0.89
CamemBERT		<b>0.89</b>	<b>0.65</b>	<b>0.84</b>	<b>0.87</b>	<b>0.90</b>

Table 8.17: Accuracy results for each type of intervention, considering the models generated with the weather bulletins.

Model	Emergency Person Rescue	Total Person Rescue	Heating	Storm/Flood
CNN	0.76	0.82	0.84	0.82
LSTM	0.76	0.83	0.85	0.83
FlauBERT	0.79	0.87	0.88	0.85
CamemBERT	<b>0.80</b>	<b>0.86</b>	<b>0.92</b>	<b>0.86</b>

French transformers models, CamemBERT and FlauBERT, outperform both traditional techniques. This is mainly because the LSTM and CNN models were trained only with the vocabulary of the bulletins while the transformers were pre-trained with an extensive vocabulary of the French language (Common Crawl Oscar corpus). Thus, this set of experiments confirm the recent literature results [183] on text classification problems and the superiority of Transformers models. Note that, Accuracy and Balanced Accuracy are quite different due to the imbalance of the dataset as mentioned in Section 8.2.2. Nevertheless, when looking at the Micro F1, all the models in this section remarkably outperform the results obtained in the previous subsection with decision trees, in Table 8.14. What is more, the best model obtained with CamemBERT overcomes by far the best model obtained with Random Forest by 8%, 14%, 1%, 5%, and 9%, when comparing Micro F1,

Accuracy, Balanced Accuracy, Precision, and Recall, respectively.

When we examine accuracy by independent category in Table 8.17, we proved that the feature extraction from texts improves the recognition of intervention peaks due to rare events. The best NLP model reached accuracies of 80%, 86%, 92%, and 86% for Emergency Rescue People, Total Rescue People, Heating, and Storm/Flood, respectively. Moreover, the last 2 categories Heating and Storm/Flood, which represent interventions generated by rare events and which were complicated to recognize by the approaches analyzed in the previous sections, are the ones that demonstrated high accuracy with the NLP techniques and bulletin texts, without degrading the accuracy of the 2 rescue-type categories.

## 8.4/ DISCUSSION AND RELATED WORK

Among the works reviewed and related to the optimization of fire departments responses to incidents, we mainly found contributions for the prediction of interventions [146, 150, 152, 135] and fires [168]. As mentioned in section 8.1, in certain parts of the world firefighters are part of EMS, since they also provide ambulance services. In this way, predictions of traffic accidents [90], ambulance response time, and resource allocation are also included [80, 94, 149]. Furthermore, we can find works related to the predictions of rare events such as earthquakes [137] and hurricanes [104], which would allow firefighters to identify a specific location for damage assessment and develop a resource deployment plan.

In fact, the aforementioned works did not use NLP techniques. However, in other studies, NLP techniques demonstrate their outstanding utility by enriching data sources and predictions. This is achieved by recovering habits, preferences, emotions, feelings, and distress messages through the recognition of semantic patterns [82, 128, 96] from various media such as social networks, the news, therapeutic reminders, information systems under the video-based, audio-based and, text-based formats [169]. On the one hand, in [136], the authors presented a cognitive assistant system for EMSs based on Google Speech API, in which the voice records (incident description and patient status) received by the respondent are converted into texts for extract medical concepts. This way, the system responds by providing information to rescuers on protocols to follow such as resuscitation and airway management. On the other hand, NLP has contributed to the generation of terminological sources for the classification and forecasting of rare events or crises [78, 162, 129, 47]. For example, in [162], it is presented the first study for crisis management using French transformer-based architectures (BERT, FlauBERT, and CamemBERT) apply to French social media, to classify tweets for natural disasters. In [83], the authors make use of the Bayesian model averaging approach and linear-

chain conditional random fields to extract knowledge from tweets and build a decision support system to identify early warning signs of earthquakes. Also, in [117] is developed the Flood AI Knowledge Engine, which is a system composed of ontology management, query mapping and execution, and NLP modules. The system provides emergency preparedness and response, as well as knowledge about flood-related resources for the population. This knowledge is returned by interpreting natural language queries from users.

Having seen the impact of NLP on the recognition of extreme events, at the time of this study, and to the authors' knowledge, no previous studies have exploited a source such as weather bulletins to detect trends in the number of firefighter interventions. For this reason, the present study took advantage of NLP techniques to process these bulletins and predict peak intervention periods for the categories: Emergency Person Rescue, Total Person Rescue, interventions related to Heating, and Storm/Flood.

First, in our experiments developed in subsections 8.3.1, 8.3.2, 8.3.3, we observed that tabular data were very useful for recognizing seasonalities and trends in the flow of incidents, since the major operational burden are the interventions of type rescue people, i.e., human activity. For example, people are more active during the day than at night, there are more drownings in the pool during the summer than in the winter, as the population increases the incidents also increase, etc. However, some interventions are difficult to detect, since they occur only a few times a year. These are produced by rare events such as natural phenomena (storms, floods, forest fires, etc.), and although their occurrence is minimal over the years, the workload they produce in a small period can be 28 times more than normal in some cases. For example, in 2016, the average number of interventions assisted by SDIS25 per hour was 3.34. However, there was an hour in which 84 interventions occurred due to a storm. The storm caused flooding in the region, human and material damage, and breakdowns in the service of SDIS25 [152]. Therefore, in subsection 8.3.4, we developed models based on NLP techniques to predict intervention peaks using meteorological bulletins, being the models with the best performance those built with the Transformers CamemBERT and FlauBERT. The results obtained are significant for practical purposes. In production, the best model could be deployed as a small stand-alone application or the predictions (the binary indicators) could be included in a larger set of tabular data that would be the input for a predictive model of the number of interventions for a certain time and location. In this way, with an initial application or with a more robust system, the fire department could reorganize its personnel and armament to cope with these periods of high demand, reduce breakdowns in service due to lack of resources, and save more lives.

Finally, the present work was published in the Neural Computing and Applications journal [191].

## 8.5/ CONCLUSION

The present study demonstrates the effectiveness of NLP techniques for the recognition of rare events that will cause an increase above the average in certain periods of firefighter interventions. This is done by processing the texts contained in the weather bulletins using the traditional techniques LSTM and CNN, and transformers CamemBERT and FlauBERT. The results of the NLP models and bulletin texts exceed those of the baselines with Decision Trees (XGBoost and Random Forest) and tabular data by 8% and 14% when comparing the best Micro F1 and Accuracy, respectively. The advantage of using these texts is also reflected when assessing the accuracy of the 2 categories with interventions related to rare events, achieving 92% for Heating and 86% for Storm/Flood with the best CamemBERT model developed. Our results were published in [191].

In this way, fire departments and EMS, in general, would be able to identify peak periods of interventions and optimize their response by establishing better strategies to prepare their armament for natural disasters (storms, floods, etc.) and keep the population better protected and safe.

# IV

## CONTRIBUTION: OPTIMIZATION OF FIRE DEPARTMENT RESOURCES



# OPTIMIZING THE BREAKAGE CALCULATION

In Chapter 3, we have reviewed the internal data of the SDIS25 and the design of the breakage calculation. This indicator of quality will be constructed in the present Chapter. In addition, we will describe the variables involved, analyze the results obtained, and explore other ways to exploit the results such as predicting the number of daily breaks.

## 9.1/ INTRODUCTION

As mentioned in 1.1, several fire departments around the world are vulnerable due to resource shortages. In addition, if there is no tool to monitor the performance of the service, it will be very difficult to establish resource allocation strategies to deal with the seasons or areas of highest demand. From the analyzed data, we have seen that for seasons in the year the number of emergencies decreases and increases. What is more, the increment is exponential when unexpected natural disasters occur. And if there are not enough resources to attend all these emergencies, the results are economic, social and environmental losses.

The first step for a better distribution of resources in the face of a shortage is to identify the current state of the service, i.e., to determine the types of operations that generate more demand, the areas that generate more interventions, the types of engines more demanded and consequently the most requested training for the agents, the centers with more and less resources, and the proximity of the centers to the areas. Second, measure the service quality with an indicator to respond the questions: 1) at what moment does a service breakdown occur?, 2) how long is the breakdown time?, and 3) what was the cause (lack of engine, agents or both)? In our case, these last 3 questions are the ones we will answer when implementing *the breakage calculation* from 3.2, which is our service quality indicator.

Finally, the data generated for several years with the indicator is complemented by external data and the new dataset will be used to predict breakages per day.

## 9.2/ OPTIMIZED SEARCH FOR AVAILABLE ADAPTED ARMAMENT

As mentioned in 1.1, there are 5 types of interventions in SDIS25, from these types, SAP and INC represent the largest amount of operational load. For this reason, we will take only SAP and INC interventions to be processed by the breakage calculation. Also, we will use the SDIS25 internal data detailed in 3.1.

To calculate breakdowns, the set of interventions should correspond to a large period of time (e.g., 30 000 interventions per year or at least 4 months with the highest workload), since it better represents the demand behavior. Each intervention will pass through the summarized flow showed in Fig. 9.1.

First, we collect all data needed from the intervention: location (commune-quartier and zone), list of defense centers, we identify the first aid and adapted engine from all engines used in the intervention, agents, and datetimes.

Second, as we can see in Figures 3.3, 3.4, 3.5, and 3.6, BSC calculation is a simple subtraction of 2 values. However, BPS calculation requires a search to find an another adapted engine in 3 occasions, i.e., at the moment of the departure, between the departure and the arrival at the scene, and between the arrival at the scene and the end of the intervention. For this reason, each time that an adapted engine is requested in a specific datetime, we will filter all the possible available adapted engines and agents

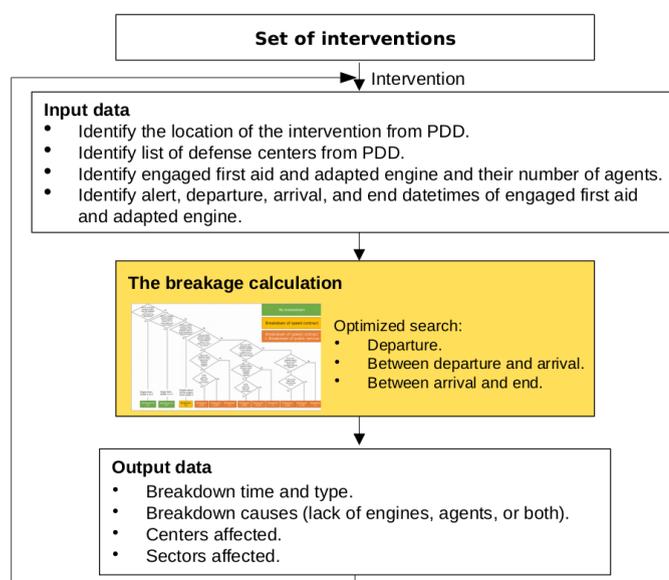


Figure 9.1: Breakage Calculation Input and Output Data.

by center. For example, there is an INC intervention, our filter will consider the following administrative specifications: there are 9 possible adapted engines for INC (in french are CCR, CCRSR, FPT, FPTGP, FPTHR, FPTL, FPTSR, CCFM, and FTLHR), the number of agents have to be 6 (1 driver –COND, 2 crewman –EQUI, 1 engine chief –CA, and 2 team leaders –CE), and the agents skill codes have to be FPCOND that corresponds to the function COND, FPEQUI corresponds to the function EQUI, FPCA corresponds to the function CA, and FPCE corresponds to the function CE (this could vary according to the chosen engine). Then, we could make a search by trying all combinations in each adapted center with engines, agents, skills, and availabilities to build and adapted engine, but this would be using brute force and it would take hours, days or weeks to process all interventions. Therefore, we created a model to optimize the search. We used Advanced Process OPTimizer (APOPT), which is a MINLP solver in the Gekko library [110]. The model is described in 9.1.

$$\begin{aligned}
 & \min \quad 0, \\
 & \text{s.t.} \quad A_{ij}x_j \geq b_i, \\
 & \quad \quad A_{ij}, x_j \in \{0, 1\}, \\
 & \quad \quad b_i \geq 0
 \end{aligned} \tag{9.1}$$

Where:

$A_{ij}$  is a Boolean matrix representing the ability  $i$  which has an available agent  $j$ .

$x_j$  is a Boolean vector that represents if an agent  $j$  was selected (it takes value 1) or not (it takes value 0) to be part of the adapted engine.

$b_i$  represents the minimum number of agents required per skill for the adapted engine.

Notice that the aim is not to minimize an objective function, but to quickly find a feasible solution. We established the constraint solver to compute an arbitrary solution within the space of feasible solutions. Hence, the objective function is just a constant (zero) independently of the solution.

Given an adapted engine and knowing that an agent can have several skills at the same time, some of the principal restrictions were: the number of available agents must be greater or equal to the minimum number of agents required to depart. At least one of the available agents must have the ability to be a driver for a specific adapted engine. The skills found in the available agents must cover the required skills. For a clearer view of the model, an example is shown as follows:

Let us imagine that there is an intervention of SAP type, there are 10 firefighters available at the moment and the armament needs to be assembled for an adapted engine VSAV. This engine needs 3 agents: at least 1 agent with the function COND (the skill required is VSCOND), at least 1 agent with the function CA (the skill required is VSCA), and at least 1 agent with the function EQUI (the skill required is VSEQUI). Then, the system would

be:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{10} \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (9.2)$$

Each row of the Boolean matrix corresponds to a function (COND, CA and EQUI) and each column represents an available agent. Together they symbolize the function per agent. The vector column  $x$  represents the firefighters or agents that will be chosen and the vector on the right side shows the number of agents needed per function. The solution of this system allows us to rebuild an armament in a specific time and ensure the availability of an adapted engine in reserve and in case of a new emergency.

Third, and finally, the output of the breakage calculation helps us to respond to the three questions made in the introduction, i.e., identify the service state, by recognizing the times and types of breakdowns, their causes, and affected centers and territories.

### 9.3/ PREDICTING THE NUMBER OF BREAKDOWNS PER DAY

If for several years, we apply the breakage calculation, the data generated will be statistically rich. This data could allow us to identify trends over time. For example, if we predict the number of breakdowns for the coming weeks or months, we will have a broader view of what SDIS25 will face if there is no acquisition or reorganization of the current resources. Therefore, in this subsection we will exploit these data and more from external sources with ML techniques. The goal is to forecast the daily number of breakdowns for the next months.

The testing set will be 12 months of the year 2019 and the training set will start with the years 2017 and 2018, and according to the month predicted the training set will increase. In addition, we considered only BPS and are grouped by day. Although the initial dataset is small, the predictions will be a first approach of another way of resource optimization, different from the approach of workload simulation, which is beyond the scope of the current chapter. The predictions will be the total number of BPS that the SDIS25 will have for each month of 2019.

#### 9.3.1/ DATA PREPROCESSING

In order to populate our data set, extract more characteristics from the explanatory variables and normalize them to assist in convergence, we performed data preprocessing.

The data collected cover internal variables such as: the number of engines available per

day, the number of agents who worked on a given day, and breakdowns that occurred in Doubs region. In previous articles [122, 146, 151, 150], several models were constructed for the prediction of interventions. Therefore if we know the future interventions with a reasonable accuracy, these can be used as input data for the prediction of breakdowns. In our tests, the number of interventions per day is included in the data set as an explanatory variable.

Furthermore, external variables from sources 3.2 such as Météo-France (tabular data), Sentinelles, Bison-Futé were added. Besides, the distance between the planet Earth and the Moon to analyze possible correlations (e.g., between solar flares and fires due to electrical incidents), obtained by day from Skyfield [138]; time variables such as day, day in the year, day in the week, month and year; and time indicators such as weekend, beginning and end of month, beginning and end of year.

Numerical data such as meteorological variables, epidemiological data, distance to the moon, day, day in the week, day in the year, month and year were standardized with the RobustScaler method from the Scikit-Learn library in Python [52], where the data are centered before scaling. Categorical data such as traffic indicators were encoded using the One-Hot-Encode method, also from Scikit-Learn. Finally, the target variable was not encoded, since better results were obtained without it.

The numerical variables collected by hour were averaged over a day, and from the categorical ones their mode was calculated. The breakdowns for the years 2017 and 2018, 730 samples in total, were used during the training to predict the month of January 2019; the breakdowns for 2017-2018 and January 2019 were used to predict February 2019, and so on, i.e., a model is generated for each month of the year 2019.

### 9.3.2/ MODELLING

The baseline model was constructed by calculating the daily average BPS in each month and for the years 2017 and 2018. Then, we tested 5 ML techniques: MLP, BRR, SVM, and RF, all from Scikit-Learn, and XGBoost. Since a wide variety of hyperparameter configurations are presented in each technique, we opted for Bayesian Optimization to guide the learning of our models from the library [67]. For each technique, the Bayesian Optimization was configured with 300 initial points, 100 iterations, and the exploration strategy called Upper Confidence Bound.

## 9.4/ RESULTS AND DISCUSSION

### 9.4.1/ RESULTS OF THE BREAKAGE CALCULATION

In this subsection, we present the results obtained after applying the breakage calculation for years 2017, 2018, and 2019. Approximately 83000 interventions of SAP and INC were processed. We will show the breakdowns by type (BSC and BPS) and not by cases to generalize the results.

From the results, we generated statistics for the analysis of the service state. In Figures 9.2 and 9.3, we show the number of BSC and BPS, respectively, from January to December over the 3 years. When we compare both figures, we determine that the number of BSC is more than twice the number of BPS, which means that in the majority of cases we have had an adapted backup engine in the centers, ready for a new simultaneous intervention. However, this does not mean that the service is effective enough, on the contrary, given the high number of BSCs, it implies that many engines exceeded the time frame allowed to reach the scene of the intervention, i.e., the time of vulnerability of the victims was higher. Also, in both figures, we can see that the number of breakdowns has a tendency to increase in the months of January, July, and December, reaching a peak in July of 2019. In Table 9.1, we describe statistical information of BPS over the 3 years and by month, where  $Nb$  is the number of total breakdowns in the month,  $DAvg$  is the average number of breakdowns per day in the month,  $DStd$  is the standard deviation per day in the month,  $MAvg$  is the average of the total number of breakdowns by month. From here, we can verify that July is the month with the most breakdowns with a  $MAvg$  of 233.67 and a daily average ( $DAvg$ ) going from 5 (2017) to 9.84 (2019), almost double. What is more, in general, year after year the breakdowns have been increasing.

In addition, Figure 9.4 exhibits the number of BPS produced by lack of: engine, agents, and both. It also includes, the breakdowns caused by the lack of the training types in the agents: COND, CA, CE, and EQUI (all needed for the adapted engines used in SAP and INC interventions). On the one hand, we discovered that in the month of July 2019 the breakdowns were caused by the lack of agents. It is possible that this was due to the holidays that are generally taken at this time of the year in France, i.e., when the number of emergencies tends to increase and fewer agents are available in the centers. On the other hand, August and December present breakdowns produced by the lack of engines, i.e., there was a high operational demand that was not supported by the amount of resources allocated in centers, even if there were agents there were not enough engines to depart). The breakdowns in August could be due to the holidays, where there is a higher human activity (traffic accidents, drowning in swimming pools, etc), and in December due to Christmas and winter (traffic accidents, domestic fires, slips on ice, etc).

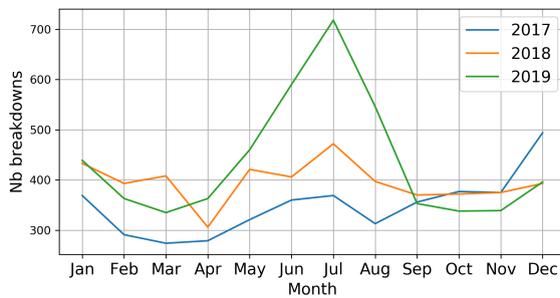


Figure 9.2: Number of BSC by month.

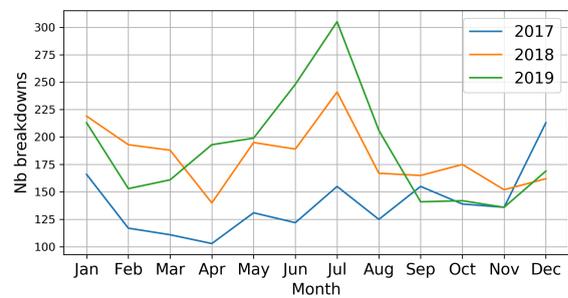


Figure 9.3: Number of BPS by month.

Table 9.1: Statistical information of breakdowns of public service by month from 2017 to 2019.

Month	2017			2018			2019			MAvg
	Nb	DAvg	DStd	Nb	DAvg	DStd	Nb	DAvg	DStd	
January	166	5.35	2.98	219	7.06	4.05	213	6.87	3.11	199.33
February	117	4.18	1.58	193	6.89	2.73	153	5.46	2.53	154.33
March	111	3.58	1.95	188	6.06	3.65	161	5.19	2.61	153.33
April	103	3.43	1.96	140	4.67	2.43	193	6.43	2.75	145.33
May	131	4.23	2.50	195	6.29	2.59	199	6.42	3.80	175.00
June	122	4.07	2.32	189	6.30	3.25	248	8.27	4.08	186.33
July	155	5.00	2.96	241	7.77	3.85	305	9.84	3.30	233.67
August	125	4.03	2.28	167	5.39	2.98	206	6.65	2.56	166.00
September	155	5.17	2.38	165	5.50	2.98	141	4.70	2.39	153.67
October	139	4.48	2.02	175	5.65	1.93	142	4.58	2.39	152.00
November	136	4.53	2.63	152	5.07	2.58	136	4.53	2.60	141.33
December	213	6.87	3.22	162	5.23	3.00	169	5.45	2.83	181.33
Total Avg	128.5	4.58	2.40	182.17	6.00	3.00	188.83	6.20	2.91	

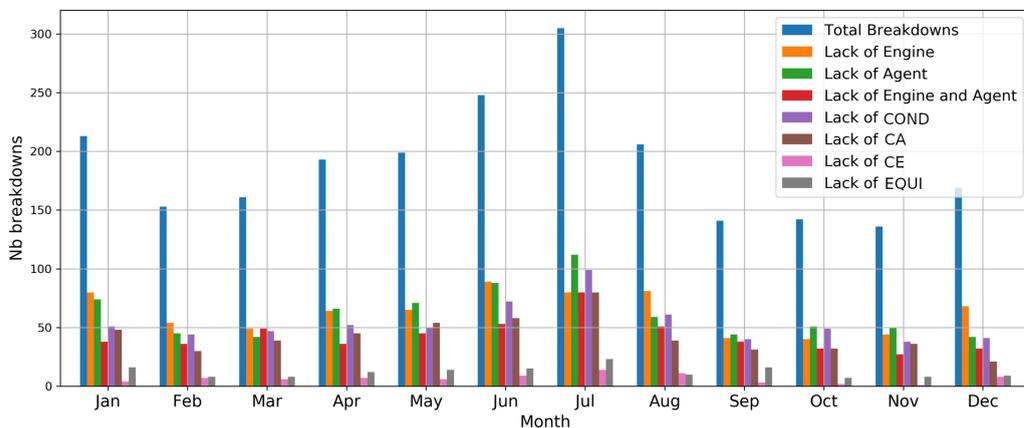


Figure 9.4: Causes of breakdowns of public service by month in 2019.

The visual representation given in Figures 9.5 and 9.6 depicts the percentage of BPS of SAP and INC types, by sectors commanded by adapted centers and deployed on the Doubs department. When comparing both maps, apparently, we could say that there are more INC breakdowns. However, we have to remember that the largest operational demand is of SAP type, thus, both would be at different scales. We can identify that centers Mouthe, Arc-Et-Senans, and Le Russey, in both maps, are the most affected. Therefore, they would require a better distribution of adapted resources in certain periods of the day or year.

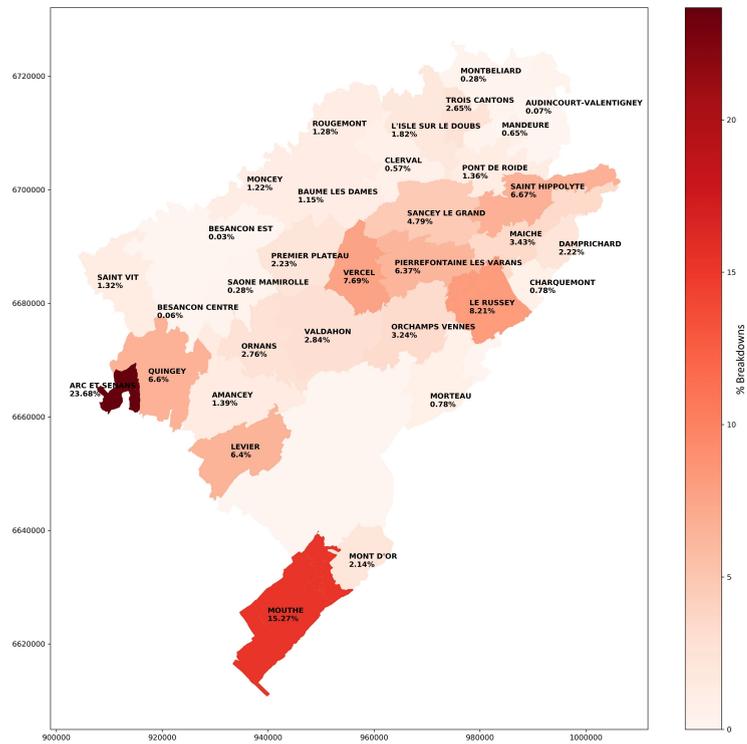


Figure 9.5: Breakdowns of public service of SAP type and by sector (Map of Doubs 2019).

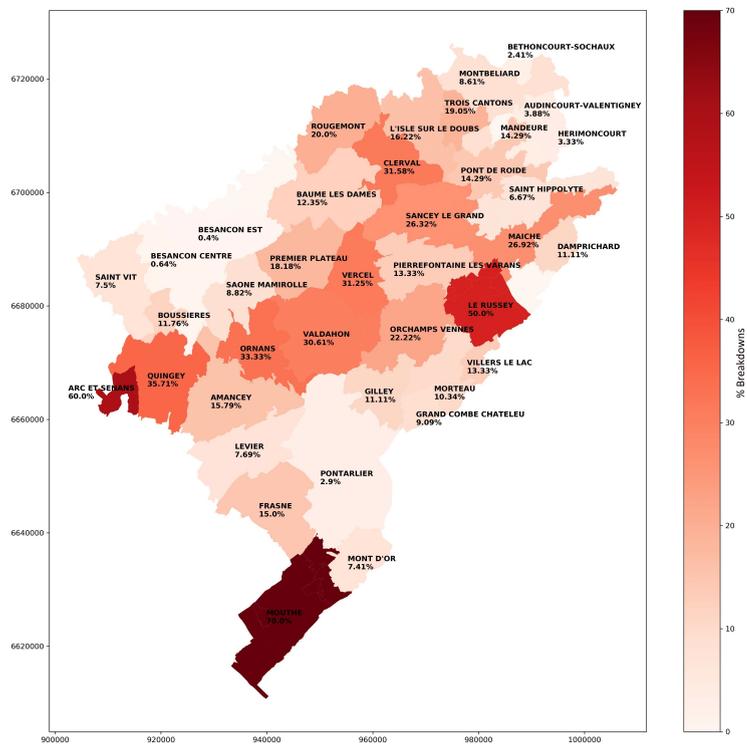


Figure 9.6: Breakdowns of public service of INC type and by sector (Map of Doubs 2019).

## 9.4.2/ RESULTS OF THE BREAKDOWN PREDICTIONS

In this section we describe and discuss the obtained results for the predictions of BPS. Table 9.2 shows the Root Mean Square Error (RMSE) and the Mean Absolut Error (MAE) metrics for each month of 2019 and by each technique. In the last row the averages are calculated in general. Table 9.3 shows the configurations of the best models with the XGBoost technique for each month of 2019. Table 9.4 presents the RMSE and MAE metrics calculated based on standardized breakdowns with the Scikit-Learn MinMaxScaler method. Thus, it is possible to compare which month is presenting a poor performance and analyze it. Figure 9.7 compares the daily predictions of public service breakdowns and the real number of them for each month in 2019.

Table 9.2: Predictions of public service breakdowns by month for 2019.

Month	Baseline		XGBoost		MLP		BRR		SVM		RF	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
January	3.46	2.71	2.29	1.90	2.52	2.03	2.85	2.35	3.51	2.68	2.76	2.13
February	3.54	2.93	2.05	1.64	2.25	1.79	3.30	2.79	2.41	1.82	2.27	1.86
March	3.61	2.94	2.08	1.68	2.24	1.74	2.30	1.84	2.54	2.06	2.29	1.81
April	3.91	3.23	2.60	1.90	2.61	2.03	2.97	2.33	3.08	2.40	2.89	2.23
May	4.17	2.94	3.65	2.84	3.63	2.68	3.87	2.81	3.96	2.84	3.92	2.87
June	5.19	4.13	3.03	2.27	3.32	2.40	4.58	3.80	5.01	3.87	3.73	2.90
July	4.94	3.97	3.48	2.84	4.01	3.23	5.06	4.06	5.30	4.26	4.58	3.68
August	3.88	3.26	2.46	2.10	2.48	2.03	2.75	2.26	2.64	2.26	2.44	2.06
September	3.24	2.53	2.60	2.00	2.19	1.67	2.74	2.20	2.34	1.87	2.64	2.17
October	2.92	2.32	2.03	1.55	2.16	1.58	2.46	1.87	2.33	1.81	2.24	1.71
November	2.37	1.80	2.26	1.83	2.36	1.97	2.67	2.27	2.53	2.07	2.29	1.83
December	4.00	3.25	2.19	1.74	2.38	1.81	2.88	2.23	2.69	2.00	2.25	1.65
Average	3.77	3.00	2.56	2.02	2.68	2.08	3.20	2.57	3.19	2.50	2.86	2.24

Table 9.3: Hyperparameters of the best XGBoost model by month for 2019.

Hyper-Parameter	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
max_depth	9	2	4	5	9	3	3	3	9	9	2	9
n_estimators	285	70	194	154	57	75	483	50	426	215	267	91
learning_rate	0.27	0.39	0.16	0.55	0.74	0.81	0.90	0.05	0.17	0.33	0.72	0.51
subsample	0.51	0.66	0.51	0.53	0.80	0.93	0.99	0.73	0.53	0.84	0.98	0.83
colsample_bytree	0.64	0.85	0.53	0.61	0.53	0.70	0.73	0.5	0.94	0.93	0.95	0.75

Table 9.4: Metrics of the best XGBoost model with scaled breakdowns predictions by month for 2019.

Metric	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
RMSE	0.21	0.19	0.19	0.26	0.24	0.15	0.25	0.31	0.24	0.17	0.23	0.17
MAE	0.17	0.15	0.15	0.19	0.19	0.11	0.20	0.26	0.18	0.13	0.18	0.13

From Table 9.2, we can see that our baseline is surpassed by all the 5 techniques. Also, we can detect that the months of May, June, and July presents the lowest performance with the 5 techniques. Those are the months with the higher number of breakdowns. When looking at the average, XGBoost shows the best results. The objective function defined in XGBoost was the regression with squared loss and the Bayesian Optimization search space considered the bounds: max\_depth between 1 and 10, learning\_rate between 0.05 and 1, estimators between 50 and 500, subsample between 0.5 and 1,

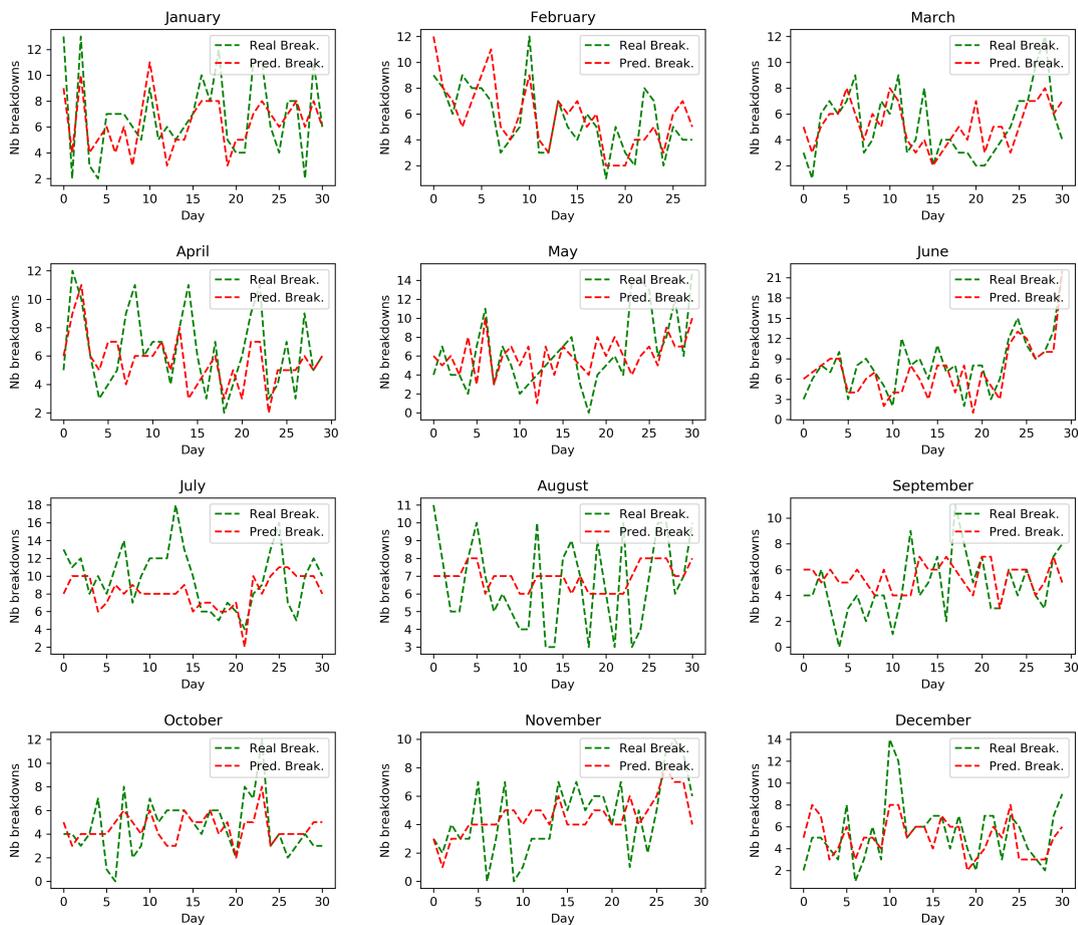


Figure 9.7: Predictions of public service breakdowns using the best XGBoost model by month for 2019.

and `colsample_bytree` between 0.5 and 1. In second place, it is MLP with an RMSE of 2.6804 and a MAE of 2.0791. Its grid for Bayesian Optimization followed the bounds: `hidden_layer_sizes` between 50 and 900 neurons, `learning_rate_init` between 0.0001 and 0.1, `max_iter` between 0.05 and 0.5, `tol` between 0.00001 and 0.01, `momentum` between 0.00001 and 0.01, `alpha` between 0.00001 and 0.01, and by default the activation function ReLu, solver Adam and `learning_rate` constant.

When we visualize Fig. 9.7, we can see that the predictions for June follow the trend, but in Table 9.2, the metrics RMSE and MAE are one of the highest. This is because June is one of the months with the highest number of BPS and the highest standard deviation. Since all the months are in different intervals of breakdowns and in order to compare them, we standardized the number of breakdowns for each month between 0 and 1, using `MinMaxScaler`. The true number of breakdowns were fitted and the predicted numbers were transformed according to the scale defined by the true ones. These results can be seen in Table 9.4, which confirms that actually the model for June is one of the best compared to the other months as it is visually validated in Fig. 9.7. What is more, the

model for the month of August is clearly underfitted. This is a future work, where we will have to identify the months that produced predictions with low precision and study them to know their causes, i.e., to recognize which more internal and external variables should be added in the modeling process and what events tend to occur on those dates in the region and in the fire department.

#### 9.4.3/ DISCUSSION AND RELATED WORK

The results of the breakage calculation gave the fire department an overview of the status of their service. The SDIS25 was able to identify the centers with the most breakdowns in public service and these, in turn, received more resources as trained personnel and engines. Also, it allowed to recognize the territories with the highest number of breakdowns to the speed contract and to establish strategies to reduce these arrival times.

When reviewing the literature, to the author's knowledge, there are no research works with the specific objective of recognizing breakdowns. However, an approximate work to our approach could be [214], where tools for the operational planning of a fire department in Sweden were created. At [214], they initially created and defined the term "preparedness" based on indicators such as the number of available resources, response time, and demand. What is more, a way to quantitatively measure this term was established. Although the approach is different from the one published by us, it is also used as an indicator of operational capacity.

The results of breakdown predictions showed that through ML techniques we can recognize tendencies in the breakdowns throughout the year, considering variables such as the number of interventions that may occur, the amount of resources in personnel and engines, external variables such as temperature, humidity, and traffic indicators, among others. Although not all the models of the months showed remarkable results, the XG-Boost technique was the one that showed the best performance. Over the years, the indicator will generate more breakdown data that will improve the performance of the ML models. Also, these data could be used in other future optimization approaches (e.g., simulation of the operational load).

Some works related to breakdown predictions are [150, 146], where the prediction of the number of interventions is performed, i.e., the models predict the operational load that the fire department will have in the next few hours and for regions, respectively. As a future work, these models could be part of an integrated system for the prediction of breakdowns. The preliminary results of this chapter were published in the article [152].

## 9.5/ CONCLUSION

This chapter implements the breakage calculation modality as an indicator of the operational service quality. For this, it is proposed an optimized search method for the adapted engine (essential for breakdown identification) based on MINLP. The results obtained allow to determine the type and time of the breakdowns produced during the interventions, the affected centers and the causes that produced them, such as the lack of personnel or engines, and the affected sectors.

In addition, we have proposed an initial approach to predict the number of daily breakdowns per month, based on ML techniques (XGBoost, MLP, BRR, SVM, and RF) and using as input data the interventions that have occurred, resources considered, external variables (meteorological, traffic, etc), and the breakdowns generated during 3 years. Our results were published in the article [152]. This would allow fire departments, in the long run, to test different combinations of resource distribution to forecast the possible number of breakdowns that could be generated. Thus, fire departments could make better decisions for allocating human and technical material or to quickly regain operational stability in the face of an over demand of resources.

## DEVELOPMENT OF THE OPERATIONAL LOAD SIMULATOR

In Chapter 9, we have developed an optimized search to implement a tool called *the breakage calculation*. In the present Chapter, we will use the breakage calculation together with an operational load simulator that we will build. The main objective is to simulate the resource demand that occurred in a period of time. Then, we will validate the accuracy of the simulation with the breakage calculation. Thus, in the next chapters, the simulator will work as a core and it will be integrated into several algorithms developed for resource optimization in fire departments.

### 10.1/ INTRODUCTION

As we have discussed in previous chapters, fire departments are subject to the vulnerability of their response system if they do not establish contingency measures. Part of these contingency measures are resource distribution strategies, which would help them minimize negative consequences. We know that there are various ways to define policies for the distribution of resources and in this case, our goal is to do so through the development of algorithms based on artificial intelligence and automated tools.

Since the beginning of this thesis and with the full support of SDIS25, we analyzed their data and understood their processes. We identified that we had historical data on the interventions that occurred in the last 16 years (2006-2022), but we did not have data on the number of breakdowns generated by a certain distribution of resources.

In chapters 4 and 8 we built intelligent models for the prediction of interventions, this allowed knowing the future demand with good precision and making better operational decisions at the moment. However, these models did not provide punctual and long-term visual recommendations on the distribution of resources.

Thus, seeing the need to optimize the distribution of resources, the SDIS25 defined the

breakage calculation modality. This was implemented as a service quality indicator in chapter 9 and only with data available from 2017. In chapter 9, we also showed a ML-based solution to predict breakdowns per day using the number of resources, number of interventions expected, and external variables. However, the amount of breakdown data to feed ML models and give outstanding predictions is still small.

Therefore, and despite the myths related to the simulation [27], the solution proposed in this chapter is to create an operational load simulator for the fire department. In this way, we could experiment with various combinations of resources (engines, agents, training, new centers), allowing us to see the impact of the combinations on all the centers deployed in the territory. The results will be visualized by means of maps and statistics, in a global way such as the total result of the breakdowns, and detailed such as the identification of the necessary armament for each center.

## 10.2/ SCENARIO VARIABLES

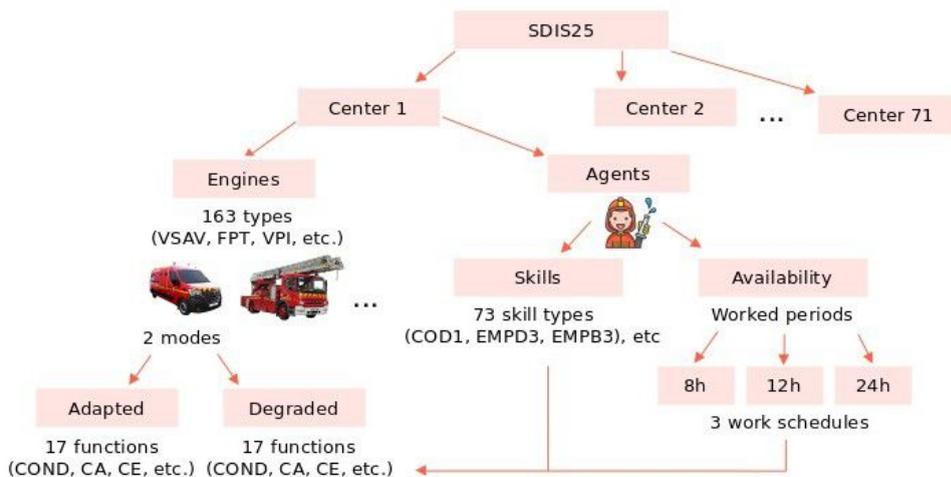


Figure 10.1: Variables with more impact on the simulation of the operational load.

The distribution of resources in one or several centers provokes a chain reaction in all the centers of the territory, generating an increase or decrease of breakdowns in the service. To build the simulator, we have to know the scenario variables and among them identify those that will have the greatest and least impact during the simulation.

In Fig. 10.1, we can see an scheme of the most important variables involved in the simulation. These are the 71 centers, the 163 types of engines, their 2 modes of use (adapted or degraded), the 17 possible functions, the 73 types of agent skills, and the 3 possible periods of availability for an agent. In real life, there are also 2 types of firefighters: professionals and volunteers, with volunteers being on average 3 times more than professionals. However, to reduce the search space, we do not consider this distinction. Once

the simulator has been built, in the following chapters and depending on the type of resource optimization, we will establish different search spaces with variables such as the number of engines by type and by center if we want to optimize the distribution of engines, different coordinates if we want to position a new center, and the number of firefighters by skill and availability if we want to size a new center.

The reasons for considering all these variables are detailed below:

1. The centers. There are 5 categories, from the smallest with only first aid engines located mostly in rural areas and with less frequency of interventions, to the largest with first aid engines, adapted engines, and others, located in the main cities (urban areas) and with more frequency of interventions. Therefore, the type of center and its location limits the amount of resources allocated and defines the deployment plan used in the breakage calculation.
2. The engines. There are various types of engines, among them, there are specific engines for a mode and type of intervention (e.g., adapted VSAV for SAP, adapted FPT for INC, first aid VTU for SAP and INC), if these were missing, several interventions could not be carried out simultaneously and breakdowns would occur.
3. The agents. There is a certain number of agents per center to operate the engines. If there were engines but not enough firefighters, the engines could not depart and breakdowns would be generated.
4. Agent training. Agents have various types of skills (training) and functions to fulfill in a type of engine and mode. If we had engines and firefighters, but the firefighters were not well trained, the engines would not be able to depart and breakdowns would occur.
5. Availability of agents. Mainly, there are 3 types of periods identified for firefighters: 8h (usually, it is 7h but we have rounded it to 8h so that it is divisible by 24h), 12h, and 24h. These periods can start and end at different times. For example: 3 periods of 8 hours in one day (0-8, 8-16, 16-24), 2 periods of 12 hours in one day (0-12 and 12-24), 1 period of 24 hours (0-24). In each period, we will find a number of firefighters available. Since there are more interventions during the day than at night, if we had fewer firefighters deployed during daytime hours, more breakdowns would occur.

### 10.3/ SIMULATOR OVERVIEW

When the breakage calculation is used as a quality indicator at SDIS25, it processes SAP and INC interventions. Although AVP, OD, and RTN interventions are not processed, the

resource consumption generated by these interventions is represented by time periods in which the resources are unavailable. All this operational load must be taken into account when simulating a period. We will use the breakage calculation to measure the simulation and compare it to the breakdowns that occurred in reality. In this way, we will determine if what is simulated is close to reality. To create a final simulation model, we will establish a baseline model M1 and two models named M2 and M3.

Before applying the simulation, we have to select a period, calculate its breakdowns, and generate statistics. The statistics obtained will be our real reference to compare with the statistics generated from the simulation. This real reference is our baseline model M1.

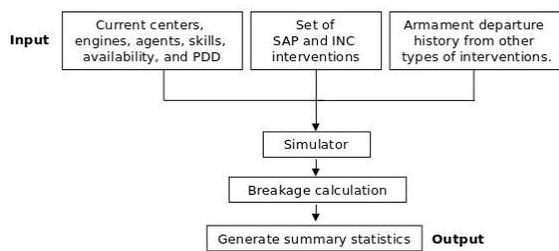


Figure 10.2: Model M2.

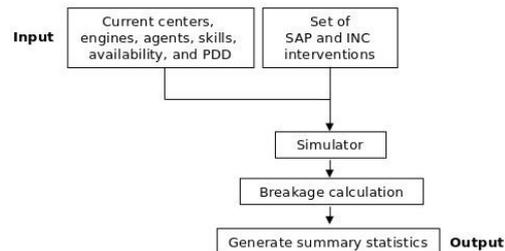


Figure 10.3: Model M3.

Then, in Fig. 10.2 and Fig. 10.3, we can see the schemes of the simulation models M2 and M3, respectively. In both models, the breakage calculation is applied and statistics are generated immediately after the simulation. The difference between the two lies in the fact that M2 is a model that considers the operational load generated by AVP, OD, and RTN interventions, similar to M1, while M3 does not consider this operational load.

Thus, M1 represents what happened in reality and M2 is a direct simulation of reality. Given that the greatest demand is generated by SAP and INC and we know their deployment rules, we will remove the load generated by AVP, OD, and RTN from M2. This will generate M3, which is an indirect simulation of reality. We deduce that M3 will present less breakdowns than M2, since it does not consider the entire operational load. However, M3 will be the image of M2 on a smaller scale, since it will maintain parameters identified from M1 and keep the flow of the largest operational load (SAP and INC interventions). Also, M3 will be more light when processing data for resource allocation optimizations made in the next chapters.

Several types of engines are used during an intervention. These come from different centers and are requested at different times, depending on the development of the intervention and their availabilities in the centers. This makes forecasting the demand for engines difficult. However, we know that for a SAP or INC intervention, the deployment rule indicates that at the beginning we will need an available adapted engine, but if there is none at the requested time, we will send a first aid engine and then an adapted engine. Table 10.1 describes the basic engines and functions required for SAP and INC.

In addition, there are secondary engines (e.g., vehicle with retractable ladder –in french EPA, large capacity tanker truck –in french CCGC, light medical rescue vehicle –in french VLMS, etc.), which are used during these interventions and will be considered during the simulation because they generate the unavailability of resources.

In Fig. 10.4, we can see a detailed scheme of the processes carried out inside the simulator. Initially, we define a set of SAP and INC interventions. Since we do not predict the exact moment an emergency will occur, we will use the actual start time of the intervention. In addition, we will identify the types of engines used in the real intervention and these will be processed one by one by the simulator. The end of the intervention is the end datetime of its last engine in operation. On the one hand, we have first aid and adapted engines. For them, we will determine their alert time (when we find an available armament with the optimized armament search algorithm in 9.2), departure time (with an XGBoost predictive model), arrival on the scene (with an XGBoost predictive model), and the end of their operation (with the actual duration of the engine in the scene until it returns to its center). The need to define these times is due to the fact that after the simulation, the breakage calculation will be launched and the times of the adapted and first aid engines will be validated. On the other hand, for the secondary engines, we will only calculate the alert time (with the optimized armament search algorithm in 9.2) and the end of their operation (based on probabilities). The departure and arrival times of these engines will not be necessary because the breakage calculation does not validate them. However, the alert and the end are important, since they will let us know the period of unavailability of the engine and its agents.

The departure and arrival times will be calculated using two predictive models described in more detail in the following sections. These models are built with data of engine departures from existing centers and their predictions are better than probability models. However, these intelligent models could not be used by a new center when we will simulate its creation. Therefore, we will use two probabilistic models, *genextreme* and *lognormal*, to calculate the departure and arrival at the scene, respectively, of the engine coming from the new center. These models, from the SciPy library, were selected based on the closest possible similarity to the time distributions of all first aid and adapted engines.

In the case of the secondary engines, we will select a random value from a uniform distribution between 0 and 60. This random value simulates the minutes after the start of the intervention, in which the engines will be requested. The values are between 0 and 60 because according to historical data 99% of the requests for secondary engines are made in the first hour of the intervention. Then, we will use the armament search algorithm in 9.2 according to the availability times of the engines and agents. The time in which we find an available armament and it is closest to the requested engine time will be identified as the alert time. From the set of real SAP and INC interventions, we

will obtain the duration times between the alert and the end of the operation grouped by engine type. Thus, we will construct a frequency distribution with the times and for each time range we will calculate its probability. This set of probabilities by type of secondary engine will be used in the simulator to generate the duration of the engine and to obtain the end datetime of its operation. For example, a VLSM has had durations of 15, 15, 30, 30, 30, and 50 minutes in the real period, if we use a number of bins=3 and a base range of 0-60, we will create the ranges 0-20, 20-40 and 40-60, the frequency vector [2,3,1], and from here the probability vector [33%, 50%, 17%]. These 2 vectors are set to the simulator as parameters, and when a VLSM engine has to be simulated we will select a random range based on the probability vector. In the example, let's imagine that the range 20-40 is chosen, from here, a random value between 20 and 40 will be generated, let's suppose we get 25, then, 25 will be the minutes added to the alert time to obtain the end datetime of the VLSM engine operation.

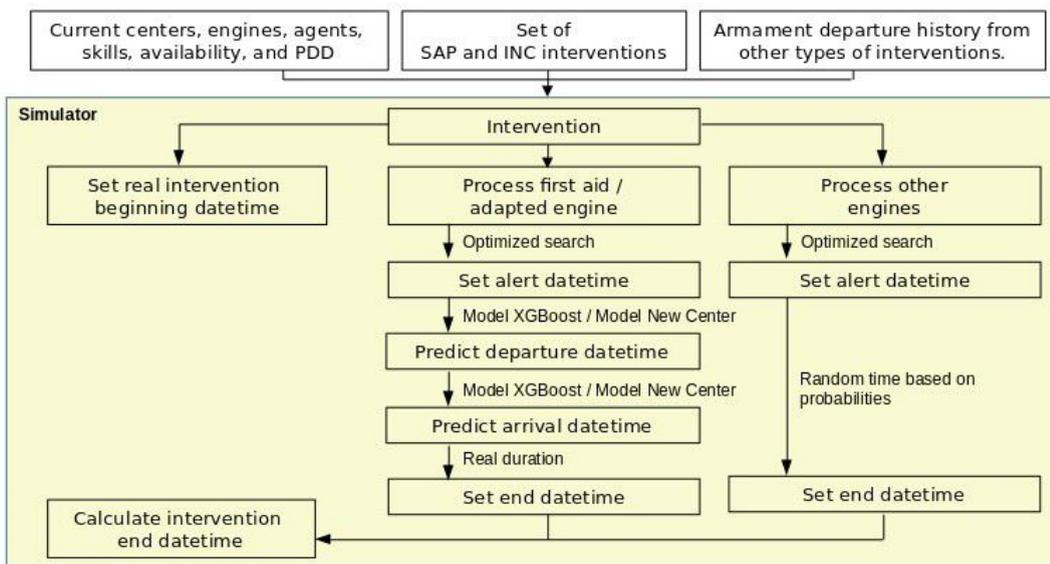


Figure 10.4: Simulator scheme.

Table 10.1: Basic armament to attend SAP and INC emergencies.

Type Interv.	Engine Mode	Engine Type	Function : Nb Agents	Skills
SAP	First Aid	VLU (VTU, VPI, VPIL, VPITU)	COND : 1 CA or EQUI: 1	VLCOND VLCA, VLEQUI
	Adapted	VSAV	COND : 1 CA : 1 EQUI : 1	VSCOND VSCA VSEQUI
INC	First Aid	VLU (VTU, VPI, VPIL, VPITU)	COND : 1 CA or EQUI: 1	VLCOND VLCA, VLEQUI
	Adapted	FPT (FPTGP, FPTHR, FPTL, FPTSR, FTLHR, CCR, CCRSR, CCFM)	COND : 1 CA : 1 CE : 2 EQUI : 2	FPCOND, CCFCOND FPCA, CCFCA FPCE, CCFCE FPEQUI, CCFEQUI

## 10.4/ DEPARTURE TIME PREDICTION

Once the center receives the alert to respond to the intervention, it may be ready to go or it may take a few minutes to organize the available armament (e.g., there are volunteer firefighters who are at home and at the moment of receiving the alert they have to go to their center, put on their protective equipment, and depart). As we know there are different types of centers and according to the quantity and availability of their armament, it is possible to recognize patterns of the time between the alert and the departure of engines. Instead of using probabilistic models, these patterns could be easily recognized by intelligent models. For this reason, we will build a ML-based departure time prediction model, specifically with the XGBoost technique, which has shown remarkable results in previous works.

The departure time prediction will be defined as a multivariate regression problem. The data set that we will use is all the registered departures of the first aid and adapted engines from 2017 to 2021. Since the resulting model will be localized within the simulator and we do not want to increase the computational time, we will build it with a reduced number of variables.

The input variables considered are the type of intervention, the coordinates of the intervention, the type of engine, and the mode of the engine. In addition, based on the datetime alert we will extract the hour, the month, the day of the week, the day of the year, and an indicator for the weekend. Also, we will add data of the centers such as the center, type of center, center coordinates, the average of the departure times according to the engine mode and center, and since the major operational load is produced by SAP and its adapted engine is VSAV, the average of the departure times of the VSAV engines per center will be considered. The target variable will be the number of minutes between alert and departure. Thus, during the simulation, our model will receive the mentioned variables and predict the minutes that we will add to the alert time to obtain the engine departure time.

As part of the preprocessing, we will encode the center, center type, motor type, motor mode, and intervention type with LabelEncoder from the Scikit-Learn library. The other variables will not be transformed since in previous experiments they presented better results without transformation.

For the creation of the models, the data will be shuffled and we will divide them into 70% for the learning phase and 30% for testing. The objective in XGBoost will be defined as *reg:squarederror*.

For model selection, we will use the Bayesian optimization algorithm *tpe.suggest* from the HyperOpt library. The search space will consider the hyperparameters *max\_depth* [15-30], *n\_estimators* [100-1000], *learning\_rate* [0.008-0.5], *subsample* [0.1-1], *colsample\_*

bytree [0.1-1], min\_child\_weight [1-30], and alpha [1-30]. We will perform 500 iterations and minimize the loss function in 10.1.

$$LOSS = 0.7 * (\log_{10}(RMSE)) + 0.3 * (\log_{10}(HISTO)) \quad (10.1)$$

Where:

RMSE is the root mean squared error of the truth target values and the estimated target values.

HISTO is the root mean squared error of two histograms. The first histogram is based on the truth target values with  $bins = 120$  and  $range = [0, 120]$ . And the second histogram is based on the estimated target values that are correct with a margin of error of 1 bin. For example, a real value is 25.2 minutes, in the first histogram this value is located in bin 25, if the predicted value is 25.8, we will consider it correct and it will belong to bin 25, if on the contrary the predicted value is 27.1, it will not be considered as a hit in any bin.

This loss function was implemented since in the first experiments with several search spaces and with  $loss = RMSE$  we observed that the metric was very good but the predictions were very focused on the mean, i.e., few maximums or minimums were recognized and we needed of these variations in the simulator. For this reason, we created the loss function 10.1, which allows us to recover a good RMSE, since it has the most weight in equation (0.7), while maintaining flexibility with HISTO (0.3).

## 10.5/ ARRIVAL TIME PREDICTION

Once the departure time of the engines has been calculated, we will need to predict the arrival time of the engines at the scene. There are several factors that could define this time, for example: vehicular traffic, weather, location of the incident, distance to the center, time of day, etc. For this reason, we again discard the idea of a probabilistic model and we will build an intelligent model with XGBoost.

The arrival time prediction will be defined as a multivariate regression problem. We will use the same data set as the departure model with the difference that the target variable is the number of minutes the engine used to arrive at the scene. Since we do not want to increase the processing time of the simulator, this model will also receive a reduced number of variables.

The input variables will be the same as the starting model, except that we will not consider the weekend indicator, since in previous experiments we did not see any impact generated, and we will add the distance variable, which is the distance between the center and

the intervention location. The distance is calculated using the `Great_Circle` method of the `GeoPy` library.

The preprocessing of the input variables is the same as for the departure model. The target variable will also not be transformed. However, there are departures recorded with arrival time equal to 0 minutes and it is very likely that it is a manual error by the agents when recording their departures. For this reason, all departures less than 1 minute have been limited to 1 minute.

For the creation of the models, the samples will be shuffled and divided into 70% for learning and 30% for testing. The objective type of XGBoost will be `reg:squarederror`.

To select the best hyperparameter configuration we will use Bayesian optimization with the `tpe.suggest` algorithm from the `HyperOpt` library. The search space will be defined by the hyperparameters `max_depth` [8-15], `n_estimators` [100-500], `learning_rate` [0.005-0.08], `subsample` [0.1-1], `colsample_bytree` [0.1-1], `min_child_weight` [1-10], and `alpha` [1-10]. We will perform 500 iterations and the defined loss function to minimize is the same as specified for the departure model 10.1.

## 10.6/ RESULTS

In this section, it will be described the results of the predictive models of departure and arrival, and the results of the models M2 and M3 generated by the simulator.

### 10.6.1/ RESULTS OF DEPARTURE TIME PREDICTIONS

In figure 10.5, on the left, it is presented the progress made by the Bayesian optimization with 500 iterations, where a minimum with `RMSE=2.73`, `HISTO=843.7`, and `LOSS=1.183` was reached and the best model was retrieved. The hyperparameters of the best configuration were `max_depth=26`, `learning_rate=0.02`, `n_estimators=702`, `subsample=0.83`, `colsample_bytree=0.58`, `min_child_weight=1`, `alpha=1`. On the right, it is shown the predictions for a set of samples using the best XGBoost model obtained. As we can see several maximums and minimums are recognized by the model. What is more, the average and standard deviation of the distribution of true targets were 5.86 and 4.21, respectively, and the distribution of predicted targets were 5.84 and 3.36, respectively, which shows quite similarity between them. This model will allow simulations to better capture the peaks in the departure times, since these are the ones that contribute to the generation of breakdowns.

In figure 10.6, we can see the feature importance of the best departure model. The intervention coordinates, the time variables (except for the weekend indicator), and the

center represent the 10 most used variables in the construction of trees by XGBoost, during the learning process. This tell us that during the day there are cycles in which the agents and engines take more time to depart from the center.

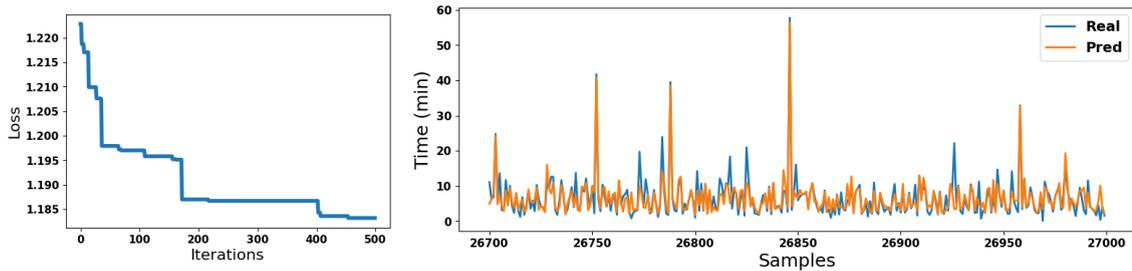


Figure 10.5: On the left, the progress of the hyperparameter optimization with HyperOpt for the depart model. On the right, the prediction results of the best depart model with XGBoost.

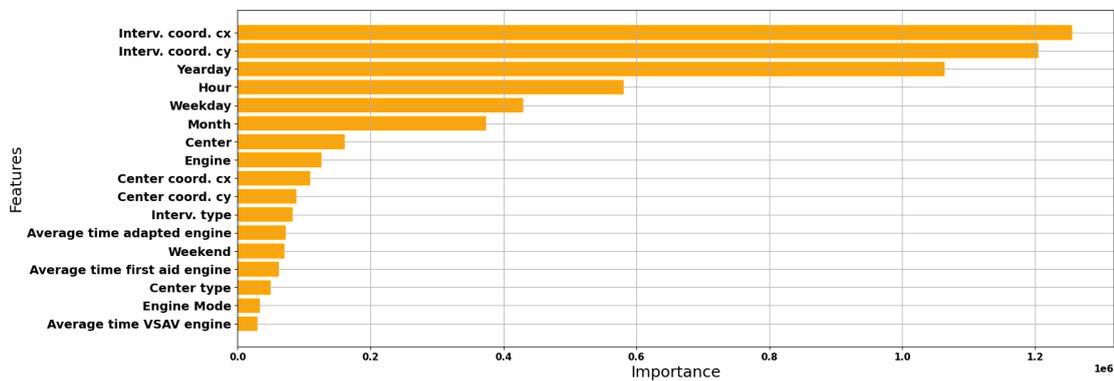


Figure 10.6: Feature importance of the best depart model with XGBoost, generated by counting the number of times a feature appears in a tree.

### 10.6.2/ RESULTS OF ARRIVAL TIME PREDICTIONS

Similar to the departure model, in Fig. 10.7, left side, we can see the optimization progress over 500 iterations, reaching a minimum of LOSS=1.42, RMSE=6.76, and HISTO=626.17. This best-performing model had the configuration max\_depth=14, learning\_rate=0.08, n\_estimators=462, subsample=0.92, colsample\_bytree=0.66, min\_child\_weight=1, alpha=1. On the right side, the predictions for a random set of samples are shown. From here, we observe that apparently fewer minimums and maximums are correctly predicted when visually compared to the predictions of the departure model. However, the mean and standard deviation of the true (7.48 and 7.89, respectively) and predicted (7.64 and 5.29, respectively) distributions are very close, so this model is useful for our simulator.

In Fig. 10.8, the feature importance is shown. Again, we see that the temporal variables

have a greater impact on the construction of the trees. Besides, it is logical to find the intervention coordinates and the distance from the center as part of the most important ones, since the farther away the center is, the longer it will take for the engine to reach the scene. However, in this period rare random events can occur such as a blown tire, an accident on the road that starts just when the engine is about to pass, the appearance of an animal on the road such as a deer, cow or horse that block the way, etc. For this reason, in the simulator and immediately after making the arrival time prediction, we will include a noise that increases the arrival time. According to historical data since 2017, the largest peaks have occurred with a probability of 2%, a value that will be defined in the simulator to generate the noise.

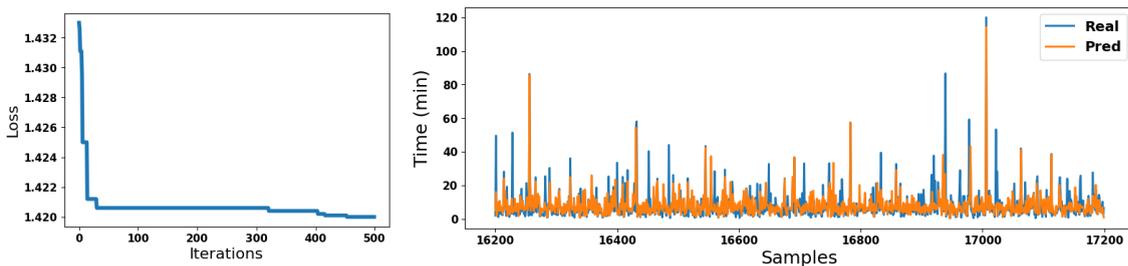


Figure 10.7: On the left, the progress of the hyperparameter optimization with HyperOpt for the arrival model. On the right, the prediction results of the best arrival model with XGBoost.

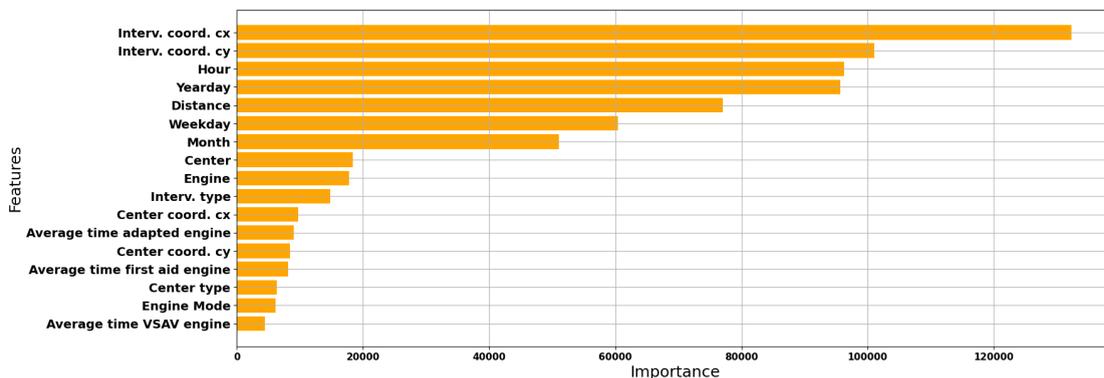


Figure 10.8: Feature importance of the best arrival model with XGBoost, generated by counting the number of times a feature appears in a tree.

### 10.6.3/ RESULTS OF BASE MODELS

Once all parts of the simulator were integrated, we launched the simulation for the period from 01/06/2020 00:00:00:00 to 01/10/2020 00:00:00:00, 4 months in total. The data used come from the SDIS25 internal database described in Chapter 3.

In Table 10.2, we detail the number of breakdowns by case and model. In Fig-

ures 10.9, 10.10, and 10.11, it is shown the number of interventions per 5-minute intervals for the departure, arrival and total response times of the first aid engines, where the total response is the sum of departure and arrival. In the same way, Figures 10.12, 10.13, and 10.14 show the 3 times for the adapted engines. Furthermore, on the top of these 6 figures, we depict the average (Avg) and standard deviation (Std) for the 3 times and by model. Fig. 10.15 presents the breakdowns of public service (BPS, cases from C2BIS to C1) as red points on the Doubs department map, administration of the SDIS25. On the top of the figure, we can see the model name and the number of BPS (Nb BPS).

From Table 10.2, on the one hand, we can see that the M2 simulation model recognized 914 fewer R1 cases than the real M1. What is more, a large part of the 914 cases have been identified as a simple delay R3. This would mean that the arrival prediction model would have to be improved. On the other hand, as we assumed at first and confirmed by looking at the results, the M3 model has fewer breakdowns of public service because the operational burden generated by the AVP, OD, and RTN interventions was removed. However, the M3 model maintains the pattern of breakdowns generated in this period.

Despite the aforementioned differences, the simulation results demonstrated 70% of accuracy. This accuracy was calculated using the `classification_report` method from the Scikit-Learn library, where the true and predicted labels were the breakdowns cases (R1, R2, R3, C2BIS, etc) of each intervention processed in M1 and M2, respectively. This good accuracy is due to the fact that the departure, arrival, and total response times of the first aid and adapted engines are very similar to the actual times. This can be clearly seen in Figures 10.9, 10.10, 10.11, 10.12, 10.13, and 10.14, where the average and standard deviation of the real and simulated distributions are close. What is more, in Fig. 10.15, we can visualize that the breakdown concentration pattern of the base model is maintained in the 2 simulation models.

Table 10.2: Number of breakdowns per model and per case.

Model	R1	R2	R3	C2BIS	C2TER	C2	C3BIS	C3TER	C3	C1BIS	C1TER	C1
M1	7655	38	816	49	9	554	29	10	121	3	1	35
M2	6741	46	1753	58	1	580	16	3	54	14	2	52
M3	6771	34	1831	55	2	513	12	3	37	12	0	50

## 10.7/ DISCUSSION AND RELATED WORK

In the literature, we can find several works in which simulators are modeled for resource optimization. For example, in [142], a simulation-based optimization framework is proposed, where the simulator built mimics the call center call arrival process using the Gaussian Mixture Models (GMM) clustering method [140], and the arrival times at the scene, arrival at the hospital, and return to the center are determined by a probability dis-

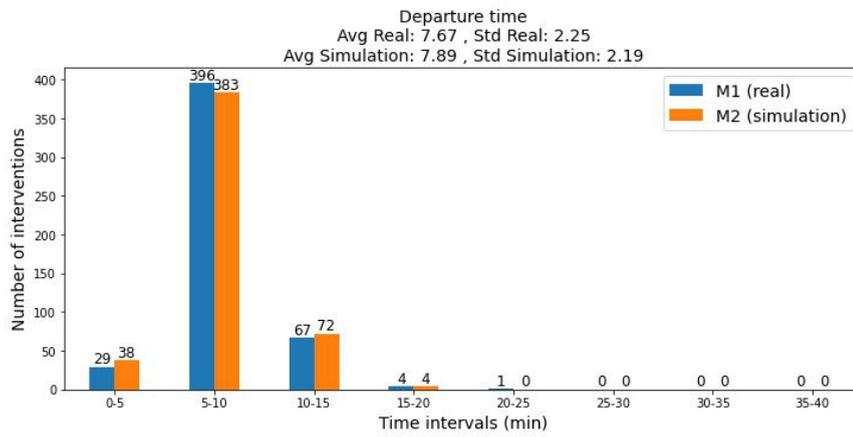


Figure 10.9: Comparison of real and simulated departure times of first aid engines by 5-minute intervals.

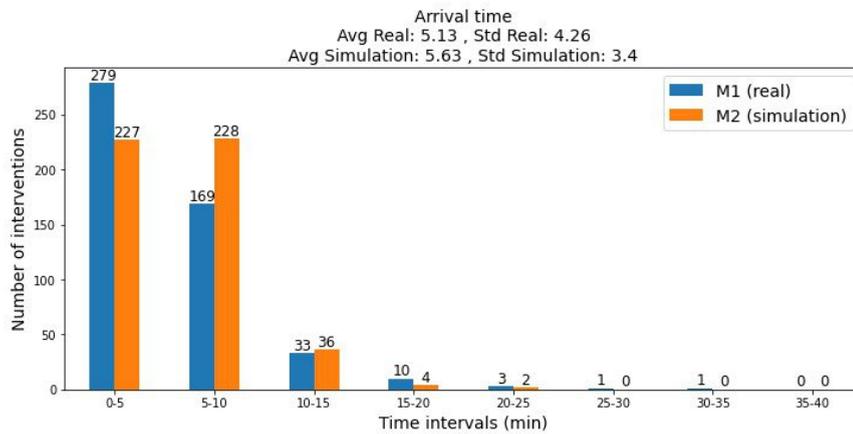


Figure 10.10: Comparison of real and simulated arrival times of first aid engines by 5-minute intervals.

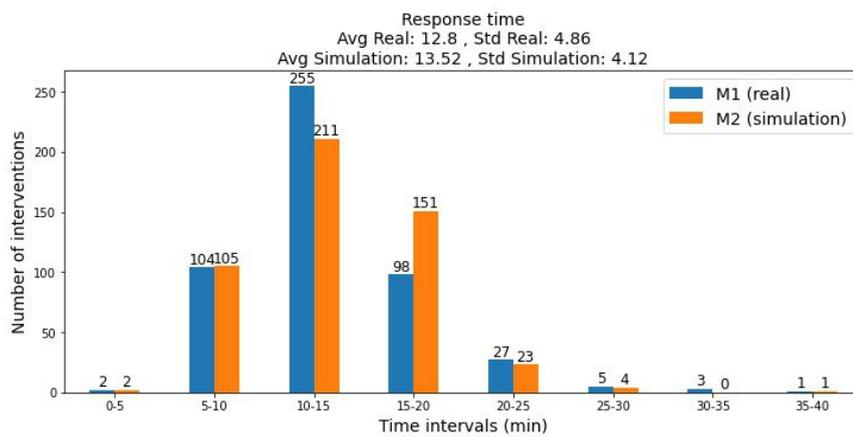


Figure 10.11: Comparison of real and simulated response times of first aid engines by 5-minute intervals.

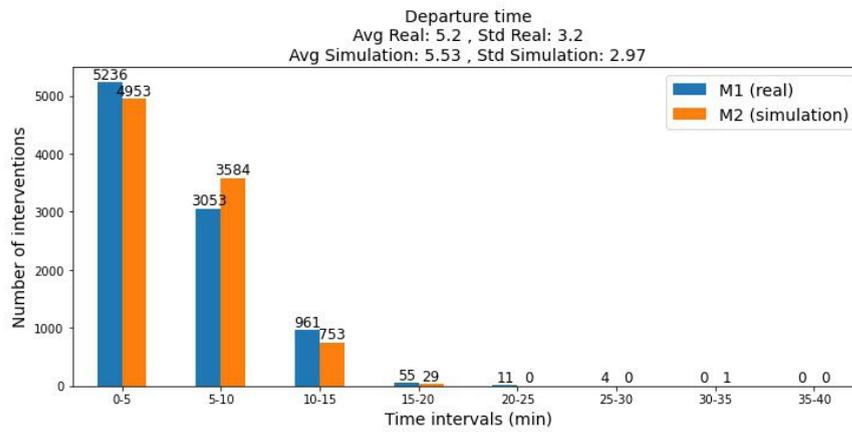


Figure 10.12: Comparison of real and simulated departure times of adapted engines by 5-minute intervals.

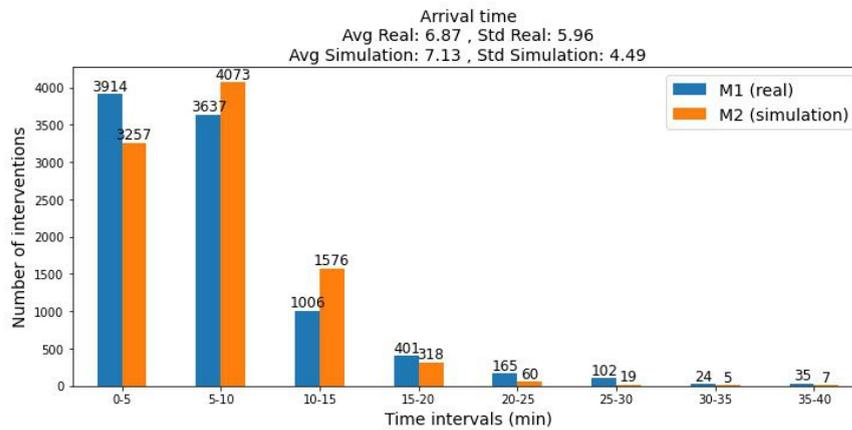


Figure 10.13: Comparison of real and simulated arrival times of adapted engines by 5-minute intervals.

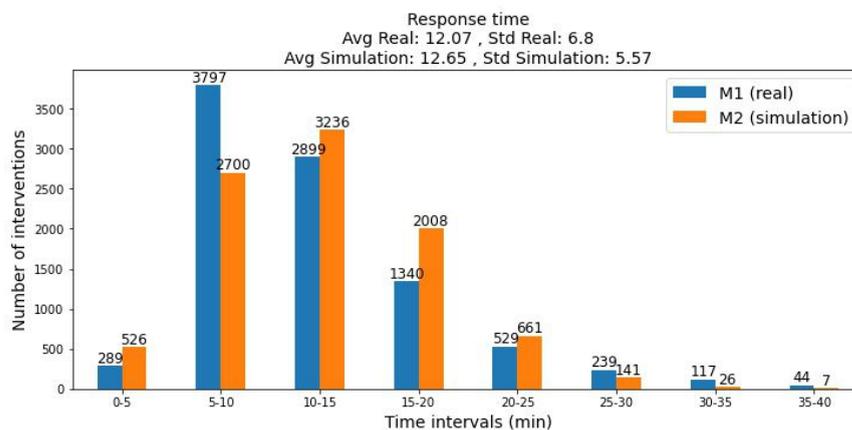


Figure 10.14: Comparison of real and simulated response times of adapted engines by 5-minute intervals.

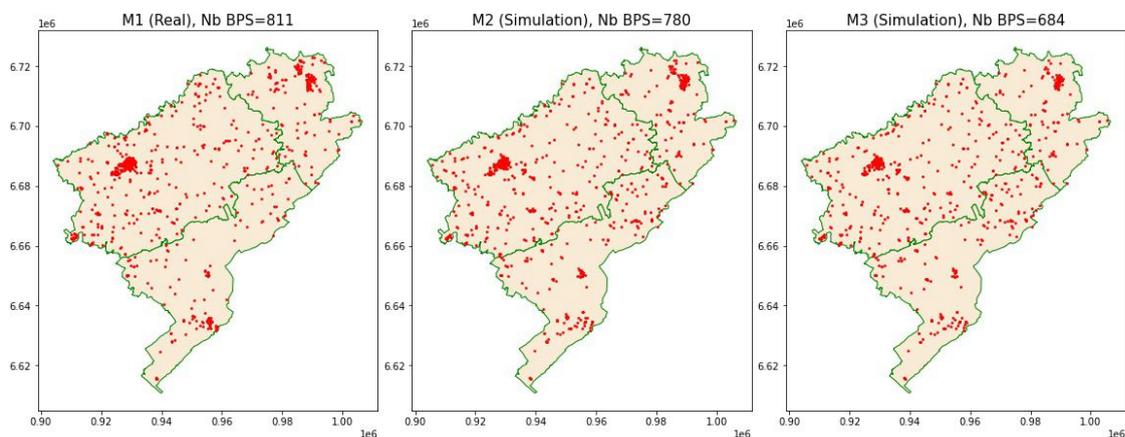


Figure 10.15: Comparison of the distribution of BPS between M1, M2, and M3, over the Doubs territory.

tribution based on the analysis of their historical data. In [147], addresses the Ambulance Location and Dispatching Problem, where a recursive simulation-optimization framework is constructed with the objective of minimizing the total response time to satisfy the demand. Different from [142], the simulation model is represented with a graph and the arrival times to the scene and to the hospital are not determined. In [214], a dynamic model was built for resource relocation, i.e., new positions are suggested for agents and engines as each iteration improves "the preparedness". The preparedness is the estimate of how prepared the armament is to respond to an accident.

In our case, our simulator mimics the operational process of a set of interventions attended in a real period, i.e., from the beginning of the intervention, through the time of alert to the centers, arrival of the engine to the scene, and end of its participation. The recognition of these times is necessary to subsequently apply our service quality indicator. Basically, our simulator uses real parameters, probabilities of occurrence, and ML-based models. These intelligent models are the heart of our simulator and are one of the main differences with the works found in the literature [125, 142, 147, 214].

Thus, in the next chapters, our simulator will be integrated to several modules in charge of optimizing the resources for a specific task, and with the help of our quality indicator, we will evaluate each solution until we identify the best one.

## 10.8/ CONCLUSION

In this chapter we have explained the implementation of the operational load simulator. For this, first, we have described the variables that influence the simulator such as the centers, the types of engines, the number of agents, the availability of the agents, etc. Second, we have defined the internal structure of the simulator, the reference model M1,

which is the breakdown calculation applied to real times, the simulation model M2, which simulates SAP and INC interventions and considers the operational load of other interventions, and the simulation model M3, which only simulates SAP and INC interventions. Inside the simulator, we used the real start of the intervention and calculate its end based on the last engine used in the intervention. Engines were mainly identified in 3 categories: first aid, adapted, and others. We calculated the alert, departure, arrival, and end times for the first aid and adapted engines, since these times will be needed when we perform the breakdown calculation. In the case of the other engines, we will only identified their alert and end time. To recognize the alert time we used the armament optimized search algorithm. For the departure and arrival times, we created two ML-based models that predict the minutes that we will use to calculate these times. To determine the end of operation of an engine, we used true duration for first aid and adapted engines and probabilities for other engines.

Thus, we simulated the period from June to September 2020 and obtained 70% accuracy. This result shows that our simulator can capture patterns of the process that occurred in real life. We will use our simulator with the M3 model to perform resource optimization in the following chapters.

# OPTIMIZATION OF ENGINE DISTRIBUTION

In the previous Chapter, we developed an operational load simulator. In this Chapter, we will use this simulator to optimize the distribution of engines in the centers of the Doubs territory.

## 11.1/ INTRODUCTION

Over the years we have seen an increase in the operational load and its variation by periods. This means that the positioning of resources in general must be dynamic to cover interventions according to time and areas. The main idea is to arrive in the shortest possible time to the victim's location and provide first aid.

In our case, there are 2 types of breakdowns identified BSC and BPS, being BPS the most critical type, since it implies that there are no resources in a center to attend another intervention, i.e., if there is an adapted engine attending an intervention and simultaneously another adapted engine available in a nearby center, the latter would reduce the response time to another intervention in the same or nearby location. There are other factors that influence the response time of an engine such as weather, traffic congestion, strikes, etc. However, the main factor remains the resilience of the centers to maintain their coverage.

For this reason, in this chapter, we will simulate the real operational load of a period using the simulator developed in the previous chapter and testing various assignments of the number of engines to each center, where the number of agents and their real availability will be kept fixed. Thus, we will discover if by using fewer engines but distributed in an optimal way we can reduce the number of breakdowns of public service. Since the number of SAP type interventions represents the largest operational load, the distribution of its adapted engine VSAV will be optimized.

## 11.2/ METHODOLOGY

Fig. 11.1 describes the optimization process for the allocation of VSAV engines. For this, we will use the M3 simulation model, where the input data are the SAP and INC interventions, the deployment plan for this period, and the real location of the 71 centers. The agents will maintain their position, training, and real availability in each center. Similarly, the first aid engines (VTU, VLU, VPIL, etc.), the adapted engines for INC (FPT, CCFM, FPTGP, etc), and other engines (e.g., VLSM, EPA, CCGC, etc.) will keep their original distribution in the centers.

For the Bayesian optimization, we will establish the limits of the search space with minimum and maximum values for the number of VSAVs in each center. The centers defined in the search space will be of types CSP, CSR, and CS, which have trained agents that are able to depart with a VSAV. For each iteration, the tpe.suggest algorithm from Hyper-Opt will generate a new VSAV distribution that will be simulated. After each simulation, we will apply the breakage calculation and generate breakdown statistics for each center. Then, we will compute the loss. After B iterations, we will retrieve the best solution.

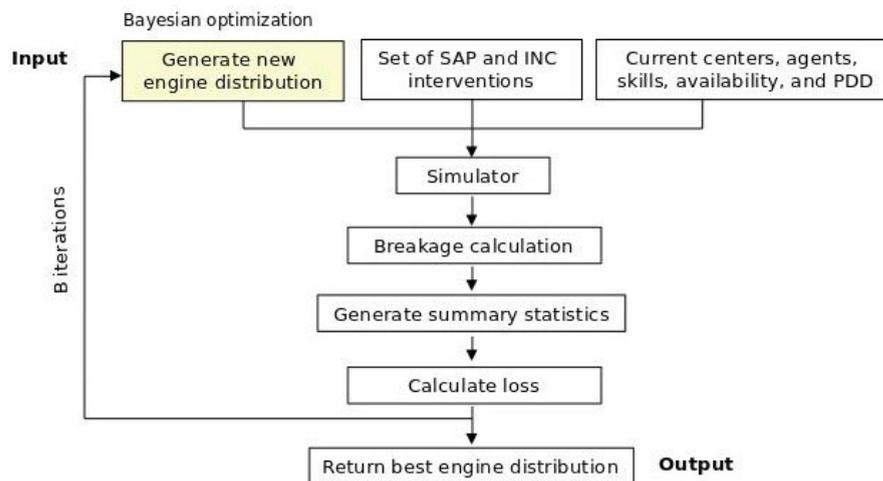


Figure 11.1: Optimization scheme for engine distribution.

In Bayesian optimization, it is necessary to define a loss function to guide the exploration and exploitation of regions in the search space. This helps to quickly converge on a good solution. Thus, new configurations of better quality are generated until a better solution is obtained according to a stop condition. In our case, 11.2a represents the loss composed of piecewise functions. The main idea in 11.2a is to reduce the number of BPSs using the least amount of VSAV engines. For this, we first validate if the difference of the actual BPSs number and the simulated BPSs number is greater than zero, if true, it would mean that we have reduced the number of BPSs, then we would give more weight to the number of VSAV to be reduced. If on the contrary the difference is less than zero, it would mean

that during the simulation we have obtained more BPSs with the new configuration, then we will give more weight to the amount of BPSs to reduce it. Finally, if there was no difference, the weight would be the same for the number of VSAV and the number of BPSs.

Sets:

$C$ : set of centers.

$I$ : set of interventions SAP and INC.

$ET$ : set of engine types

Variables:

$Nbps_c^s$ : number of BPSs in center  $c$  during simulation  $s$ .

$Nbps_c^r$ : number of BPSs in center  $c$  and in real environment  $r$ .

$Nvsav_c^s$ : number of engines VSAV in center during simulation  $s$ .

$Npi^s$ : number of processed interventions during simulation  $s$

$Npi^r$ : number of processed interventions in real environment

$Nabe_i^s$ : number of adapted backup engines in intervention  $i$  during simulation  $s$

$Nae_i^s$ : number of adapted engaged engines in intervention  $i$  during simulation  $s$

$Nfe_i^s$ : number of first aid engines in intervention  $i$  during simulation  $s$

$Noe_{it}^s$ : number of other engines by type  $t$  and in intervention  $i$  during simulation

$Noe_{it}^r$ : number of other engines by type  $t$ , in intervention  $i$ , and in real environment

Parameters:

$\alpha$ : weight parameter for existing difference between  $Nbps_c^r$  and  $Nbps_c^s$ .

$\beta$ : weight parameter for not existing difference between  $Nbps_c^r$  and  $Nbps_c^s$ .

$B$ : number of iterations in Bayesian optimization.

$$Loss = \begin{cases} \alpha * \sum_{c \in C} Nbps_c^s + (1 - \alpha) * \sum_{c \in C} Nvsav_c^s & \text{if } \sum_{c \in C} Nbps_c^r - \sum_{c \in C} Nbps_c^s > 0 \\ \beta * \sum_{c \in C} Nbps_c^s + (1 - \beta) * \sum_{c \in C} Nvsav_c^s & \text{if } \sum_{c \in C} Nbps_c^r - \sum_{c \in C} Nbps_c^s = 0 \\ (1 - \alpha) * \sum_{c \in C} Nbps_c^s + \alpha * \sum_{c \in C} Nvsav_c^s & \text{if } \sum_{c \in C} Nbps_c^r - \sum_{c \in C} Nbps_c^s < 0 \end{cases} \quad (11.1)$$

$$\min \quad Loss \quad (11.2a)$$

$$\text{s.t.} \quad Npi^r = |I|, \quad (11.2b)$$

$$Npi^s = Npi^r, \quad (11.2c)$$

$$Nabe_i^s = 1 \quad \forall i \in I, \quad (11.2d)$$

$$Nae_i^s = 1 \quad \forall i \in I, \quad (11.2e)$$

$$Nfe_i^s \in \{0, 1\} \quad \forall i \in I, \quad (11.2f)$$

$$Noe_{it}^s = Noe_{it}^r \quad \forall i \in I, t \in ET. \quad (11.2g)$$

The minimization of the loss function 11.2a is subject to constraints such as the number of interventions processed in the real and simulated environment must be equal to the set of input interventions ( 11.2b and 11.2c). This validates that all the operational load has been processed. Also, there must be at least one adapted engaged engine and one adapted backup engine per processed intervention ( 11.2d and 11.2e). In the case of first aid engines, if a first aid engine was present during the real intervention, the simulation will attempt to replicate the departure of this engine, but if an adapted engine is available before it, the first aid engine will no longer be considered 11.2f. The number of other engines used during the intervention in the simulation should be equal to the number used in the real environment 11.2g, with the difference that the number of agents used to assemble these engines is the minimum (degraded mode), this gives flexibility to the model to reach a solution.

### 11.3/ RESULTS

Set  $C$  is composed of the 71 centers deployed in the territory. Set  $I$  contains 9320 interventions that took place in the period from 01/06/2020 00:00:00:00 to 01/10/2020 00:00:00:00, of which 8571 are SAP and 749 INC. Set  $ET$  is composed of 163 types. For the Bayesian optimization, its search space is composed of 39 centers, the number of iterations  $B$  is 500, and we added 3 initial trials. For the loss function, we set the parameters  $\alpha$  and  $\beta$  with values 0.3 and 0.5, respectively.

Table 11.1: Results of the 3 best solutions for the distribution of VSAVs.

Model	BPS decrease	Nb BPS	BPS Avg. time	Nb BPS SAP	Nb BPS INC	Nb VSAV	Loss
M3 (baseline)	-	684	11.88 min	527	157	64	-
Solution 451	45.91%	370	10.22 min	209	161	78	165.6
Solution 243	33.92%	452	9.27 min	293	159	64	180.4
Solution 132	29.39%	483	11.17 min	319	164	59	186.2

In Table 11.1, we depict the summary of the best solutions. In the first line, we present statistics on the base model M3, where the number of breakdowns of public service (Nb BPS) is 684 with a mean time (BPS Avg. Time) of 11.88 minutes and a total number of VSAV (Nb VSAV) equal to 64, spread over the 71 centers. In the second line, we summarize solution 451, which represents the lowest loss value among all iterations (165.6). This solution reduced the Nb BPS from 684 to 370, this represents a 45.91% decrease of breakdowns. In the third line, we summarize solution 243, which represents the lowest loss value among all solutions that used 64 VSAVs, i.e., the same amount of VSAVs as

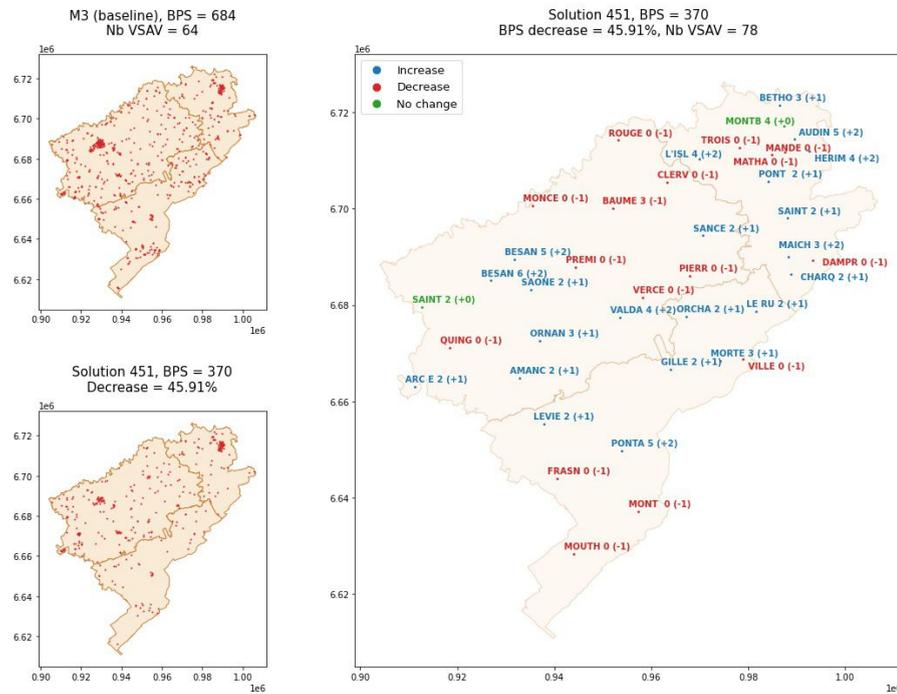


Figure 11.2: The best solution with the minimum number of RSP. On the left side, the red points indicate the RSP distribution in the Doubs for M3 and the solution. On the right side, it is shown the increment, reduction or no change of VSAV by center.

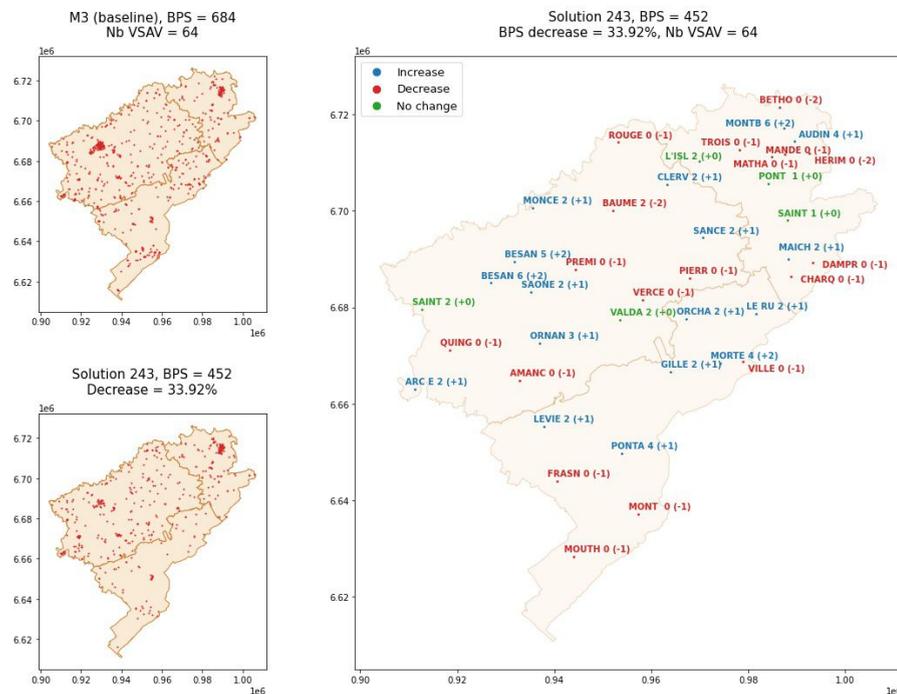


Figure 11.3: The best solution using the original quantity of VSAV. On the left side, the red points indicate the RSP distribution in the Doubs for M3 and the solution. On the right side, it is shown the increment, reduction or no change of VSAV by center.

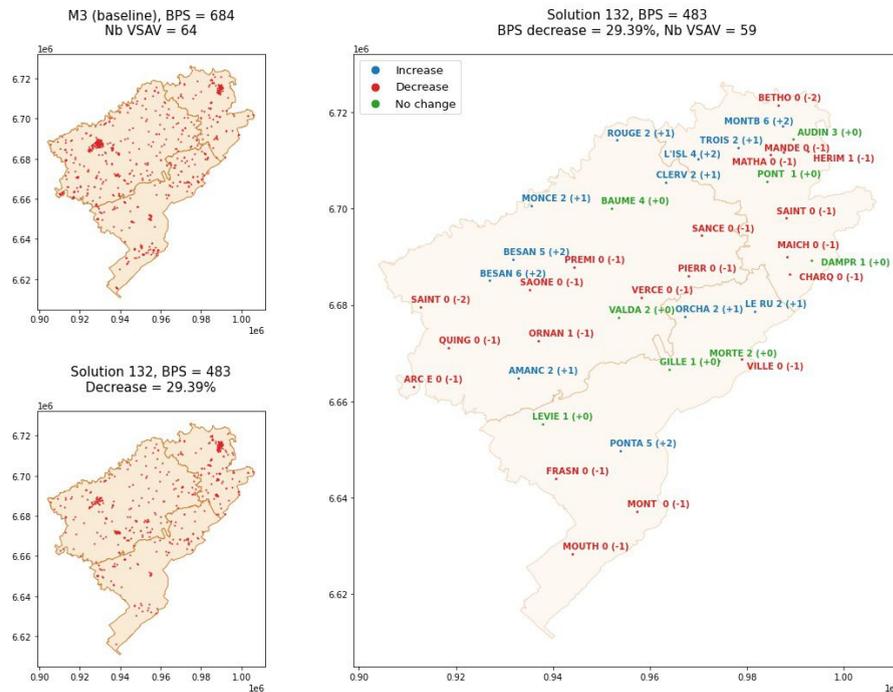


Figure 11.4: The best solution using less VSAV than the original number. On the left side, the red points indicate the RSP distribution in the Doubs for M3 and the solution. On the right side, it is shown the increment, reduction or no change of VSAV by center.

the M3 model. The BPS reduction was 33.92%. And in the fourth line, we summarize solution 132, which denotes the lowest loss value using less than 64 VSAVs, i.e., using fewer VSAVs than those used in the M3 model. This solution reduced the Nb BPS by 29.39% and shows that it is possible to use fewer VSAV resources to obtain less BPSs.

However, we should remember that the distribution of the centers and the deployment of their resources in the territory is an interconnected network; hence, the modification of resources in a center affects the entire chain. Thus, in Table 11.1, we see that in general the BPS Avg. time has been reduced in the 3 solutions and although the number of breakdowns of public service in SAP (Nb BPS SAP) also decreased, this did not happen for Nb BPS INC. The reason why Nb BPS INC did not decrease is because, first, we only optimized the engine adapted for SAP (VSAV) and, second, because several of the firefighters who were part of the crew on the adapted engines for INC in M3 became part of the adapted engines for SAP, since these ones increased in number and the current agents had the enough training to depart with them.

Fig. 11.2 shows 3 maps with the allocation of VSAV and the generated BPSs for the 39 simulated centers. In the top left map, we can see the distribution of the 684 BPSs generated with the M3 base model. In the lower left map, we see the distribution of the 370 BPSs obtained in the 451 solution. In the map on the right, the allocation of the VSAV engines to the 39 simulated centers is depicted, where blue represents the increase of

VSAV, red the reduction of VSAV, and green that no change occurred. Fig. 11.3 and Fig. 11.4 follow the same description. From the 3 solutions, we deduce that the west side of the region requires more VSAV engines that could come from the south side. In the northeast we can see that the reduction of breakdowns is not very pronounced, this suggests that not only engines but also agents are needed.

## 11.4/ DISCUSSION AND RELATED WORK

In the literature, resource allocation problems such as ambulance location are considered NP-Hard [19] [7]. We can find deterministic models represented by graphs that reflect poorly the uncertainty of the problem. For example, early works such as the location set covering model (LSCM) and the maximal covering location problem (MCLP) in [2] did not consider that an ambulance could be occupied or that various types of ambulances could be used, while the Cooperative Location Set Covering Problem and the Cooperative Maximum Covering Location Problem in [31] expanded the definition of coverage allowing all units to cover all demand sites. As a way to capture uncertainty regarding capacity, demand, and time, probabilistic models appeared. Among the first probabilistic models are the stochastic set covering design in [3] and the Maximum Expected Coverage Location Problem (MEXCLP) in [4], where the positions of the interventions are randomly generated and the *busy fraction* is created to denote the unavailability of an ambulance in an emergency. The MEXCLP model has several variations as in works [10], [8], [38], [45], and [49]. Finally, if we want to enhance the coverage by reallocating resources in real time, dynamic models could be applied more effectively. The works in [72], [9], and [63] propose to reduce the travel time of the ambulance, reposition the vehicle every 24 hours, and implement the concept of "preparedness" as an indicator of the ability of resources. Among the most current works we find [173] and [95], which are extensions of the double standard model (DSM) developed in [16]; and [189], where they present a Simulation-Optimization (SO) approach that simultaneously optimizes resource planning decisions and ambulance relocation in the emergency process.

Compared to the literature reviewed, our model is dynamic and has the advantage of integrating ML-based models to predict departure and travel times based on a history, different from previously developed models where these times are parameters defined from the beginning of the process or calculated based on distances but without considering temporal and other external variables related to the territory. In addition, in several works, we have seen that the main objective is to reduce the ambulance response time and in some they have defined an indicator called *preparedness*, which could be an equivalent to our breakage calculation. These indicators allow us to recognize that in order to reduce the response time we must have not only a well-positioned armament but also an avail-

able armament. Thus, in our case, our objective function for VSAV engines' reallocation considers that to reduce the response time we have to reduce the number of breakdowns that are the cause of the increase in response time and at the same time to consume few resources.

## 11.5/ CONCLUSION

In this chapter, we optimized the distribution of VSAV engines to 39 centers from a total of 71 centers in the Doubs territory. For this, we used Bayesian optimization with 500 iterations, each iteration simulated a new VSAV distribution configuration with the operational load of a real period and the real training of firefighters. The defined loss function consists of 3 functions, where weights are assigned to the number of BPSs and the number of VSAV engines, since the main idea is to reduce the breakdowns using the least amount of resources.

The optimization results show that it is possible to reduce the breakdowns using less resources or the same amount of resources but with a different configuration than the original one. Also, we were able to identify the centers that even if we add VSAVs to them, their breakdowns would not decrease since these centers not only need engines but also trained agents.

# OPTIMAL POSITIONING OF A NEW CENTER AND ITS CONFIGURATION

In the previous Chapter, we have performed the optimization of the engine distribution, more specifically, VSAV engines for SAP interventions. In this Chapter, we will optimize the positioning of a new center and its engines configuration.

## 12.1/ INTRODUCTION

As we have been able to validate in the previous chapter, the optimization of the engines distribution greatly reduces the BPSs. However, there are times when SDIS25 sees the need to implement a new center to extend its coverage, given that over the years the cities evolve and the SDIS25 strategy must be adapted.

The implementation of a new center involves identifying the areas with the highest number of emergencies in order to position the new center, defining the type of center according to its resources and demand, and structuring the training and availability of its agents. Once the new center is positioned, a new deployment plan is generated. The deployment plan is a guide that specifies the mobilization of all SDIS25 centers according to the commune-quartier and zone; its format is illustrated in 12.1. This guide is elaborated considering the distances between the centers and the commune-quartiers, in addition, it considers the operational load and risks that may occur in the commune-quartiers. In this way, it seeks to maintain the coverage of the territory and protect the population.

Therefore, the main objective of this chapter is to develop a methodology to position a new center and to define its resources (engines) using our simulator.

Table 12.1: Illustration of the Deployment Plan.

Commune	Quartier	Zone	Center 1	Center 2	...	Center 20
Abbans-Dessous	Abbans-Dessous	Z3	Bouss	Quin	...	Montbel
Abbans-Dessous	Abbans-Dessous	Z3	Quin	Bouss	...	Montbel
...	...	...	...	...	...	...
Audincourt	Audincourt	Z1	Audin-Val	Audin-Val	Pontar	...
...	...	...	...	...	...	...
Voujeaucourt	Belchamp	Z2	Montbel	Audin-Val	Math	Pontar

## 12.2/ METHODOLOGY

This methodology proposes 2 stages for the positioning of a new center.

**Stage 1: recognition of the operational load to be absorbed by the new center according to its position.** Fig. 12.1 shows the scheme of the methodology proposed for this stage. We will use a set of SAP and INC interventions occurred in a specific period, we will guide the learning of the best explored zones with Bayesian optimization (HyperOpt), and in each iteration, the selected position will generate a new deployment plan. This deployment plan will help us to calculate the distribution of the operational load by first intervention center (Center 1 in Table 12.1) and by commune-quartier. The operational load absorbed by the new center will represent our loss function that we will maximize. The more operational load absorbed by the new center, the better its position will be.

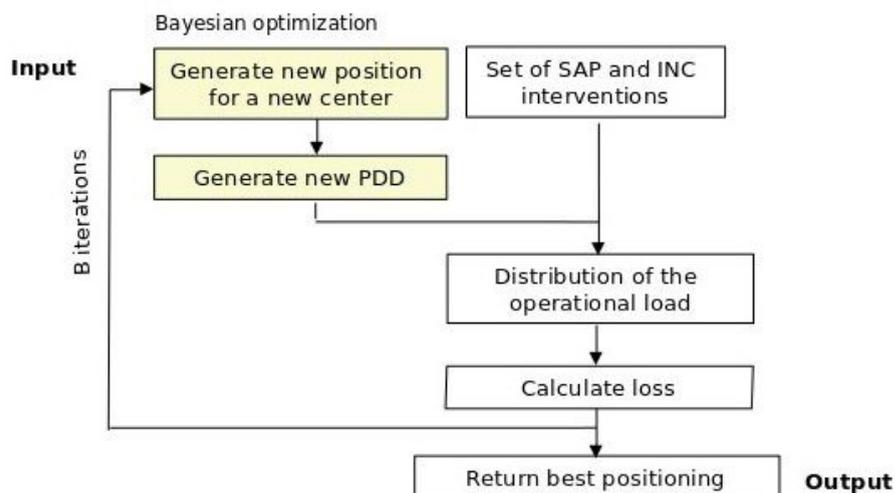


Figure 12.1: Schema of the search for the best position for the new center (stage 1).

This process is detailed in Algorithm 1, where we initially build a grid over the west zone of the territory. The western zone is where the capital of the Doubs territory, Besançon, is located, it is one of the zones with the highest flow of emergencies and breakdowns, and it is the most suitable zone to establish a new center. Starting from line 2, Bayesian will choose a square of the grid as a possible area to implement a center, we will calculate the

centroid of the square and this will be the position of the new center. Within a Bayesian iteration, we will iterate through all the commune-quartiers of the original PDD and through all the defense centers of each commune-quartier with the objective of replacing the old center by the new center, if the distance between the centroid of the community-quartier and the new center is smaller than with the old center. Thus, we will create a new list of defense centers for each community-quartier and thus a new simulated PDD. Starting from line 17, we will iterate for each intervention to calculate the total number of interventions per commune-quartier. From line 21, for each commune-quartier of the simulated PDD, we will calculate the total number of interventions that would be absorbed by the new center as the first defense center, and since we seek to maximize the absorbed operational load and HyperOpt minimizes, we will change the sign of the loss function. Finally, after  $B$  iterations, we will obtain the best position for the new center.

Sets:

$I$ : set of interventions SAP and INC.

$GS$ : set of squares' geometries (grid).

$GCQ$ : set of commune-quartiers' geometries.

Variables:

$pdd^r$ : dictionary of real deployment plan.

$pdd^s$ : dictionary of simulated deployment plan.

$nbinterv$ : dictionary of the interventions' number by commune-quartier.

$cq$ : a commune-quartier.

$LC_{cq}^r$ : list of defense centers of a commune-quartier in original environment.

$LC_{cq}^s$ : list of defense centers of a commune-quartier in simulated environment.

$ncenter$ : new center.

$ocenter$ : old center.

$p^*$ : best solution for new position.

Parameters:

$B$ : number of iterations in Bayesian optimization.

**Algorithm 1** Positioning of the new center

---

**Input :**  $I, GS, GCQ, B, pdd^r$   
**Output :**  $p^*$

- 1:  $b = 0$
- 2: **Start HyperOpt:** choose  $gs$  from  $GS$
- 3: **for**  $cq$  in  $pdd^r$  **do**
- 4:      $nbinterv[cq] = 0$  ▷ Initialization of interventions' number by  $cq$ .
- 5:      $LC_{cq}^r = pdd^r[cq]$
- 6:      $LC_{cq}^s = \text{copy}(LC_{cq}^r)$
- 7:      $d_{ncenter} = \text{distance}(\text{centroid}(gs), \text{centroid}(GCQ(cq)))$
- 8:     **for**  $ocenter$  in  $LC_{cq}^r$  **do**
- 9:          $d_{ocenter} = \text{distance}(\text{coordinates}(ocenter), \text{centroid}(GCQ(cq)))$
- 10:         **if**  $d_{ncenter} < d_{ocenter}$  **then**
- 11:              $LC_{cq}^s.\text{replace}(ocenter, ncenter)$
- 12:             **break**
- 13:         **end if**
- 14:     **end for**
- 15:      $pdd^s[cq].\text{add}(LC_{cq}^s)$  ▷ Adding new defense list for  $cq$ .
- 16: **end for**
- 17: **for**  $i$  in  $I$  **do**
- 18:      $cq = \text{locate}(i)$  ▷ Locate the  $cq$  of the intervention.
- 19:      $nbinterv[cq] += 1$  ▷ Increase interventions' number by  $cq$ .
- 20: **end for**
- 21:  $loss = 0$
- 22: **for**  $cq$  in  $pdd^s$  **do**
- 23:      $LC_{cq}^s = pdd^s[cq]$
- 24:     **if**  $LC_{cq}^s[0] == ncenter$  **then** ▷ Add up the operational load for  $cq$ .
- 25:          $loss += nbinterv[cq]$
- 26:     **end if**
- 27: **end for**
- 28:  $\text{save}(loss, ncenter, \text{centroid}(gs), pdd^s)$  ▷ Save all data related to new center.
- 29:  $loss = -loss$  ▷ Change sign to minimize
- 30:  $b += 1$
- 31: **End HyperOpt:** read  $loss$ , stop  $b == B$
- 32:  $p^* \leftarrow$  best solution of HyperOpt ▷ Retrieve best solution

**return :**  $p^*$

---

**Stage 2: search of the best configuration for the new center.** Fig. 12.2 shows the outline of the methodology developed for this stage. We will use the same set of interventions from stage 1 and all the existing centers, engines, and agents for the specified period. Since in this stage we will simulate the behavior of the entire network of centers including the new center with the operational load of the period, we must replace the actual PDD by the simulated PDD, obtained with the position of the new center in stage 1. For the new center to be considered available, it must have engines and according to their types and quantity, it will belong to a center type. To define the size of the center, we will set up different configurations with the most representative engines, for example:

the VTU engine will represent the first aid engines, the VSAV engine will represent the engines adapted for SAP interventions, and the FPT engine will represent the engines adapted for INC interventions. Engines without agents cannot depart, so it is necessary to define the sizing of the agents. However, in the present chapter, we will assume that the number of agents per day depends on the number of engines owned, i.e., using Table 10.1, if the new center has 1VSAV it needs 3 firefighters, if the new center has 1VSAV + 1FPT it needs 9 firefighters, and so on. Firefighter training will be with respect to the functions required to operate the engines as indicated in Table 10.1 too. The availability of the agents will be 24h for every day of the period. We know that this maximum availability would not be applied in real life, but it will help us to identify the best engine configuration that will help to reduce total breakdowns.

From a set of  $F$  defined configurations, we will test each one in one iteration. Thus, in each iteration, all the above data will be processed by the simulator with a specific configuration for the new center. Then, we will apply the quality indicator, and we will obtain a loss which will be the total number of BPSs generated. Finally, we will analyze the results of each configuration to choose the best one.

The advantage of having 2 stages allows the first stage to be omitted if SDIS25 has already analyzed or has available a new position for a center and a new PDD that they would like to simulate only in the second stage to validate its impact. This flexibility is necessary when building the system and bringing it to real life, since the implementation of a new center is too costly and SDIS25 needs a broad vision to make a better decision.

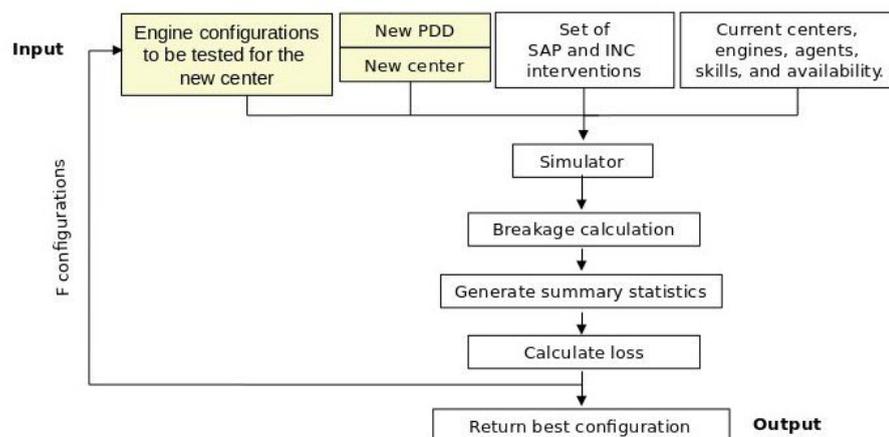


Figure 12.2: Schema of the search for the center configuration (stage 2).

## 12.3/ RESULTS

Set  $I$  consists of 9320 interventions that occurred in the period from 01/06/2020 00:00:00 to 01/10/2020 00:00:00. Set  $GS$  consists of 900 squares (30x30 grid) minus those outside

the territory. The set  $F$  is composed of 12 configurations. In stage 1, the distances are calculated using the `Great_Circle` method of the `GeoPy` library, in the future this could be replaced by distances generated by libraries such as `Google-Maps` or `OpenStreetMap`.

The obtained results are described in Table 12.2, where `BPS decrease (%)` represents the decrease in percentage of the number of BPSs compared to Baseline (last line of the table), `Nb BPS` represents the total number of BPSs, `Nb BPS SAP` and `INC` are the total number of BPSs for SAP and INC, respectively. The best results are in bold.

On the one hand, we have the center named NC-AU (new center based on an automated decision) which represents the best solution found during stage 1. In its stage 2, 13 configurations were tested. The first 8 configurations represent various combinations of engines with respect to the existing center types and the last 5 configurations are random tests without identified center type (X configuration), in order to measure how far BPSs can be reduced.

On the other hand, SDIS25 suggested testing 5 other positions that they had planned. For which only stage 2 was executed. From these 5, the best position was named NC-HU (new center based on a human decision). For this center, we tested 12 configurations, their results are also shown in Table 12.2.

Fig. 12.3 shows the Doubs territory and highlights the western area, where the optimization of the positioning for the new center was performed (stage 1). The existing centers are represented with brown dots, the best position obtained for NC-AU is denoted by a red dot, and the best position suggested by SDIS25 is represented by a blue dot (NC-HU). As we can see, the two positions are very close, which indicates that more operational load is present in that area and that it could be better distributed if we positioned more resources. From Table 12.2, we deduce the amount of resources for the new center. If we implement the NC-AU center, the best configuration, without consuming too much resources, would be 3VSAV, reaching a decrease of 11.84% BPSs. Furthermore, the nearest centers: *Besançon Centre*, *Besançon Est*, and *Saone Mamirolle* presented a reduction of 41%, 47%, and 5% in the number of emergencies attended, respectively, and a reduction of 86%, 74%, and 4% in the number of BPS, respectively. This was due to the redistribution of the operational load. If we wish to further reduce breakdowns but using more engines, a better configuration could be 4VSAV-1VTU-1FPT, reducing BPS in 14.33%. Regarding implementation of the NC-HU center, the best configuration would be CSR-2VSAV-1FPT, reducing the breakdowns by 6.14%. This configuration should be enough since the following configurations would use more engines but the reduction of breakdowns would be minimal, only 1 or 2 more BPS.

Table 12.2: Results of configurations simulated for the new center from the automated decision (NC-AU) and from human decision (NC-HU)

Center	Configuration	BPS decrease (%)	Nb BPS	Nb BPS SAP	Nb BPS INC
NC-AU	CPI = 1VSAV	-14.77	785	632	153
	CPIR = 2VSAV	6.43	640	488	152
	CPIR = 1VSAV + 1VTU	-14.91	786	633	153
	CS = 2VSAV + 1VTU	6.43	640	488	152
	CS = 1VSAV + 1FPT	-13.60	777	631	146
	CSR = 2VSAV + 1FPT	7.60	632	487	145
	CSR = 1VSAV + 1VTU + 1FPT	-13.45	776	631	145
	CSP = 2VSAV + 1VTU + 1FPT	7.46	633	488	145
	<b>X = 3VSAV</b>	<b>11.84</b>	<b>603</b>	<b>451</b>	<b>152</b>
	X = 3VSAV + 1VTU + 1FPT	13.01	595	450	145
	X = 4VSAV + 1VTU + 1FPT	14.33	586	441	145
	X = 4VSAV + 1VTU + 2FPT	14.47	585	441	144
	X = 4VSAV + 2VTU + 2FPT	14.47	585	441	144
	NC-HU	CPI = 1VSAV	3.22	662	505
CPIR = 2VSAV		5.26	648	491	157
CPIR = 1VSAV + 1VTU		3.22	662	505	157
CS = 2VSAV + 1VTU		5.26	648	491	157
CS = 1VSAV + 1FPT		4.09	656	505	151
<b>CSR = 2VSAV + 1FPT</b>		<b>6.14</b>	<b>642</b>	<b>491</b>	<b>151</b>
CSR = 1VSAV + 1VTU + 1FPT		4.24	655	505	150
CSP = 2VSAV + 1VTU + 1FPT		6.29	641	491	150
X = 3VSAV + 1VTU + 1FPT		6.29	641	490	151
X = 4VSAV + 1VTU + 1FPT		6.43	640	490	150
X = 4VSAV + 1VTU + 2FPT		6.43	640	490	150
X = 4VSAV + 2VTU + 2FPT		6.43	640	490	150
Baseline			684	527	157

## 12.4/ DISCUSSION AND RELATED WORK

In the literature, we generally find approaches aimed at dynamic redeployment of engines to improve coverage [66, 112], this also gives clues about which areas require more resources or the positioning of a new center. Furthermore, the positioning of a center and its resources is also a case of the facility location and production planning problem, since it is a multi-echelon multi-plant multi-item multi-customer multi-period production-distribution network design problem. Here, we can find works like [194], where they developed a supervised learning-driven heuristic that obtains better solutions than CPLEX. Also, there is [186], where they proposed three linear relaxation based heuristics and an evolutionary heuristic to solve the dynamic facility location problem. More directly related to the positioning of a center, we found [55], published in 2012, where a consortium of universities and a private firm built an operational tool to define fire station locations, equipment assignments, response times, etc., to build a national reorganization plan for Belgium.

In our case, we used the previously developed simulator and quality indicator to measure the impact of the new center with respect to the number of breakdowns per center and globally. The positioning and selection of the new center configuration can be built as a module within a system, where the simulator would be a shared engine with other optimization modules. Comparing with existing literature, we noted that our approach not only seeks to minimize one or several objectives but also to give more details on how

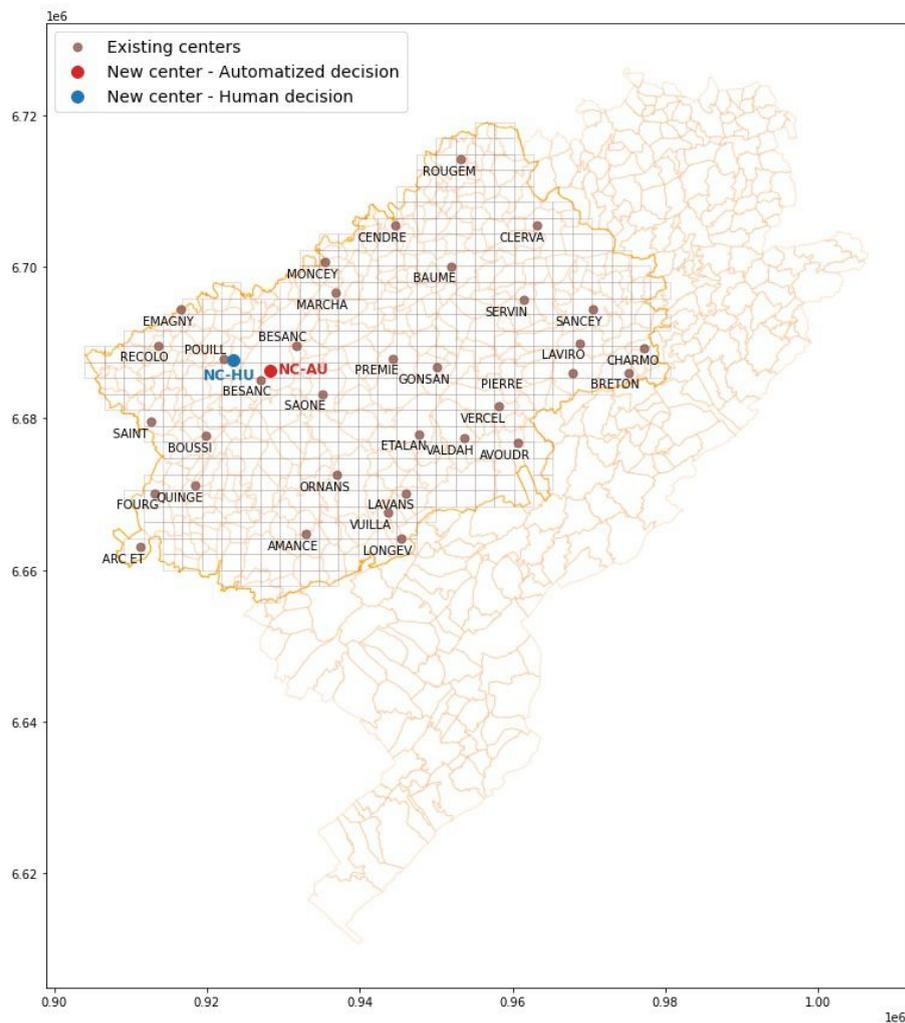


Figure 12.3: Map of Doubs, highlighted in the west zone. The automated search located the new center in red, while the SDIS25 proposed the location in blue.

to perform the new implementation and by means of statistics and reports visualize its effects, similar to the objectives of the tool developed in Belgium, but with the difference that we apply machine learning. In this way, with the support of a data-driven decision tool, SDIS25 and other Fire Departments would improve the quality of their services.

## 12.5/ CONCLUSION

In this chapter, we optimized the positioning and configuration of a new center. For this we created a 2-stage methodology. In the first stage, we performed a search to establish the most optimal position of the new center and built a new deployment plan, we distributed the operational load among all the centers, and finally, we validated how much operational load was absorbed by the new center. In the second stage, we tested different engine

configurations for the new center, considering that the number of agents per day depends on the number of engines and their availability is every day for 24h.

The results showed that the new center (NC-AU) selected with the proposed methodology reduced by 11.84% the number of BPSs, consuming less resources such as 3VSAV. What is more, the two nearest centers to NC-AU reduced their breakdowns by more than 70% and its impact was better than the center proposed by SDIS25 (NC-HU), since the decrease was only 6.14% of BPSs. Although the results were satisfactory and allow SDIS25 to make a better decision to implant a new center, the decision is also related to geographic availability and the risks involved, which can be added in a future work. Finally, this methodology in real life would work well since it would give the flexibility to SDIS25 to simulate the positioning of a new center through a 2-stage automated search or to simulate one or several suggestions by SDIS25 using only stage 2.



## OPTIMAL SIZING OF AGENTS

In the previous Chapter, it was created a methodology for positioning a new center and defined its type according to the number of adapted and first aid engines. In the present Chapter, we will complete the organization of the new center by developing methodologies for sizing agents, considering quantity, training, and availability.

### 13.1/ INTRODUCTION

From the previous chapter, we know the position of the new center that would reduce the maximum number of breakdowns. We also determined its number of engines considering an ideal availability of agents. With these results in mind, in this chapter, we will seek to determine the real sizing of the agents by answering the question (Q1): *what would be the optimal sizing of the personnel and their training to make the engines available for a given period?* For this, the following restrictions should be considered:

- Depending on the type of interventions that a new center is going to cover (emergency assistance to people, traffic accidents, fires, industrial risks and pollution, forest fires, wildlife protection, etc.), one or more types of engines will be used.
- There is an intrinsic relationship between engines and agents, i.e., depending on the type of engine a specific number of agents will be needed. In addition, agents must have the necessary skills to manipulate engines, that is, not just any agent can be part of the engine's crew. However, allocating more agents than necessary for a new center would result in a high cost to the fire department.
- During training, an agent can acquire 1 or more skills to manipulate 1 or more engines. Having more skills means having more training, which results in a high cost to the fire department. However, if the agents do not have the necessary training, the engines could not depart to attend interventions and a region would not be protected.

- There is a maximum total number of hours per agent in a month. And per day, an agent can work periods of 8, 12 or 24 hours (h) and immediately rest 16h, 12h or 24h, respectively. For example, if an agent on day 1 worked 24h, on day 2 he will not work, and on day 3 he will be able to work 8h, 12h or 24h.

Table 13.1: Example of the definition of resources for a new center

Type of intervention	Engine	Skill	Number of Agents
SAP	E1	S1	1
		S2	1
		S3	1
INC	E2	S4	1
		S5	1
		S6	2
		S7	2
<b>Total number of agents per day</b>			<b>9</b>

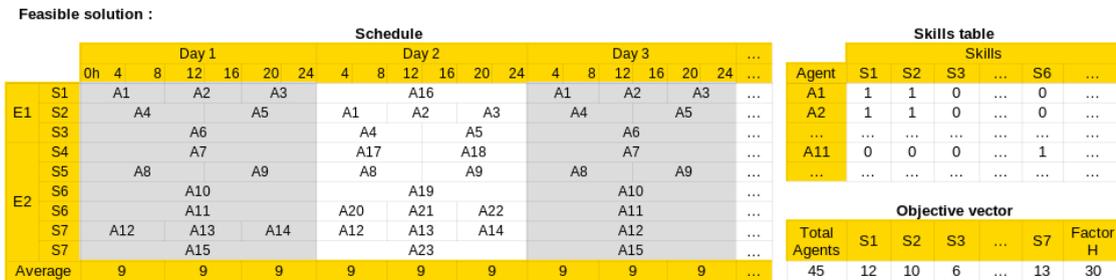


Figure 13.1: Illustration of a feasible solution for agent sizing.

In chapter 10, we explained that an engine will be able to depart if its functions are covered by a set of firefighters with the necessary skills, i.e., a function can be completed by different skills. However, to reduce the number of combinations, we will set that 1 function will be represented by 1 skill. For example, Table 13.1 shows the number of agents per required skill according to 2 types of engines and interventions. We can make the question (Q2): if a new center has 1 engine for emergency assistance and 1 for fires and the maximum number of worked hours per agent in a month is 134, how many total agents and agents per skill will be needed during 4 months? From Table 13.1, it is known that 9 agents on average by period will have to be available in one day. Then, a possible algorithm to build a solution is to choose a number of agents and agents per skill and try to organize them in schedules. The arrangement could be in a random or greedy form, and the combination of agents and skills could be represented in a single objective function to minimize in each iteration. However, with the combination of skills by agent, periods, days, and increment of engines (if we applied it for different kinds of centers), this approach would be intractable. Besides, to build a single objective function, we would have to know the order of priority of each objective, which is not possible because it could change in the long term in the fire department administration.

Therefore, we propose to start by building a set of random feasible solutions, i.e., as we

build a solution we will add agents and skills. These solutions will not necessarily be of good quality but will respect all constraints. An example of a feasible solution for Q2 is depicted in Fig. 13.1, where a number of agents (A1, A2, A3, etc.) are organized by day in periods of 8, 12, and 24h and has 9 agents on average. Then, as there are three main objectives to minimize: number of total agents, number of agents per skill, and hours not worked, we will search optimal solutions using multi-objective optimization. For this, we developed the NSGA-II Adapted for Firemen Optimization (NSGA-II-AFO) algorithm, which is an adaptation of NSGA-II. In this way, there will be a set of best solutions as a result, from which the fire brigade will be able to select one or several according to the priority they give to the objectives (for example, if the training for the skill S1 is expensive, a solution that requires fewer agents with that skill could be chosen).

Furthermore, the fire department needs to have a broad view of the source of the firefighters, i.e., to analyze whether it is more convenient to have all new agents or whether it is possible to redeploy existing agents from other centers to organize the new center maintaining the reduction of breakdowns. For this reason, question Q1 will be answered for 3 cases: (1) when all agents are new, (2) when some are new and some are existing, and (3) when all are existing agents.

In this way, the SDIS25 will have the capable, necessary and available personnel to complement the organization of the new center, reduce costs, improve the quality of its service, and protect the population.

## 13.2/ OPTIMAL SIZING FOR NEW AGENTS

Case (1) considers that all agents are new, i.e., the new center will not take agents coming from other centers, so there will be no need to launch the simulator to determine their impact. In addition, we will need a multi-objective optimization algorithm to identify the optimal sizing of the new agents. The objectives to be minimized are (a) the total number of agents, since fewer agents means lower hiring or training costs for the fire department; (b) the total number of agents per skill, this would also reduce training costs, prioritize skills, and identify the optimal sizing per skill; and (c) the total number of hours not worked, since the optimal distribution of agents availability reduces the number of agents. In real life there are 2 types of firefighters: professional and volunteer, both types receive the same training but their availability is different. To simplify the problem, we will not make a distinction in the type of firefighter, i.e., if our results, for example, would show the need for 1 firefighter with skill S1 and working 24h in a row, in real life this could be solved with 1 professional firefighter or with 2 volunteer firefighters working 12h each.

In Fig. 13.2, we present the schema of the NSGA-II-AFO algorithm to solve this case. The

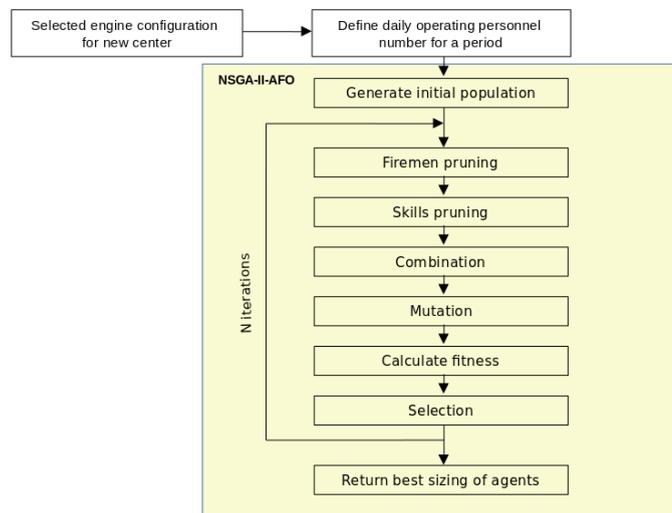


Figure 13.2: Methodology for optimizing the personnel sizing in the new center, considering only new agents.

pseudocode of the main function is in Algorithm 2. Given the number of engines for the new center, we define the daily number of agents for a given period. Then, we construct a set of initial feasible solutions with Algorithm 3, similar to Fig. 13.1. We intuitively calculate the approximate number of firefighters for a month by multiplying the number of daily firefighters  $e_{jo}$  by a  $increment\_factor = 5$ . The resulting value will help us to initially create a group of firefighters and more will be added if they are not enough. Since the initial solutions are not of good quality, we will perform an iterative process to optimize these solutions. Initially, we will reduce the number of agents (pruning firemen with Algorithm 5) and eliminate unused skills (pruning skills with Algorithm 6). Next, we will combine the firefighters' skills (combination with Algorithm 7), i.e., we will randomly select 2 agents to add and remove their skills. We will mutate the agents' skills (mutation with Algorithm 8), i.e., we will randomly add or remove an skill from a random agent. We will calculate the fitness of each chromosome by counting the number of agents, the number of agents per skill, and computing  $Factor\_H$ .  $Factor\_H$  is the dot product between an index vector and an agent vector. For example: we have 4 agents with [4, 8, 12, 32] free hours; knowing that the maximum number of free hours per agent is 40, we create an index vector with an increment of 8 hours [0:0, 1:8, 2:16, 3:24, 4:32, 5:40] that represents the pair index:max-free\_hours; we create the agent vector based on the number of agents per free hours and according to the index vector, i.e., if the first agent had 4 free hours, he would correspond to the pair 0:0, then the vector for the 4 agents would be [1, 2, 0, 0, 0, 1, 0]; finally, we apply the dot product between the index vector [0, 1, 2, 3, 4, 5] and the agent vector [1, 2, 0, 0, 0, 1, 0] and we obtain 6, which is the value of  $Factor\_H$  for this example. This allows us to reduce the total number of hours not worked, giving more weight (index) to the number of agents who have more free hours, the closer to zero, the better. The fitness

is represented by the objective vector in Fig. 13.1. Then, we will select the best offspring (selection process according to [25]) that we will optimize in a following iteration. Finally, after  $N$  iterations, we will return the best  $nbsol$  solutions.

Variables:

$skls$ : set of skills.

$ejo$ : set of number of agents needed per skill in a day (daily personnel number), according to the number of engines.

$per$ : set of periods worked with their respective time off (example: 8h worked, time off of 16h; 12h worked, time off of 12h; and 24h worked, time off of 24h).

$np$ : set of new chromosomes (new population).

$schedule_i$ : set of organized agents per day, skill, and worked period, corresponding to solution  $i$ .

$fm_i$ : set of agents with their respective skills and free hours, corresponding to solution  $i$ .

$solution$ : a chromosome/solution.

$ag$ : an agent.

$sk$ : a skill.

$p$ : a period.

$d$ : a day.

$s^*$ : set of best solutions.

Parameters:

$N$ : number of generations (iterations) in NSGA-II-AFO.

$increment\_factor$ : factor for increasing the number of firefighters in a month.

$max\_wh$ : maximum number of hours that an agent can work.

$nbip$ : size of initial population.

$rfp$ : rate of firemen pruning.

$rsp$ : rate of skills pruning.

$rcb$ : combination rate.

$rmt$ : mutation rate.

$rpareto$ : pareto rate.

$nbsol$ : number of best final solutions.

$nbdays$ : number of days of the simulated period.

$nbappear$ : number of times that a skill appear in  $schedule_i$ .

$cb\_nbtrials\_sol$ : number of attempts to modify a solution, during the combination phase.

$cb\_nbtrials\_ag$ : number of attempts to combine agents in a solution.

**Algorithm 2** NSGA-II-AFO

---

**Input :** *nbip, rfp, rsp, rpp, rcb, rmt, rpareto, nbsol, ejo, per*

- 1: *np*  $\leftarrow$  `create_initial_population(nbip, ejo, per, nbdays, max_wh)`
- 2: **for** *i* in  $\{1, \dots, N\}$  **do**
- 3:   *offspring*  $\leftarrow$   $\{\}$
- 4:   *offspring.add*(`firemen_pruning(np, rfp, nb_fm_lfh, per, max_wh)`)
- 5:   *offspring.add*(`skills_pruning(np, rsp, nbappear, max_wh)`)
- 6:   *offspring.add*(`combination(np, rcb, max_wh, skls, per, cb_nbtrials_sol, cb_nbtrials_ag)`)
- 7:   *offspring.add*(`mutation(np, rmt, max_wh, skls, per)`)
- 8:   *fitness\_values*  $\leftarrow$  `calculate_fitness(offspring)` ▷ calculate objective vector
- 9:   *np*  $\leftarrow$  `selection(fitness_values, rpareto)`
- 10: **end for**
- 11: *s\**  $\leftarrow$  `selection(np, nbsol)` ▷ select best final solutions

**return :** *s\**

---

**Algorithm 3** NSGA-II-AFO: `create_initial_population`


---

**Input :** *nbip, ejo, per, nbdays, max\_wh*

- 1: *solutions*  $\leftarrow$   $\{\}$
- 2: **for** *i* in  $\{1, \dots, nbip\}$  **do**
- 3:   *fm<sub>i</sub>*  $\leftarrow$  `create_increment_factor * ejo` initial agents with *max\_wh* available hours
- 4:   *schedule<sub>i</sub>*  $\leftarrow$   $\{\}$
- 5:   **for** *d* in  $\{1, \dots, nbdays\}$  **do**
- 6:     **if** *d*  $\neq$  1 **then**
- 7:       *rand\_per*  $\leftarrow$  choose random periods from *per* for each agent needed in *ejo*
- 8:       *avail\_ag*  $\leftarrow$  choose available agents from *fm* to complete *rand\_per*
- 9:       **if** *avail\_ag* is not enough **then**
- 10:          *new\_ag*  $\leftarrow$  create new agents
- 11:          *avail\_ag.add*(*new\_ag*)
- 12:       **end if**
- 13:     **else**
- 14:       *rand\_per*  $\leftarrow$  choose random periods from *per* for each agent needed in *ejo*
- 15:       *avail\_ag*  $\leftarrow$  choose available agents from *fm* to complete *rand\_per*
- 16:     **end if**
- 17:     update *fm<sub>i</sub>* with *avail\_ag*
- 18:     update *schedule<sub>id</sub>* with *avail\_ag*
- 19:   **end for**
- 20:   *solution<sub>i</sub>*  $\leftarrow$  `save(fmi, schedulei)`
- 21:   *solutions.add*(*solution<sub>i</sub>*)
- 22: **end for**

**return :** *solutions*

---

**Algorithm 4** NSGA-II-AFO: `replace_fireman`


---

**Input :** *fm, schedule, worked\_days, ag\_1*

```

1: changed_days ← {}
2: for d in worked_days do                                     ▷ replace agents that worked few hours
3:   fm_lfh ← sort fm from less free hours to more free hours, excluding ag_1
4:   for ag_2 in fm_lfh do
5:     if ag_2 can replace ag_1 then
6:       update fm and schedule replacing ag_1 by ag_2 for day d
7:       changed_days.add(d)
8:       break
9:     end if
10:  end for
11: end for
12: unchanged_days ← worked_days-changed_days                               ▷ change period with different agents
13: for d in unchanged_days do
14:   per_ag ← identify period worked by ag_1
15:   sk_ag ← identify occupied skill by ag_1
16:   unused_per ← per - per_ag
17:   ags_1 ← extract agents that worked per_ag hours in day d, occupying skill sk_ag
18:   for p in unused_per do
19:     ags_2 ← select available agents from fm to work periods of p hours for day d
20:     if ags_2 can replace ags_1 then
21:       update fm and schedule replacing ags_1 by ags_2 for day d
22:       break
23:     end if
24:   end for
25: end for
return : fm, schedule

```

---

**Algorithm 5** NSGA-II-AFO: `firemen_pruning`


---

**Input :** *np, rfp, nb\_fm\_lfh, per, max\_wh*

```

1: offspring ← {}
2: selected_solutions ← select rfp random solutions from np
3: for solution in selected_solutions do
4:   fm ← extract all agents from solution
5:   schedule ← extract schedule from solution
6:   fm_mfh ← sort fm from more free hours to less free hours and take the nb_fm_mfh first agents
7:   for ag_1 in fm_mfh do
8:     worked_days ← identify the worked days of ag_1
9:     fm, schedule ← replace_fireman(fm, schedule, worked_days, ag_1)
10:    if worked hours of ags_1 is max_wh then                               ▷ delete agent ags_1
11:      delete ags_1 from fm
12:    end if
13:  end for
14:  fm ← delete agents from fm that have max_wh free hours                       ▷ delete agents
15:  fm ← delete unused skills by agent from fm                                   ▷ delete skills
16:  update solution with fm and schedule
17:  offspring.add(solution)
18: end for
return : offspring

```

---

---

**Algorithm 6** NSGA-II-AFO: skills\_pruning
 

---

```

Input :  $np, rsp, nbappear, max\_wh$ 
1:  $offspring \leftarrow \{\}$ 
2:  $selected\_solutions \leftarrow$  select  $rsp$  random solutions from  $np$ 
3: for  $solution$  in  $selected\_solutions$  do
4:    $fm \leftarrow$  extract all agents from  $solution$ 
5:    $schedule \leftarrow$  extract schedule from  $solution$ 
6:   for  $nap$  in  $1, \dots, nbappear$  do ▷ number of times that an agent appears as  $sk$ 
7:      $appearances \leftarrow \{\}$ 
8:     for  $ag\_1$  in  $fm$  do
9:       if  $ag\_1$  has any  $sk$  that appears  $nap$  times in  $schedule$  then
10:         $appearances.add((ag\_1, sk, d, p))$  ▷  $d$  is the day of appearance and  $p$  is the worked period in  $d$ 
11:       end if
12:     end for
13:     for  $(ag\_1, sk, d, p)$  in  $appearances$  do
14:        $fm\_lfh \leftarrow$  sort  $fm$  from less free hours to more free hours, excluding  $ag\_1$ 
15:       for  $ag\_2$  in  $fm\_lfh$  do ▷ replace agent  $ag\_1$  by other agent
16:         if  $ag\_2$  can replace  $ag\_1$  in  $sk, d$ , and  $p$  then
17:           update  $fm$  and  $schedule$  replacing  $ag\_1$  by  $ag\_2$ 
18:           break
19:         end if
20:       end for
21:     end for
22:    $fm \leftarrow$  delete agents from  $fm$  that have  $max\_wh$  free hours ▷ delete agents
23:    $fm \leftarrow$  delete unused skills by agent from  $fm$  ▷ delete skills
24:   update  $solution$  with  $fm$  and  $schedule$ 
25:    $offspring.add(solution)$ 
26: end for
return :  $offspring$ 

```

---

**Algorithm 7** NSGA-II-AFO: combination

---

**Input :** *np, rcb, max\_wh, skls, per, cb\_nbtrials\_sol, cb\_nbtrials\_ag*

```

1: offspring ← {}
2: selected_solutions ← select rcb random solutions from np
3: for solution in selected_solutions do
4:   flag_sol_add = False                                ▷ Boolean to represent the addition of a skill
5:   flag_sol_del = False                                ▷ Boolean to represent the removal of a skill
6:   while (flag_sol_add==False or flag_sol_del==False) and cb_nbtrials_sol < 0 do
7:     fm ← extract all agents from solution
8:     schedule ← extract schedule from solution
9:     while cb_nbtrials_ag > 0 do
10:      ag_1, ag_2 ← choose randomly 2 agents from fm with free hours
11:      sk_added ← choose randomly a skill from skls that ag_1 does not has
12:      sk_deleted ← choose randomly a skill from ag_2
13:      if sk_added != sk_deleted then
14:        break
15:      else
16:        cb_nbtrials_ag -= 1
17:      end if
18:    end while
19:    worked_days ← extract days from schedule, where ag_2 worked as sk_deleted
20:    fm, schedule ← replace_fireman(fm, schedule, worked_days, ag_1)
21:    if all worked_days were modified in schedule then                                ▷ remove skill sk_deleted
22:      flag_sol_del = True
23:      fm ← remove sk_deleted from ag_2 and update fm
24:    end if
25:    add sk_added to ag_1                                                                ▷ add skill sk_added
26:    not_worked_days ← extract days where ag_1 did not work                            ▷ replace agents by ag_1
27:    for d in not_worked_days do
28:      ags ← extract all agents that worked in d with skill sk_added
29:      if an agent ag_4 from ags is replaced by ag_1 then
30:        update schedule replacing ag_4 by ag_1
31:        flag_sol_add = True
32:        break
33:      end if
34:    end for
35:  end while
36:  if flag_sol_add==True and flag_sol_del==True then                                ▷ if there was addition and removal, save the new solution
37:    fm ← delete agents from fm that have max_wh free hours                            ▷ delete agents
38:    fm ← delete unused skills by agent from fm                                        ▷ delete skills
39:    update solution with fm and schedule
40:    offspring.add(solution)
41:  end if
42: end for
return : offspring

```

---

**Algorithm 8** NSGA-II-AFO: mutation

---

```

Input :  $np, rmt, max\_wh, skls, per$ 
1:  $offspring \leftarrow \{\}$ 
2:  $selected\_solutions \leftarrow$  select  $rmt$  random solutions from  $np$ 
3: for  $solution$  in  $selected\_solutions$  do
4:    $fm \leftarrow$  extract all agents from  $solution$ 
5:    $schedule \leftarrow$  extract schedule from  $solution$ 
6:    $sk \leftarrow$  choose a random skill from  $skls$ 
7:    $delete\_sk \leftarrow$  choose a random value between True and False
8:    $sol\_changed = False$  ▷ Boolean to represent if the solution changed
9:   if  $delete\_sk == True$  then ▷ delete skill  $sk$ 
10:     $ag\_1 \leftarrow$  choose randomly an agent from  $fm$  with skill  $sk$ 
11:     $worked\_days \leftarrow$  extract days from  $schedule$  where  $ag\_1$  appears occupying  $sk$ 
12:     $fm, schedule \leftarrow$   $replace\_fireman(fm, schedule, worked\_days, ag\_1)$ 
13:    if if all  $worked\_days$  were modified then ▷ remove skill  $sk$ 
14:       $sol\_changed = True$ 
15:       $fm \leftarrow$  remove  $sk$  from  $ag\_1$  and update  $fm$ 
16:    end if
17:  else ▷ add skill  $sk$ 
18:     $ag\_1 \leftarrow$  choose randomly an agent from  $fm$  without skill  $sk$ 
19:    add  $sk$  to  $ag\_1$ 
20:     $not\_worked\_days \leftarrow$  extract days where  $ag\_1$  did not work ▷ replace agents by  $ag\_1$ 
21:    for  $d$  in  $not\_worked\_days$  do
22:       $ags \leftarrow$  extract all agents that worked in  $d$  with skill  $sk\_added$ 
23:      if an agent  $ag\_2$  from  $ags$  is replaced by  $ag\_1$  then
24:        update  $schedule$  replacing  $ag\_2$  by  $ag\_1$ 
25:         $sol\_changed = True$ 
26:        break
27:      end if
28:    end for
29:  end if
30:  if  $sol\_changed == True$  then ▷ if there was a change, save the new solution
31:     $fm \leftarrow$  delete agents from  $fm$  that have  $max\_wh$  free hours ▷ delete agents
32:     $fm \leftarrow$  delete unused skills by agent from  $fm$  ▷ delete skills
33:    update  $solution$  with  $fm$  and  $schedule$ 
34:     $offspring.add(solution)$ 
35:  end if
36: end for
return :  $offspring$ 

```

---

### 13.3/ OPTIMAL SIZING FOR NEW AND EXISTING AGENTS

The second case considers new and existing agents for the organization of the new center, i.e., a part of the agents will come from existing centers so we will need to measure the impact of their redistribution in the whole territory with the simulator. Besides, this will allow us to identify if it is possible to redistribute some agents and reach a reduction of total breakdowns.

Fig. 13.3 shows the methodology developed for the optimization of case (2). Initially, we define the number of daily agents. Since we need to explore various combinations of agents, we will parallelize the process into  $T$  tasks with different seeds. The parallelization will be performed with the Ray library [213]. In addition, we will establish a list of centers from which we will take the agents. For each seed, we will generate a random set of existing agents from the listed centers and launch a Bayesian optimization with HyperOpt. In each iteration of the Bayesian optimization, it will be selected a subset of agents.

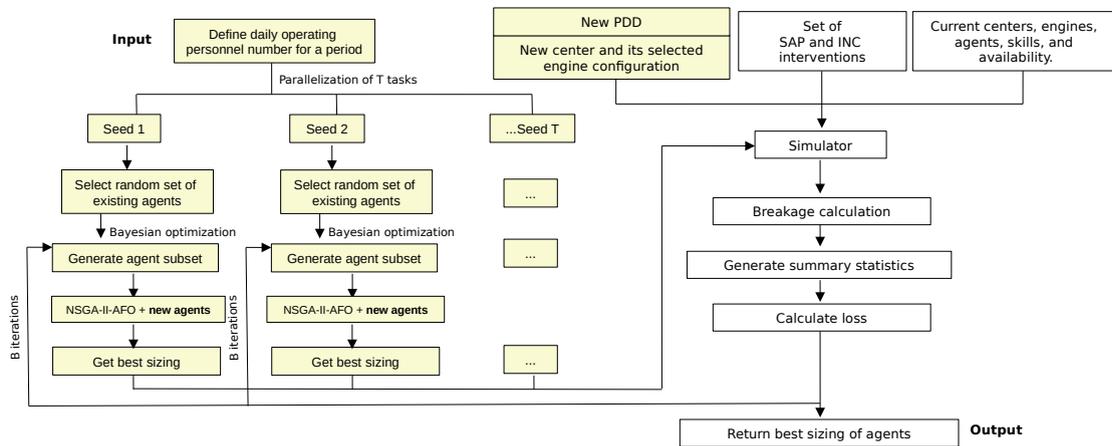


Figure 13.3: Methodology for optimizing the personnel sizing in the new center, considering new and existing agents.

Then, we will apply the NSGA-II-AFO algorithm to minimize the objectives (a), (b), and (c) mentioned in 13.2. we will create several initial solutions, adding new agents if necessary. For this case, we will adapt the NSGA-II-AFO algorithm to not produce changes in the existing agents and force them to be used in the solution. The initial solutions will be based on existing agents and new ones will be added as needed. In 5, 6, 7, and 8, we will add a filter to replace and eliminate new agents, since we want to use as few new firefighters as possible. In addition, only the skills of the new agents will be modified and eliminated, since we want to train only new agents and maintain the training of existing agents. Thus, we would reduce costs in the hiring and training of the agents. After  $N$  iterations for each task, we will retrieve the best solution from each iteration and send it to the simulator. The loss, defined in 13.2a, will be calculated based on the total number of BPS throughout the territory and the number of existing agents and new agents for the new center. This function is the one that Bayesian will try to minimize when searching for the best agent configuration.

Sets:

$C$ : set of centers.

$I$ : set of interventions SAP and INC.

$ET$ : set of engine types

Variables:

$I$ : set of interventions.

$Nbps_c^s$ : number of BPSs in center  $c$  during simulation  $s$ .

$Nbps_c^r$ : number of BPSs in center  $c$  and in real environment  $r$ .

$Nags_{new}$ : number of new agents in new center.

$Nags\_exist$ : number of existing agents in new center.

$Npi^s$ : number of processed interventions during simulation.

$Npi^r$ : number of processed interventions in real environment.

$Nabe_i^s$ : number of adapted backup engines in intervention  $i$  during simulation.

$Naee_i^s$ : number of adapted engaged engines in intervention  $i$  during simulation.

$Nfe_i^s$ : number of first aid engines in intervention  $i$  during simulation.

$Noe_{it}^s$ : number of other engines by type  $t$  and in intervention  $i$  during simulation.

$Noe_{it}^r$ : number of other engines by type  $t$ , in intervention  $i$ , and in real environment.

Parameters:

$\alpha$ : weight parameter for total number of BPS  $Nbps$ .

$\beta$ : weight parameter for total number of new agents  $Nags\_new$  in the new center.

$\theta$ : weight parameter for total number of existing agents  $Nags\_exist$  in the new center.

$B$ : number of iterations in Bayesian optimization.

$$Loss = \begin{cases} \alpha * \sum_{c \in C} Nbps_c^s + \beta * Nags_{new} + \theta * Nags_{exist} & \text{if } \sum_{c \in C} Nbps_c^s \leq \sum_{c \in C} Nbps_c^r \\ (\alpha + 0.2) * \sum_{c \in C} Nbps_c^s + (\beta - 0.1) * Nags_{new} + (\theta - 0.1) * Nags_{exist} & \text{if } \sum_{c \in C} Nbps_c^s > \sum_{c \in C} Nbps_c^r \end{cases} \quad (13.1)$$

$$\min \quad Loss \quad (13.2a)$$

$$\text{s.t.} \quad \alpha + \beta + \theta = 1, \quad (13.2b)$$

$$Npi^r = |I|, \quad (13.2c)$$

$$Npi^s = Npi^r, \quad (13.2d)$$

$$Nabe_i^s = 1 \quad \forall i \in I, \quad (13.2e)$$

$$Naee_i^s = 1 \quad \forall i \in I, \quad (13.2f)$$

$$Nfe_i^s \in \{0, 1\} \quad \forall i \in I, \quad (13.2g)$$

$$Noe_{it}^s = Noe_{it}^r \quad \forall i \in I, t \in ET. \quad (13.2h)$$

As can be seen in Fig. 13.3, the methodology use 2 levels of optimization: internally NSGA-II-AFO, which allows recovering the best sizing of agents, and externally Bayesian, which allows minimizing the increase of breakdowns while redistributing the agents. Moreover, it is observed that initially several sets of existing agents is created from a list of centers and then in each iteration Bayesian selects a subset, this may seem redundant but it is due to the fact that we search between minimum 1000 and maximum 3000

existing agents, i.e., in order to apply our methodology in real life and make it scalable, we need to explore by creating sets of agents and exploit by generating subsets.

### 13.4/ OPTIMAL SIZING FOR EXISTING AGENTS

The third case considers existing agents, i.e., we will take agents from other centers to organize the new center, therefore, it is necessary to measure the impact of this redistribution with the simulator. The optimization of this case will help us identify if there are enough agents to transfer to the new center, from which centers they should come, and how many breakdowns will be generated in the origin centers.

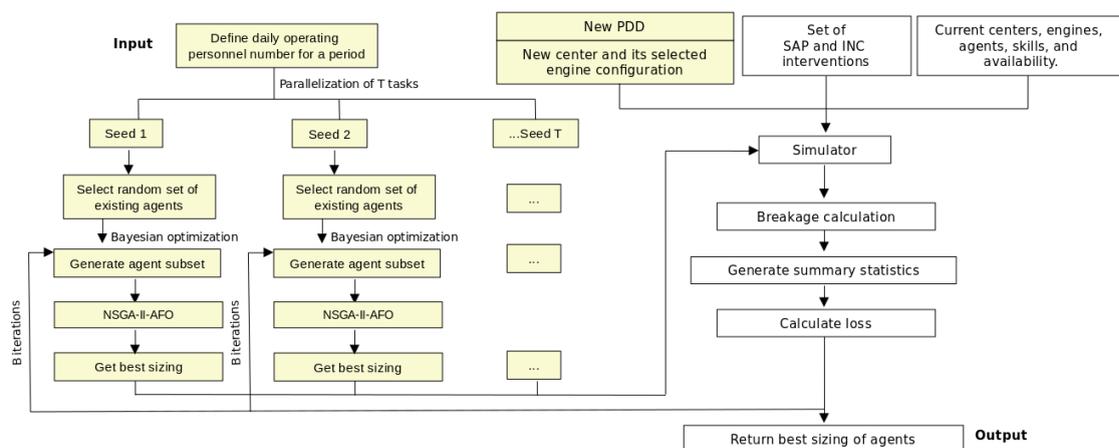


Figure 13.4: Methodology for optimizing the personnel sizing in the new center, considering only existing agents.

Fig. 13.4 illustrates the methodology developed for the optimization of case (3). The methodology is quite similar to the one developed in the previous section, but in this case we do not add new agents. In each iteration of the Bayesian optimization, a subset of existing agents will be selected. Then, we will apply the NSGA-II-AFO algorithm to minimize the objectives (a), (b), and (c) mentioned in 13.2. In this case, the NSGA-II-AFO algorithm had the following main changes: Algorithm 6 does not eliminate skills but based on the lowest number of occurrences of an agent's skill, he is replaced by other agents with few free hours; the combination algorithm (7) is not used; and in Algorithm 8, a random skill is not eliminated or added but a randomly chosen agent and its skill are replaced by other agents with few free hours. After  $N$  iterations, we retrieve the best solution and we send it to the simulator. After the simulation, the loss, defined in 13.4a, is estimated based on the total number of BPS throughout the territory and the number of agents used in the new center. This function is the one that Bayesian will try to minimize when searching for the best agent configuration.

Sets:

$C$ : set of centers.

$I$ : set of interventions SAP and INC.

$ET$ : set of engine types

Variables:

$I$ : set of interventions.

$Nbps_c^s$ : number of BPSs in center  $c$  during simulation  $s$ .

$Nbps_c^r$ : number of BPSs in center  $c$  and in real environment  $r$ .

$Nags\_new$ : number of new agents in new center.

$Nags\_exist$ : number of existing agents in new center.

$Npi^s$ : number of processed interventions during simulation.

$Npi^r$ : number of processed interventions in real environment.

$Nabe_i^s$ : number of adapted backup engines in intervention  $i$  during simulation.

$Nae_i^s$ : number of adapted engaged engines in intervention  $i$  during simulation.

$Nfe_i^s$ : number of first aid engines in intervention  $i$  during simulation.

$Noe_{it}^s$ : number of other engines by type  $t$  and in intervention  $i$  during simulation.

$Noe_{it}^r$ : number of other engines by type  $t$ , in intervention  $i$ , and in real environment.

Parameters:

$\alpha$ : weight parameter.

$B$ : number of iterations in Bayesian optimization.

$$Loss = \begin{cases} \alpha * \sum_{c \in C} Nbps_c^s + (1 - \alpha) * Nags & \text{if } \sum_{c \in C} Nbps_c^s \leq \sum_{c \in C} Nbps_c^r \\ (1 - \alpha) * \sum_{c \in C} Nbps_c^s + \alpha * Nags & \text{if } \sum_{c \in C} Nbps_c^s > \sum_{c \in C} Nbps_c^r \end{cases} \quad (13.3)$$

$$\min \quad Loss \quad (13.4a)$$

$$\text{s.t.} \quad Npi^r = |I|, \quad (13.4b)$$

$$Npi^s = Npi^r, \quad (13.4c)$$

$$Nabe_i^s = 1 \quad \forall i \in I, \quad (13.4d)$$

$$Nae_i^s = 1 \quad \forall i \in I, \quad (13.4e)$$

$$Nfe_i^s \in \{0, 1\} \quad \forall i \in I, \quad (13.4f)$$

$$Noe_{it}^s = Noe_{it}^r \quad \forall i \in I, t \in ET. \quad (13.4g)$$

## 13.5/ RESULTS

The experiments were done considering the best configuration found in the previous chapter 12 for the NC-HU center, since the location is available to implant the new center. The selected configuration was 2VSAV and 1FPT. According to this, the necessary skills would be VSCA, VSCOND, and VSEQUI for the VSAV, and FPCA, FPCE, FPCOND, and FPEQUI for the FPT. In this way, the objective vector to be minimized will be composed of the total number of agents needed, the number of agents for each skill previously mentioned, and the Factor\_H, which represents the remaining total free hours. In addition, we will use the baseline simulation with the values in Table 12.2 to compare the simulations of cases (2) and (3). The simulation period remains from 01/06/2020 00:00:00 to 01/10/2020 00:00:00.

### 13.5.1/ RESULTS FOR OPTIMAL SIZING OF NEW AGENTS

This subsection presents the results for case (1), where the optimization was performed with parameters:  $N = 400$ ,  $increment\_factor = 5$ ,  $max\_wh = 536h$ ,  $nbip = 100$ ,  $rfp = 0.4$ ,  $rsp = 0.4$ ,  $rcb = 0.5$ ,  $rmt = 0.5$ ,  $rpareto = 1$ ,  $nbsol = 10$ ,  $nbdays = 122$ ,  $nbappear = 6$ ,  $cb\_nbtrials\_sol = 100$ , and  $cb\_nbtrials\_ag = 100$ . Fig. 13.5 illustrates the performance of the NSGA-II-AFO algorithm. And, Table 13.2 shows the 3 best solutions obtained.

In Fig. 13.5, in red, we can see the reduction of the total number of agents as more iterations are done. In orange, we present the minimization of the number of agents for each skill versus the total number of agents. From here, we can see the non-dominated solutions in darker orange or the *Pareto Front*, which shows that our NSGA-II-AFO algorithm can converge to desired minima. In green, we observe the reduction of the total free hours (Factor\_H) as the iterations progress. All this allows us to identify the best sizing for a group of agents, reducing their free hours and distributing the necessary training.

The results of the 3 most efficient solutions in Table 13.2 suggest that it is possible to have a reduction of 6.14% of BPS (compared to the baseline in 12.2) if the new center has 66 agents working periods of 8h, 12h, and 24. It also recommend the best training for agents, which in average and by skill is 13-VSCA, 13-VSCOND, 14-VSEQUI, 8-FPCA, 15-FPCE, 8-FPCOND, and 17-FPEQUI.

Table 13.2: The 3 best solutions for sizing new agents.

Solution	Reduction BPS (%)	VSCA	VSCOND	VSEQUI	FPCA	FPCE	FPCOND	FPEQUI
1 <sup>o</sup>	6.14	13	14	15	8	18	8	20
2 <sup>o</sup>	6.14	15	14	13	7	14	8	15
3 <sup>o</sup>	6.14	11	11	14	8	14	8	15
<b>Average</b>	6.14	13	13	14	8	15	8	17
<b>Total number of agents: 66</b>								
<b>Worked periods: 8h, 12h, 24h</b>								

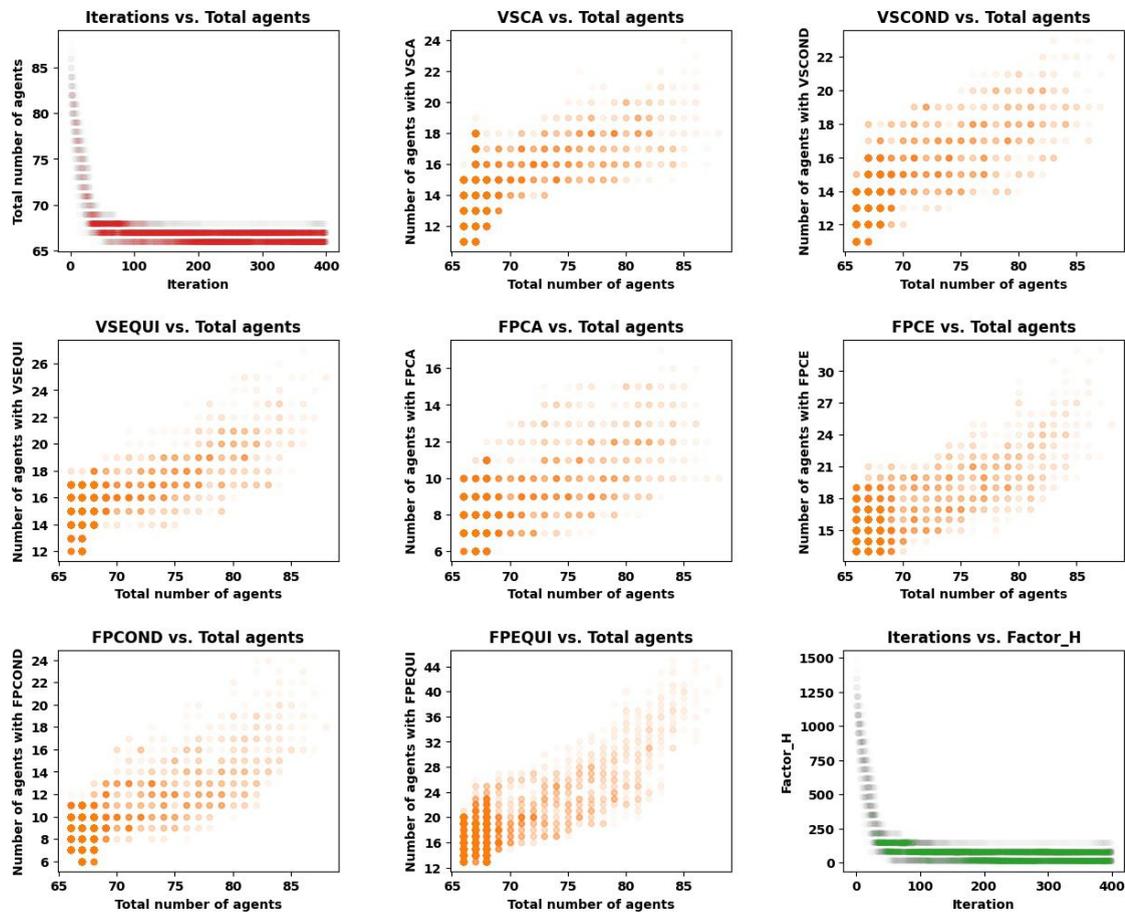


Figure 13.5: Optimization results of new agents sizing. The figure represents all the minimized targets. In the upper left, in red, and in the lower right, in green, it is presented the minimization of the total number of agents and the Factor\_H, respectively. In orange, the minimization of the number of agents per skill is illustrated.

### 13.5.2/ RESULTS FOR OPTIMAL SIZING OF NEW AND EXISTING AGENTS

This subsection presents the results for case (2), where the optimization was performed with parameters:  $T = 4$ ,  $B = 100$ ,  $N = 100$ . The other parameters for the NSGA-II-AFO algorithm are the same of the previous subsection. Fig. 13.6 presents the performance of the methodology developed in 13.3. Table 13.3 presents the 3 most efficient solutions.

From Fig. 13.6, in the upper left image, we can see the minimization of BPSs as the iterations increase. In the upper right image, we can see how the loss and the BPSs converge to a minimum. In the lower left image, we illustrate that as the total number of agents decreases, total free hours also decrease, i.e., lower cost of staff acquisition and higher productivity. And, in the lower right image, the *Pareto Front* appears as consequence of minimizing the number of new agents and existing agents, since we want to consume only the necessary resources from the other centers and acquire the fewest number of new agents.

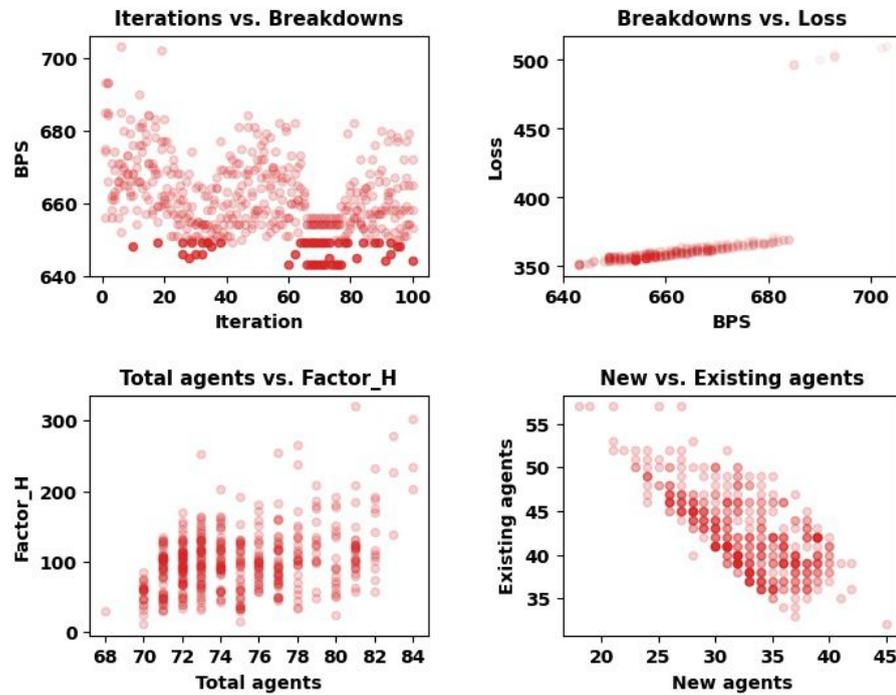


Figure 13.6: Optimization results of new and existing agents sizing. The top 2 images show the reduction of loss and breakdowns with Bayesian optimization. The lower 2 images illustrate the minimization of the number of agents and Factor\_H with the NSGA-II-AFO algorithm.

In Table 13.3, we see that the best solutions obtained a reduction between 5% and 6% of BPSs compared to the baseline in 12.2, i.e., agents from other centers did not generate a negative impact globally, on the contrary, the new center covered interventions that would be assigned to the centers close to him, reducing the operational overload. The number of new agents is less than the existing agents, which is satisfactory, since hiring a larger number of new agents and training them would increase costs and reduce immediate availability. Also, we observe that the number of agents per skill is greater than the numbers in Table 13.2. This is because the existing agents were considered with their actual skills, i.e., they did not increase or decrease training during the optimization with

Table 13.3: The 3 best solutions for sizing new and existing agents.

Solution	1 <sup>º</sup>	2 <sup>º</sup>	3 <sup>º</sup>
Reduction BPS (%)	6	5.70	5.12
Total agents	75	75	75
New agents	36	31	30
Existing agents	39	44	45
VSCA	25	30	28
VSCOND	39	43	44
VSEQUI	34	41	39
FPCA	17	17	20
FPCE	34	39	41
FPCOND	30	37	36
FPEQUI	38	43	44
Worked periods:	8h, 12h, 24h		

NSGA-II-AFO, so either it is necessary increase more agents or it is necessary to train more new agents.

### 13.5.3/ RESULTS FOR OPTIMAL SIZING OF EXISTING AGENTS

This subsection presents the results for case (3), where the optimization was performed with parameters:  $T = 4$ ,  $B = 100$ ,  $N = 100$ . The other parameters for the NSGA-II-AFO algorithm are the same of subsection 13.5.1. Fig. 13.7 presents the performance of the methodology developed in 13.4. Table 13.4 presents the 3 best solutions found.

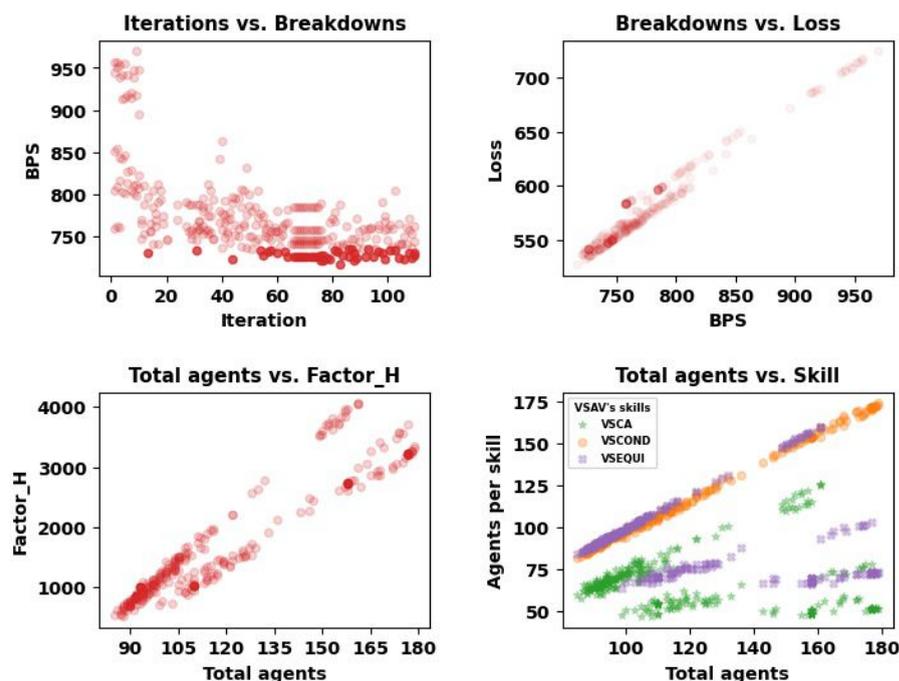


Figure 13.7: Optimization results of existing agents sizing. The top 2 images show the minimization of loss and breakdowns with Bayesian optimization. The lower 2 images illustrate the optimization of agents sizing with the NSGA-II-AFO Algorithm.

From Fig. 13.7, in the upper left image, we see how the BPSs are minimized as the iterations increase, i.e., convergence to a minimum is validated. In the upper right image, we observe how the BPSs and the loss are decreasing together in a linear way. In the lower left image, we see the decrease of agents as the free hours (Factor\_H) are reduced, i.e., less shared agents and more productivity. In the lower right image, we visualize three sets of skills: VSCA, VSCOND, and VSEQUI, which are the skills of the VSAV engine, the most requested in interventions. This image suggest a linear relationship between the minimization of total agents and the number of agents per skill. This remarkable linearity is predictable since we don't remove or add skills to existing agents (we kept the original training), i.e., fewer total agents, fewer agents per skill, so the main goal was to recover fewer agents but with the best training.

Table 13.4: The 3 best solutions for sizing existing agents.

Solution	1 <sup>o</sup>	2 <sup>o</sup>	3 <sup>o</sup>
Reduction BPS (%)	-4.82	-5.41	-5.56
Existing agents	85	107	107
VSCA	60	50	53
VSCOND	82	101	102
VSEQUI	84	68	70
FPCA	41	38	41
FPCE	79	63	66
FPCOND	74	100	102
FPEQUI	85	68	70
Worked periods:	8h, 12h, 24h		

As shown in Table 13.4, none of the best solutions resulted in a reduction of BPSs. On the contrary, there was an overall increase between 4% and 6%, i.e., from all the solutions explored, none of them recovered enough agents for the new center without deteriorating the overall quality of service. Also, we note that the number of total agents is significantly higher compared to Table 13.2 and Table 13.3, demonstrating a lack of training in the existing agents.

## 13.6/ DISCUSSION AND RELATED WORK

According to the literature, firefighter sizing is a case of the human resource allocation problem (HRAP), a variation of the assignment problem (AP) and a NP-hard combinatorial optimization problem. In AP, one seeks to assign a set of tasks to a set of individuals (e.g., assigning jobs to machines, jobs to workers, or workers to machines), where the objective of AP is to maximize the total profit of allocation or minimize the total cost, since human resources or machines have deficiencies or limitations represented in terms of costs to solve a task [39, 43, 92]. In [92], the mathematical formulation of HRAP for single-objective and multi-objective models and its corresponding variations are presented.

Human resources allocation is an essential factor for success in multi-task organizations, since it considers the quantity of professionals needed and their availability to perform a task. Among the related researches, we find [164, 34, 181], where approaches for optimizing HRAP are based on genetic algorithms and [64], where the authors developed a simulating annealing algorithm in 3 stages to reduce the number of stations via balancing the workload and maximizing the weighted efficiency. In [184], it is developed an intelligent model for workforce allocation, considering scenarios of multi-skilled operators. In [192], it is designed a complete proposal to support the modeling, implementation, and dynamic use of HRAP in industry 4.0. What is more, in [101], the authors proposed a methodology multi-criteria to optimize the allocation of human resources involved in an emergency service. The criteria were related to residence, seniority, family status, and also meritocratic aspects of the workers.

On the one hand, [101] defines an environment similar to ours, i.e., in both cases the ambulances/engines must be available 24h a day, there is a period of time to be simulated, and there are types of employees. On the other hand, there are differences, e.g., [101] considers 3 types of employment and in our case we could consider at least 3 types and at most 9 types (2 types of VTU, 3 types of VSAV, and 4 types of FPT), we do not use specific schedule changes such as [101], but possible periods with breaks that an agent can take (work 8h and rest 16h, work 12h and rest 12h, or work 24h and rest 24h). Furthermore, in [101], the authors aim to compute a final ranking for each station to distribute the employees, different from our case, which aims to optimize the number of agents and determine their optimal training. All these differences make our problem have a larger search space and be more complex. However, the advantage of our methodology is in all the information retrieved for the organization of the new center and the measurement of its impact when there is a synergy of all the centers in the territory.

### 13.7/ CONCLUSION

In this chapter, we have developed methodologies for optimizing the staffing of the new center. Thus, we were able to determine the total number of agents required and their corresponding training.

According to the engine configuration for the new center, we established 3 sizing cases: (1) when all agents are new, (2) when some are new and some are existing, and (3) when all are existing agents. For case (1) we developed the NSGA-II-AFO algorithm, where the number of agents, agents per skill, and hours not worked are reduced in  $N$  iterations and the best solution is retrieved. For case (2) and (3) we create 2 methodologies based on the NSGA-II-AFO algorithm, since we seek to find the best organization of the agents, and the simulator developed in 10, since we need to measure the impact of the redistribution of existing agents.

For case (1) and (2), the results were satisfactory because it was possible to maintain the reduction of BPSs. And even if in case (3), there was no reduction of breakdowns, we can see that the negative impact is small and perhaps by performing a broader search, we could further reduce this impact.

Finally, the results of this chapter allowed SDIS25 to analyze in more detail the complete implementation of a new center and its impact on the quality of service, considering its operational load and all its existing resources.

# V

## CONCLUSION & PERSPECTIVES



## CONCLUSION & PERSPECTIVES

### 14.1/ GENERAL CONCLUSION

In this manuscript, we developed several methodologies for predicting interventions and optimizing operational resources for fire departments, for short- and long-term, and based on ML and OR techniques. The document is mainly divided into 5 parts. The first part, Chapter 1, describes the scissors effect that has been taking place in fire departments, i.e., the constant increase in the operational load and the budget reduction to attend to it; the two main objectives; the ten contributions generated; and the outline of the thesis.

The second part, Chapter 2, presents definitions about the tasks and types of learning performed by ML. It describes techniques for data preprocessing and feature engineering, statistical methods for modeling, algorithms based on decision trees, and the neural networks used in the development of our studies. In addition, it briefly explains the OR process, definitions of integer programming, multi-objective programming, and Bayesian optimization, all of them also used in the construction of our methodologies.

The third part, Chapter 3, explains the internal data provided by the SDIS25 such as interventions performed, center's resources, victims, deployment plans, etc; data from external sources such as meteorological, traffic, epidemiological, hydric variables, etc; and the operational flow from the call received by the SDIS25 call center, through the arrival at the scene of the incident, the transport of the victim to a medical center, and the re-entry into availability of the engaged armament, as well as the description of the modality of the breakage calculation.

The fourth part, from Chapter 4 to Chapter 8, depicts the contributions related to the predictions of the interventions such as the prediction of the number of interventions in the next hour, the prediction of the turnaround time of ambulances in hospitals, the forecast of the response time of ambulances and the variables that influence it, the prediction of victim's mortality and their need for transport to health facilities, and the forecast of operational load peaks for short periods and due to rare events.

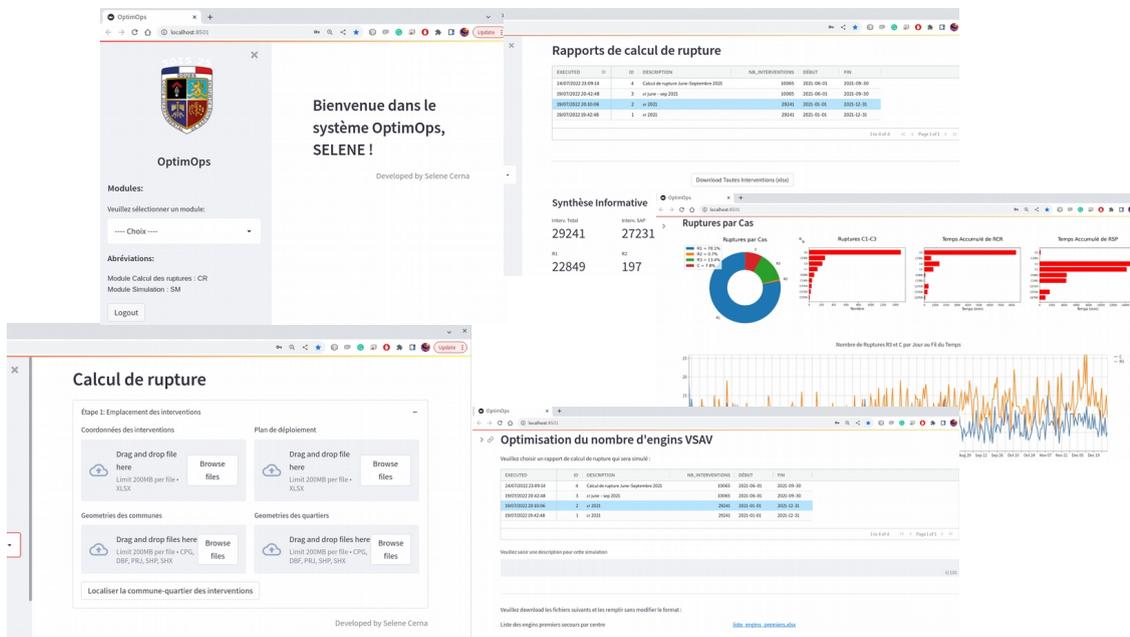


Figure 14.1: OptimOps System interface.

The fifth part, from Chapter 9 to Chapter 13, describes the contributions related to the optimization of resources such as the implementation of the optimized search for available armament as part of the breakage calculation (quality indicator) and the prediction of the breakdowns; the construction of the operational load simulator; the implementation of a new methodology for optimizing the redistribution of ambulances in the centers of the territory, based on the previously created simulator and indicator; the development of another methodology for optimizing the positioning of a new center and the definition of its resources; and another one for the optimization of the dimensioning of agents of the new center, considering new agents and those coming from other centers.

In general, the methodologies developed demonstrated good performances and it is feasible to apply them in real life as decision support systems. For example, preliminary studies on the prediction of the number of interventions in the next hours led to the creation of the PredictOps System, developed by *SAD Marketing* [208]. Also, the works developed on the optimization of resources such as the breakage calculation, operational load simulator, redistribution of ambulances, and the implementation of a new center, led to the creation of the OptimOps System, under construction by the present author, Fig. 14.1 show the interface of the system and some functionalities implemented with the open-source application framework for Machine Learning, named Streamlit.

## 14.2/ PERSPECTIVES

The independent application or hybridization of Machine Learning and Operations Research techniques in the field of civil security, as in the case of fire departments, is an up-to-date topic, attractive to government and private organizations, and of vital importance to society. Although we have developed several prediction and optimization works during this thesis, our field of study is comprehensive and promising. For this reason, in the following, we list some perspectives of the work developed:

- For Chapter 4, to experiment with different long-term time horizons, such as predicting the number of interventions for the next week or month. Thus, fire departments would have more time to reorganize. Also, add variables on social events from newspaper or magazine texts to improve the quality of predictions. Finally, build models for the prediction of the operational load by type of intervention and especially for infrequent types.
- For Chapter 5, to add more variables such as epidemic statistics and more keywords searched on Google with a previous analysis of the feature importance. There is also another direction for further research, and that is about finding the nearest hospital with a shorter turnaround time, considering internal hospital data.
- For Chapter 6, to add variables such as traffic conditions in the last hours, the number of dispatched ambulances, and the agents and engines available at each center in the hours prior to or at the time of the ambulance request, given that while there are few resources available, the response time may be longer. And to experiment with more ML techniques and feature engineering.
- For Chapter 7, to develop new approaches that take into account textual observations recorded by operators during calls, as well as voice records of victims.
- For Chapter 8, to add meteorological bulletins from other departments of the country, which would allow us to better track a possible extreme event and its consequences on the firefighters' workload. Also, to develop and compare other modeling and text preprocessing techniques with the texts in French and English. Finally, to integrate the results of this classification approach into a larger regression model that predicts the number of interventions for a certain period (hourly, daily, and monthly) and by locality.
- For Chapter 9, to experiment with adapting the optimized search so that it initially selects less skilled firefighters and leaves more skilled firefighters free for subsequent interventions. Given that in the manuscript we offer 2 different approaches for

the identification of breaks (i.e., ML-based models - Chapter 9, and hybrid methodologies based on ML and OR - Chapters 11, 12, and 13), if we were to continue down the path of the ML-based models, we would have to explore the creation of artificial data with techniques such as the Adaptive Synthetic Sampling Method (ADASYN) or some mechanism that allows us to increase our history of breakdowns and predict them for the following months and for each center, for example.

- For Chapter 10, to develop ML models to predict how long an engine of a specific type will be engaged in an intervention and how long an intervention will last. Add predictive variables in the model of arrival at the scene, such as the distances obtained by GoogleMaps, and in the depart model, such as the resources already requested and those available in the centers at the time of the request, all this considering that it is a simulation and processing times should be minimal.
- For Chapter 11, to implement the optimization of the redistribution of agents according to their training and in conjunction with the ambulances and other redeployed engines, considering their impact on the loss function. Experiment with parallel Bayesian optimization and other optimization methods.
- For Chapter 12 and Chapter 13, to consider the times and distances calculated by GoogleMaps or OSMR to calculate the distances between the commune-quartier and the centers when preparing the deployment plan. In addition, include geographic risk factors by area or community and the history of traffic on its roads.
- Among other tasks, we would propose the optimization of the daily operational staff table, which indicates how many firefighters on average should be available during the day and at night, here we could vary the periods according to the forecast demand. Also, the optimization of the resources when a center is suppressed, a new deployment plan should be elaborated according to the redistribution of the operational demand.

As we can see, there are still many problems to be solved for fire departments, EMS, and civil security, in general. These fascinating areas such as ML and OR and their hybridization open the doors for the proliferation of data-driven systems for decision making, not only in the field of civil security but also worldwide.

## PUBLICATIONS

During the period of this thesis, the author has published the following papers and resources. The superscript \* highlights equal contribution for co-first authors **in bold**.

## JOURNAL PAPERS

- **Cerna, S.**; Laymani, D.; Guyeux, C. (2022) The Usefulness of NLP Techniques for Predicting Peaks in Firefighter Interventions Due to Rare Events. **Neural Computing and Applications**, 34, 10117–10132 [191].
- \*Arcolezi, H. H., \***Cerna, S.**, Couchot, J.-F, Guyeux, C., & Makhoul, A. (2022) Privacy-Preserving Prediction of Victim’s Mortality and Their Need for Transportation to Health Facilities. **IEEE Transactions on Industrial Informatics**, 18(8), 5592-5599 [190].
- Arcolezi, H. H., **Cerna, S.**, Guyeux, C., & Couchot, J.-F. (2021). Preserving Geo-Indistinguishability of the Emergency Scene to Predict Ambulance Response Time. **Mathematical and Computational Applications**, 26(3), 56 [176]
- **Cerna, S.**, Arcolezi, H. H., Guyeux, C., Royer-Fey, G., & Chevallier, C. (2021). Machine learning-based forecasting of firemen ambulances’ turnaround time in hospitals, considering the COVID-19 impact. **Applied Soft Computing**, 109, 107561 [178].
- **Cerna, S.**; Guyeux, C.; Royer, G.; Chevallier, C.; Plumerel, G. (2020) Predicting Fire Brigades Operational Breakdowns: A Real Case Study. **Mathematics**, 8, 1383 [152]
- Arcolezi, H. H., Couchot, J.-F., **Cerna, S.**, Guyeux, C., Royer, G., Al Bouna, B., & Xiao, X. (2020). Forecasting the Number of Firefighters Interventions per Region with Local-Differential-Privacy-Based Data. **Computers & Security**, 96, 101888 [146].

## CONFERENCE PAPERS

- **Cerna, S.**, Guyeux, C., Arcolezi, H. H., Couturier, R., & Royer, G. (2020). A comparison of LSTM and XGBoost for predicting firemen interventions. *The 8th World Conference on Information Systems and Technologies (WorldCIST)*, April, 424–434 [150].
- **Cerna, S.**, Guyeux, C., Arcolezi, H. H., & Royer, G. (2020). Boosting Methods for Predicting Firemen Interventions. *The 11th International Conference on Information and Communication Systems (ICICS)*, 001–006 [151].

## OTHER PAPERS

Furthermore, the author also participated as a co-author in the following published papers.

- Cisneros, L. L., Arcolezi, H. H., **Cerna, S.**, Brandão, J.L., Santos, G.C., Navarro, T.P., & Carvalho, A.A. (2021). Machine Learning Algorithms to Predict In-Hospital Mortality in Patients with Diabetic Foot Ulceration. In *Proceedings of the XXIII Congresso da Sociedade Brasileira de Diabetes*.

# BIBLIOGRAPHY

- [1] PARETO, V. **Manual of Political Economy**. Augustus m Kelley Pubs, 1971.
- [2] TOREGAS, C., SWAIN, R., REVELLE, C., AND BERGMAN, L. **The location of emergency service facilities**. *Operations Research* 19, 6 (1971), 1363–1373.
- [3] ALY, A. A., AND WHITE, J. A. **Probabilistic formulation of the emergency service location problem**. *Journal of the Operational Research Society* 29, 12 (Dec. 1978), 1167–1179.
- [4] SCHILLING, D., ELZINGA, D. J., COHON, J., CHURCH, R., AND REVELLE, C. **The team/fleet models for simultaneous facility and equipment siting**. *Transportation Science* 13, 2 (May 1979), 163–175.
- [5] CAMERINI, P., GALBIATI, G., AND MAFFIOLI, F. **On the complexity of finding multi-constrained spanning trees**. *Discrete Applied Mathematics* 5, 1 (1983), 39–50.
- [6] SERAFINI, P. **Some considerations about computational complexity for multi objective combinatorial problems**. In *Recent Advances and Historical Development of Vector Optimization* (Berlin, Heidelberg, 1987), J. Jahn and W. Krabs, Eds., Springer Berlin Heidelberg, pp. 222–232.
- [7] IBARAKI, T. **Resource allocation problems**. Foundations of Computing. MIT Press, London, England, Jan. 1988.
- [8] BATA, R., DOLAN, J. M., AND KRISHNAMURTHY, N. N. **The maximal expected covering location problem: Revisited**. *Transportation Science* 23, 4 (Nov. 1989), 277–287.
- [9] CARSON, Y. M., AND BATA, R. **Locating an ambulance on the amherst campus of the state university of new york at buffalo**. *Interfaces* 20, 5 (October 1990), 43–49.
- [10] REPEDE, J. F., AND BERNARDO, J. J. **Developing and validating a decision support system for locating emergency medical vehicles in louisville, kentucky**. *European Journal of Operational Research* 75, 3 (1994), 567–581.
- [11] SRINIVAS, N., AND DEB, K. **Multiojective optimization using nondominated sorting in genetic algorithms**. *Evolutionary Computation* 2, 3 (1994), 221–248.

- [12] BREIMAN, L. **Bagging predictors**. *Machine Learning* 24, 2 (Aug. 1996), 123–140.
- [13] TIBSHIRANI, R. **Regression shrinkage and selection via the lasso**. *Journal of the Royal Statistical Society. Series B (Methodological)* 58, 1 (1996), 267–288.
- [14] BREIMAN, L. **Arcing the edge (technical report)**. *University of California, Berkeley, CA* (1997).
- [15] FREUND, Y., AND SCHAPIRE, R. E. **A decision-theoretic generalization of on-line learning and an application to boosting**. *Journal of Computer and System Sciences* 55, 1 (Aug. 1997), 119–139.
- [16] GENDREAU, M., LAPORTE, G., AND SEMET, F. **Solving an ambulance location model by tabu search**. *Location Science* 5, 2 (1997), 75–88.
- [17] HOCHREITER, S., AND SCHMIDHUBER, J. **Long short-term memory**. *Neural Computation* 9, 8 (nov 1997), 1735–1780.
- [18] CONE, D. C., DAVIDSON, S. J., AND NQUYEN, Q. **A time-motion study of the emergency medical services turnaround interval**. *Annals of Emergency Medicine* 31, 2 (Feb. 1998), 241–246.
- [19] OWEN, S. H., AND DASKIN, M. S. **Strategic facility location: A review**. *European Journal of Operational Research* 111, 3 (Dec. 1998), 423–447.
- [20] AGRAWAL, R., AND SRIKANT, R. **Privacy-preserving data mining**. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00* (2000), ACM Press.
- [21] CARTER, M. W., AND PRICE, C. C. **Operations research**. *Operations Research Series*. CRC Press, Boca Raton, FL, July 2000.
- [22] TASHMAN, L. J. **Out-of-sample tests of forecasting accuracy: an analysis and review**. *International Journal of Forecasting* 16, 4 (Oct. 2000), 437–450.
- [23] AGRAWAL, D., AND AGGARWAL, C. C. **On the design and quantification of privacy preserving data mining algorithms**. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '01* (2001), ACM Press.
- [24] BREIMAN, L. **Random forests**. *Machine Learning* 45, 1 (2001), 5–32.
- [25] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. **A fast and elitist multiobjective genetic algorithm: Nsga-ii**. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

- [26] PONS, P. T., AND MARKOVCHICK, V. J. **Eight minutes or less: does the ambulance response time guideline impact trauma patient outcome?** *The Journal of Emergency Medicine* 23, 1 (July 2002), 43–48.
- [27] SCHMEISER, B. **Some myths and common errors in simulation experiments.** In *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)* (Aug. 2002), IEEE.
- [28] SWEENEY, L. **k-anonymity: A model for protecting privacy.** *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (Oct. 2002), 557–570.
- [29] AUSTIN, P. C. **Quantile regression: A statistical tool for out-of-hospital research.** *Academic Emergency Medicine* 10, 7 (July 2003), 789–797.
- [30] ECKSTEIN, M., AND CHAN, L. S. **The effect of emergency department crowding on paramedic ambulance availability.** *Annals of Emergency Medicine* 43, 1 (Jan. 2004), 100–105.
- [31] KARASAKAL, O., AND KARASAKAL, E. K. **A maximal covering location model in the presence of partial coverage.** *Computers Operations Research* 31, 9 (aug 2004), 1515–1526.
- [32] PELEG, K., AND PLISKIN, J. S. **A geographic information system simulation model of EMS: reducing ambulance response time.** *The American Journal of Emergency Medicine* 22, 3 (May 2004), 164–170.
- [33] ECKSTEIN, M., ISAACS, S. M., SLOVIS, C. M., KAUFMAN, B. J., LOFLIN, J. R., O’CONNOR, R. E., AND PEPE, P. E. **Facilitating EMS turnaround intervals at hospitals in the face of receiving facility overcrowding.** *Prehospital Emergency Care* 9, 3 (Jan. 2005), 267–275.
- [34] OSMAN, M., ABO-SINNA, M., AND MOUSA, A. **An effective genetic algorithm approach to multiobjective resource allocation problems (moraps).** *Applied Mathematics and Computation* 163, 2 (2005), 755–768.
- [35] BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics).** Springer-Verlag, Berlin, Heidelberg, 2006.
- [36] DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. **Calibrating noise to sensitivity in private data analysis.** In *Theory of Cryptography*. Springer Berlin Heidelberg, 2006, pp. 265–284.
- [37] SEGAL, E., VERTER, V., COLACONE, A., AND AFILALO, M. **The in-hospital interval: A description of EMT time spent in the emergency department.** *Prehospital Emergency Care* 10, 3 (Jan. 2006), 378–382.

- [38] CHUANG, C.-L., AND LIN, R.-H. **A maximum expected covering model for an ambulance location problem.** *Journal of the Chinese Institute of Industrial Engineers* 24, 6 (2007), 468–474.
- [39] PENTICO, D. W. **Assignment problems: A golden anniversary survey.** *European Journal of Operational Research* 176, 2 (2007), 774–793.
- [40] SILVERMAN, R. A., GALEA, S., BLANEY, S., FREESE, J., PREZANT, D. J., PARK, R., PAHK, R., CARON, D., YOON, S., EPSTEIN, J., AND RICHMOND, N. J. **The “vertical response time”: Barriers to ambulance response in an urban area.** *Academic Emergency Medicine* 14, 9 (Sept. 2007), 772–778.
- [41] KASIVISWANATHAN, S. P., LEE, H. K., NISSIM, K., RASKHODNIKOVA, S., AND SMITH, A. **What can we learn privately?** In *2008 49th Annual IEEE Symposium on Foundations of Computer Science* (Oct. 2008), IEEE.
- [42] BIRD, S., KLEIN, E., AND LOPER, E. **Natural language processing with Python: analyzing text with the natural language toolkit.** ” O’Reilly Media, Inc.”, 2009.
- [43] BLUMENFELD, D. **Operations Research Calculations Handbook.** CRC Press, dec 2009.
- [44] HASTIE, T., TIBSHIRANI, R., AND FRIEDMAN, J. **The Elements of Statistical Learning.** Springer New York, 2009.
- [45] MCLAY, L. A. **A maximum expected covering location model with two types of servers.** *IIE Transactions* 41, 8 (2009), 730–741.
- [46] ALADDINI, K. **Ems response time models: A case study and analysis for the region of waterloo.** Master’s thesis, University of Waterloo, 2010.
- [47] CORVEY, W. J., VIEWEG, S., ROOD, T., AND PALMER, M. **Twitter in mass emergency: what nlp techniques can contribute.** In *HLT-NAACL 2010* (2010).
- [48] DING, R., MCCARTHY, M. L., DESMOND, J. S., LEE, J. S., ARONSKY, D., AND ZEGER, S. L. **Characterizing waiting room time, treatment time, and boarding time in the emergency department using quantile regression.** *Academic Emergency Medicine* 17, 8 (July 2010), 813–823.
- [49] SORENSEN, P., AND CHURCH, R. **Integrating expected coverage and local reliability for emergency medical services location problems.** *Socio-Economic Planning Sciences* 44, 1 (2010), 8–18.
- [50] CHANG, C.-C., AND LIN, C.-J. **LIBSVM.** *ACM Transactions on Intelligent Systems and Technology* 2, 3 (Apr. 2011), 1–27.

- [51] LUXEN, D., AND VETTER, C. **Real-time routing with openstreetmap data**. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (New York, NY, USA, 2011), GIS '11, ACM, pp. 513–516.
- [52] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. **Scikit-learn: Machine learning in Python**. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [53] VANDEVENTER, S., STUDNEK, J. R., GARRETT, J. S., WARD, S. R., STALEY, K., AND BLACKWELL, T. **The association between ambulance hospital turnaround times and patient acuity, destination hospital, and time of day**. *Prehospital Emergency Care* 15, 3 (Apr. 2011), 366–370.
- [54] ABOUELJINANE, L., JEMAI, Z., AND SAHIN, E. **Reducing ambulance response time using simulation: The case of val-de-marne department emergency medical service**. In *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)* (Dec. 2012), IEEE.
- [55] CHEVALIER, P., THOMAS, I., GERAETS, D., GOETGHEBEUR, E., JANSSENS, O., PEETERS, D., AND PLASTRIA, F. **Locating fire stations: An integrated approach for belgium**. *Socio-Economic Planning Sciences* 46, 2 (June 2012), 173–182.
- [56] CONE, D. C., MIDDLETON, P. M., AND POUR, S. M. **Analysis and impact of delays in ambulance to emergency department handovers**. *Emergency Medicine Australasia* 24, 5 (Aug. 2012), 525–533.
- [57] DO, Y. K., FOO, K., NG, Y. Y., AND ONG, M. E. H. **A quantile regression analysis of ambulance response time**. *Prehospital Emergency Care* 17, 2 (Dec. 2012), 170–176.
- [58] GRAVES, A. **Supervised Sequence Labelling with Recurrent Neural Networks**. Springer Berlin Heidelberg, 2012.
- [59] SCHAPIRE, R. E., AND FREUND, Y. **Boosting**. Adaptive Computation and Machine Learning series. MIT Press, London, England, May 2012.
- [60] SUN, Y., TEOW, K. L., HENG, B. H., OOI, C. K., AND TAY, S. Y. **Real-time prediction of waiting time in the emergency department, using quantile regression**. *Annals of Emergency Medicine* 60, 3 (Sept. 2012), 299–308.

- [61] BERGSTRA, J., YAMINS, D., AND COX, D. D. **Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures.** In *Proceedings of the 30th International Conference on International Conference on Machine Learning* (2013), ICML'13, JMLR, p. I–115–I–123.
- [62] CLAREY, A., ALLEN, M., BRACE-MCDONNELL, S., AND COOKE, M. W. **Ambulance handovers: can a dedicated ED nurse solve the delay in ambulance turnaround times?** *Emergency Medicine Journal* 31, 5 (May 2013), 419–420.
- [63] LIU, Y., YUAN, Y., HUI LI, Y., AND PANG, H. **A chance constrained programming model for reliable emergency vehicles relocation problem.** *Procedia - Social and Behavioral Sciences* 96 (2013), 671–682. Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013).
- [64] MANAVIZADEH, N., SADAT HOSSEINI, N., RABBANI, M., AND JOLAI, F. **A simulated annealing algorithm for a mixed model assembly u-line balancing type-i problem considering human efficiency and just-in-time approach.** *Computers and Industrial Engineering* 64, 2 (2013), 669–685.
- [65] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. **Efficient estimation of word representations in vector space**, 2013.
- [66] ZARANDI, M. H. F., DAVARI, S., AND SISAKHT, S. A. H. **The large-scale dynamic maximal covering location problem.** *Mathematical and Computer Modelling* 57, 3 (2013), 710–719.
- [67] NOGUEIRA, F. **Bayesian Optimization: Open source constrained global optimization tool for Python**, 2014–.
- [68] CARTER, A. J., OVERTON, J., TERASHIMA, M., AND CONE, D. C. **Can emergency medical services use turnaround time as a proxy for measuring ambulance offload time?** *The Journal of Emergency Medicine* 47, 1 (July 2014), 30–35.
- [69] DWORK, C., ROTH, A., AND OTHERS. **The algorithmic foundations of differential privacy.** *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
- [70] KARABATSOS, G. **Fast marginal likelihood estimation of the ridge parameter(s) in ridge regression and generalized ridge regression for big data**, 2014.
- [71] KINGMA, D. P., AND BA, J. **Adam: A method for stochastic optimization**, 2014.
- [72] MALEKI, M., MAJLESINASAB, N., AND SEPEHRI, M. M. **Two new models for redeployment of ambulances.** *Computers and Industrial Engineering* 78 (2014), 271–284.

- [73] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. **Glove: Global vectors for word representation**. In *Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1532–1543.
- [74] CHOLLET, F., AND OTHERS. **Keras**. <https://keras.io>, 2015.
- [75] LANTZ, B. **Machine Learning with R -**, 2 ed. Packt Publishing, Birmingham, England, July 2015.
- [76] LEE, Y. J., SHIN, S. D., LEE, E. J., CHO, J. S., AND CHA, W. C. **Emergency department overcrowding and ambulance turnaround time**. *PLOS ONE* 10, 6 (June 2015), e0130758.
- [77] LÓPEZ, J. **Optimización multi-objetivo**. Editorial de la Universidad Nacional de La Plata (EDULP), 2015.
- [78] TEMNIKOVA, I., CASTILLO, C., AND VIEWEG, S. **Emterms 1.0: A terminological resource for crisis tweets**. In *ISCRAM* (2015).
- [79] CHEN, A. Y., LU, T.-Y., MA, M. H.-M., AND SUN, W.-Z. **Demand forecast using data analytics for the preallocation of ambulances**. *IEEE Journal of Biomedical and Health Informatics* 20, 4 (July 2016), 1178–1187.
- [80] CHEN, A. Y., LU, T.-Y., MA, M. H.-M., AND SUN, W.-Z. **Demand forecast using data analytics for the preallocation of ambulances**. *IEEE Journal of Biomedical and Health Informatics* 20, 4 (July 2016), 1178–1187.
- [81] CHEN, T., AND GUESTRIN, C. **XGBoost: A scalable tree boosting system**. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD '16, ACM, pp. 785–794.
- [82] COOK, B. L., PROGOVAC, A. M., CHEN, P., MULLIN, B., HOU, S., AND BACA-GARCIA, E. **Novel use of natural language processing (NLP) to predict suicidal ideation and psychiatric symptoms in a text-based mental health intervention in madrid**. *Computational and Mathematical Methods in Medicine 2016* (2016), 1–8.
- [83] FERSINI, E., MESSINA, E., AND POZZI, F. A. **Earthquake management: a decision support system based on natural language processing**. *Journal of Ambient Intelligence and Humanized Computing* 8, 1 (Apr. 2016), 37–45.
- [84] GOODFELLOW, I., BENGIO, Y., COURVILLE, A., AND BENGIO, Y. **Deep learning**, vol. 1. MIT press Cambridge, 2016.

- [85] NEHME, Z., ANDREW, E., AND SMITH, K. **Factors influencing the timeliness of emergency medical service response to time critical emergencies.** *Prehospital Emergency Care* 20, 6 (Aug. 2016), 783–791.
- [86] NGUYEN, D. T., MANNAI, K. A. A., JOTY, S., SAJJAD, H., IMRAN, M., AND MITRA, P. **Rapid classification of crisis-related data on social networks using convolutional neural networks,** 2016.
- [87] PENN, C., KOOLE, T., AND NATTRASS, R. **When seconds count: A study of communication variables in the opening segment of emergency calls.** *Journal of Health Psychology* 22, 10 (Feb. 2016), 1256–1264.
- [88] PERRY, M., AND CARTER, D. **The ethics of ambulance ramping.** *Emergency Medicine Australasia* 29, 1 (July 2016), 116–118.
- [89] RARDIN, R. L. **Optimization in operations research,** 2 ed. Pearson, Upper Saddle River, NJ, Jan. 2016.
- [90] SANKAR, S. H., JAYADEV, K., SURAJ, B., AND APARNA, P. **A comprehensive solution to road traffic accident detection and ambulance management.** In *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)* (Nov. 2016), IEEE.
- [91] SJARDIN, B., MASSARON, L., AND BOSCHETTI, A. **Large Scale Machine Learning with Python.** Packt Publishing, Birmingham, 2016.
- [92] BOUAJAJA, S., AND DRIDI, N. **A survey on human resource allocation problem and its applications.** *Operational Research* 17 (2017), 339–369.
- [93] CHOLLET, F. **Deep learning with python.** Manning Publications, New York, NY, Oct. 2017.
- [94] DE LA MOTA, I. F., PEREZ, E. S., AND GARDUNO, A. V. **Optimization and simulation of an ambulance location problem.** In *2017 Winter Simulation Conference (WSC)* (Dec. 2017), IEEE.
- [95] DIBENE, J. C., MALDONADO, Y., VERA, C., DE OLIVEIRA, M., TRUJILLO, L., AND SCHÜTZE, O. **Optimizing the location of ambulances in tijuana, mexico.** *Computers in Biology and Medicine* 80 (2017), 107–115.
- [96] DING, Y., LI, B., ZHAO, Y., AND CHENG, C. **Scoring tourist attractions based on sentiment lexicon.** In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (Mar. 2017), IEEE.

- [97] DU, S., LI, T., GONG, X., YANG, Y., AND HORNG, S. J. **Traffic flow forecasting based on hybrid deep learning framework**. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)* (nov 2017), IEEE.
- [98] FUKUCHI, K., TRAN, Q. K., AND SAKUMA, J. **Differentially private empirical risk minimization with input perturbation**. In *Discovery Science*. Springer International Publishing, 2017, pp. 82–90.
- [99] GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow**, 1 ed., vol. 1. O'Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472, 7 2017.
- [100] GREFF, K., SRIVASTAVA, R. K., KOUTNIK, J., STEUNEBRINK, B. R., AND SCHMIDHUBER, J. **LSTM: A search space odyssey**. *IEEE Transactions on Neural Networks and Learning Systems* 28, 10 (oct 2017), 2222–2232.
- [101] GRILLI, L., AND RUSSO, M. A. **Optimal allocation of human resources in the emergency ambulance service**. *Applied Mathematical Sciences* 11, 63 (2017), 3121–3128.
- [102] KE, G., AND OTHERS. **Lightgbm: A highly efficient gradient boosting decision tree**. *Advances in neural information processing systems* 30 (2017), 3146–3154.
- [103] LIU, Y., WANG, Y., YANG, X., AND ZHANG, L. **Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models**. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)* (oct 2017), IEEE.
- [104] LU, X. S., ZHOU, M., AND QI, L. **Analyzing temporal-spatial evolution of rare events by using social media data**. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (Oct. 2017), IEEE.
- [105] SHI, X., LI, Q., QI, Y., HUANG, T., AND LI, J. **An accident prediction approach based on XGBoost**. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)* (nov 2017), IEEE.
- [106] SHOKRI, R., STRONATI, M., SONG, C., AND SHMATIKOV, V. **Membership inference attacks against machine learning models**. In *2017 IEEE Symposium on Security and Privacy (SP)* (May 2017), IEEE.
- [107] SONG, C., RISTENPART, T., AND SHMATIKOV, V. **Machine learning models that remember too much**. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Oct. 2017), ACM.

- [108] TAYLOR, S. J., AND LETHAM, B. **Forecasting at scale**. *PeerJ Preprints* 5:e3190v2 (Sept. 2017).
- [109] ALAJALI, W., ZHOU, W., AND WEN, S. **Traffic flow prediction for road intersection safety**. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI)* (oct 2018), IEEE.
- [110] BEAL, L., HILL, D., MARTIN, R., AND HEDENGREN, J. **Gekko optimization suite**. *Processes* 6, 8 (2018), 106.
- [111] BÜRGER, A., WNENT, J., BOHN, A., JANTZEN, T., BRENNER, S., LEFERING, R., SEEWALD, S., GRÄSNER, J.-T., AND FISCHER, M. **The effect of ambulance response time on survival following out-of-hospital cardiac arrest**. *Deutsches Arzteblatt Online* (Aug. 2018).
- [112] ENAYATI, S., MAYORGA, M. E., RAJAGOPALAN, H. K., AND SAYDAM, C. **Real-time ambulance redeployment approach to improve service coverage with fair and restricted workload for ems providers**. *Omega* 79 (2018), 67–80.
- [113] FAN, L. **Image pixelization with differential privacy**. In *Data and Applications Security and Privacy XXXII*. Springer International Publishing, 2018, pp. 148–162.
- [114] FRAZIER, P. I. **Bayesian Optimization**. INFORMS, Oct. 2018.
- [115] HYNDMAN, R. J., AND ATHANASOPOULOS, G. **Forecasting: principles and practice**. OTexts, 2018.
- [116] LI, M., VANBERKEL, P., AND CARTER, A. J. E. **A review on ambulance offload delay literature**. *Health Care Management Science* 22, 4 (July 2018), 658–675.
- [117] SERMET, Y., AND DEMIR, I. **An intelligent system on knowledge generation and communication about flooding**. *Environmental Modelling & Software* 108 (Oct. 2018), 51–60.
- [118] WU, X., LU, X., AND LEUNG, H. **A video based fire smoke detection using robust AdaBoost**. *Sensors* 18, 11 (Nov. 2018), 3780.
- [119] ARIK, S. O., AND PFISTER, T. **Tabnet: Attentive interpretable tabular learning**. *arXiv preprint arXiv:1908.07442* (2019).
- [120] BLOMBERG, S. N., AND OTHERS. **Machine learning as a supportive tool to recognize cardiac arrest in emergency calls**. *Resuscitation* 138 (May 2019), 322–329.

- [121] BYRNE, J. P., MANN, N. C., DAI, M., MASON, S. A., KARANICOLAS, P., RIZOLI, S., AND NATHENS, A. B. **Association between emergency medical service response time and motor vehicle crash mortality in the united states.** *JAMA Surgery* 154, 4 (Apr. 2019), 286.
- [122] CERNA, S., GUYEUX, C., ARCOLEZI, H. H., LOTUFO, A. D. P., COUTURIER, R., AND ROYER, G. **Long short-term memory for predicting firemen interventions.** In *6th International Conference on Control, Decision and Information Technologies (CoDIT 2019)* (Paris, France, apr 2019).
- [123] COUCHOT, J.-F., GUYEUX, C., AND ROYER, G. **Anonymously forecasting the number and nature of firefighting operations.** In *Proceedings of the 23rd International Database Applications & Engineering Symposium on - IDEAS19* (2019), ACM Press.
- [124] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. **Bert: Pre-training of deep bidirectional transformers for language understanding**, 2019.
- [125] EL ITANI, B., ABDELAZIZ, F. B., AND MASRI, H. **Ambulance allocation models: A review.** In *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)* (Apr. 2019), IEEE.
- [126] FRANCESCO ARCHETTI, A. C. **Bayesian Optimization and Data Science**, 1st ed. 2019 ed. SpringerBriefs in Optimization. Springer International Publishing, 2019.
- [127] GUYEUX, C., NICOD, J.-M., VARNIER, C., MASRY, Z. A., ZERHOUNY, N., OMRI, N., AND ROYER, G. **Firemen prediction by using neural networks: A real case study.** In *Advances in Intelligent Systems and Computing*. Springer International Publishing, Aug. 2019, pp. 541–552.
- [128] HASAN, M. R., MALIHA, M., AND ARIFUZZAMAN, M. **Sentiment analysis with NLP on twitter data.** In *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)* (July 2019), IEEE.
- [129] KARAMI, A., SHAH, V., VAEZI, R., AND BANSAL, A. **Twitter speaks: A case of national disaster situational awareness.** *Journal of Information Science* 46, 3 (Mar. 2019), 313–324.
- [130] LE, H., VIAL, L., FREJ, J., SEGONNE, V., COAVOUX, M., LECOUTEUX, B., AL-LAUZEN, A., CRABBÉ, B., BESACIER, L., AND SCHWAB, D. **Flaubert: Unsupervised language model pre-training for french.** *CoRR abs/1912.05372* (2019).
- [131] LEE, D. W., MOON, H. J., AND HEO, N. H. **Association between ambulance response time and neurologic outcome in patients with cardiac arrest.** *The American Journal of Emergency Medicine* 37, 11 (Nov. 2019), 1999–2003.

- [132] LIAN, X., MELANCON, S., PRESTA, J.-R., REEVESMAN, A., SPIERING, B., AND WOODBRIDGE, D. **Scalable real-time prediction and analysis of san francisco fire department response times**. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)* (Aug. 2019), IEEE.
- [133] MARTIN, L., MÜLLER, B., SUÁREZ, P. J. O., DUPONT, Y., ROMARY, L., DE LA CLERGERIE, É. V., SEDDAH, D., AND SAGOT, B. **Camembert: a tasty french language model**. *CoRR abs/1911.03894* (2019).
- [134] MYOUNG KWON, J., AND OTHERS. **Deep-learning-based out-of-hospital cardiac arrest prognostic system to predict clinical outcomes**. *Resuscitation* 139 (June 2019), 84–91.
- [135] PIRKLBAUER, K., AND FINDLING, R. D. **Predicting the category of fire department operations**. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services* (Dec. 2019), ACM.
- [136] PREUM, S., SHU, S., HOTAKI, M., WILLIAMS, R., STANKOVIC, J., AND ALEMZADEH, H. **CognitiveEMS**. *ACM SIGBED Review* 16, 2 (Aug. 2019), 51–60.
- [137] RASEL, R. I., SULTANA, N., ISLAM, G. A., ISLAM, M., AND MEESAD, P. **Spatio-temporal seismic data analysis for predicting earthquake: Bangladesh perspective**. In *2019 Research, Invention, and Innovation Congress (RI2C)* (Dec. 2019), IEEE.
- [138] RHODES, B. **Skyfield: High precision research-grade positions for planets and Earth satellites generator**, July 2019.
- [139] SHAFAF, N., AND MALEK, H. **Applications of machine learning approaches in emergency medicine; a review article**. *Archives of Academic Emergency Medicine* 7, 1 (July 2019).
- [140] SU QIANG, YANG WEI, W. Q.-G. **Ambulance location planning considering the spatial randomness of demand**. *Chinese Journal of Management Science* 27, 10 (2019), 110.
- [141] WANG, C., LIU, L., XU, C., AND LV, W. **Predicting future driving risk of crash-involved drivers based on a systematic machine learning framework**. *International Journal of Environmental Research and Public Health* 16, 3 (Jan. 2019), 334.

- [142] YANG, W., SU, Q., HUANG, S. H., WANG, Q., ZHU, Y., AND ZHOU, M. **Simulation modeling and optimization for ambulance allocation considering spatiotemporal stochastic demand.** *Journal of Management Science and Engineering* 4, 4 (Dec. 2019), 252–265.
- [143] ZHAO, H., YU, H., LI, D., MAO, T., AND ZHU, H. **Vehicle accident risk prediction based on AdaBoost-SO in VANETs.** *IEEE Access* 7 (2019), 14549–14557.
- [144] ZHENG, M., LI, T., ZHU, R., CHEN, J., MA, Z., TANG, M., CUI, Z., AND WANG, Z. **Traffic accident’s severity prediction: A deep-learning approach-based CNN network.** *IEEE Access* 7 (2019), 39897–39910.
- [145] ANASTASSOPOULOU, C., RUSSO, L., TSAKRIS, A., AND SIETTOS, C. **Data-based analysis, modelling and forecasting of the COVID-19 outbreak.** *PLOS ONE* 15, 3 (Mar. 2020), e0230405.
- [146] ARCOLEZI, H. H., COUCHOT, J.-F., CERNA, S., GUYEUX, C., ROYER, G., BOUNA, B. A., AND XIAO, X. **Forecasting the number of firefighter interventions per region with local-differential-privacy-based data.** *Computers & Security* 96 (Sept. 2020), 101888.
- [147] BÉLANGER, V., LANZARONE, E., NICOLETTA, V., RUIZ, A., AND SORIANO, P. **A recursive simulation-optimization framework for the ambulance location and dispatching problem.** *European Journal of Operational Research* 286, 2 (Oct. 2020), 713–725.
- [148] CARROLL, W., STRENGER, V., EBER, E., PORCARO, F., CUTRERA, R., FITZGERALD, D., AND BALFOUR-LYNN, I. **European and united kingdom COVID-19 pandemic experience: The same but different.** *Paediatric Respiratory Reviews* 35 (Sept. 2020), 50–56.
- [149] CARVALHO, A., CAPTIVO, M., AND MARQUES, I. **Integrating the ambulance dispatching and relocation problems to maximize system’s preparedness.** *European Journal of Operational Research* 283, 3 (June 2020), 1064–1080.
- [150] CERNA, S., GUYEUX, C., ARCOLEZI, H. H., COUTURIER, R., AND ROYER, G. **A comparison of LSTM and XGBoost for predicting firemen interventions.** In *Trends and Innovations in Information Systems and Technologies*. Springer International Publishing, 2020, pp. 424–434.
- [151] CERNA, S., GUYEUX, C., ARCOLEZI, H. H., AND ROYER, G. **Boosting methods for predicting firemen interventions.** In *2020 11th International Conference on Information and Communication Systems (ICICS)* (Apr. 2020), IEEE, pp. 001–006.

- [152] CERNA, S., GUYEUX, C., ROYER, G., CHEVALLIER, C., AND PLUMEREL, G. **Predicting fire brigades operational breakdowns: A real case study.** *Mathematics* 8, 8 (Aug. 2020), 1383.
- [153] CHAMIKARA, M. A. P., BERTOK, P., KHALIL, I., LIU, D., AND CAMTEPE, S. **Privacy preserving face recognition utilizing differential privacy.** *Computers & Security* 97 (Oct. 2020), 101951.
- [154] CHINAZZI, M., DAVIS, J. T., AJELLI, M., GIOANNINI, C., LITVINOVA, M., MERLER, S., Y PIONTTI, A. P., MU, K., ROSSI, L., SUN, K., VIBOUD, C., XIONG, X., YU, H., HALLORAN, M. E., LONGINI, I. M., AND VESPIGNANI, A. **The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak.** *Science* 368, 6489 (Mar. 2020), 395–400.
- [155] DA SILVA, R. G., RIBEIRO, M. H. D. M., MARIANI, V. C., AND DOS SANTOS COELHO, L. **Forecasting brazilian and american COVID-19 cases based on artificial intelligence coupled with climatic exogenous variables.** *Chaos, Solitons & Fractals* 139 (Oct. 2020), 110027.
- [156] GONIEWICZ, K., KHORRAM-MANESH, A., HERTELENDY, A. J., GONIEWICZ, M., NAYLOR, K., AND BURKLE, F. M. **Current response and management decisions of the european union to the COVID-19 outbreak: A review.** *Sustainability* 12, 9 (May 2020), 3838.
- [157] GRASSELLI, G., PESENTI, A., AND CECCONI, M. **Critical care utilization for the COVID-19 outbreak in lombardy, italy.** *JAMA* 323, 16 (Apr. 2020), 1545.
- [158] HOLMÉN, J., HERLITZ, J., RICKSTEN, S.-E., STRÖMSÖE, A., HAGBERG, E., AXELSSON, C., AND RAWSHANI, A. **Shortening ambulance response time increases survival in out-of-hospital cardiac arrest.** *Journal of the American Heart Association* 9, 21 (Nov. 2020).
- [159] HUANG, C., WANG, Y., LI, X., REN, L., ZHAO, J., HU, Y., ZHANG, L., FAN, G., XU, J., GU, X., CHENG, Z., YU, T., XIA, J., WEI, Y., WU, W., XIE, X., YIN, W., LI, H., LIU, M., XIAO, Y., GAO, H., GUO, L., XIE, J., WANG, G., JIANG, R., GAO, Z., JIN, Q., WANG, J., AND CAO, B. **Clinical features of patients infected with 2019 novel coronavirus in wuhan, china.** *The Lancet* 395, 10223 (Feb. 2020), 497–506.
- [160] KANG, D.-Y., AND OTHERS. **Artificial intelligence algorithm to predict the need for critical care in prehospital emergency medical services.** *Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine* 28, 1 (Mar. 2020).

- [161] KANG, Y., LIU, Y., NIU, B., TONG, X., ZHANG, L., AND WANG, W. **Input perturbation: A new paradigm between central and local differential privacy.** *arXiv preprint arXiv:2002.08570* (2020).
- [162] KOZLOWSKI, D., LANNELONGUE, E., SAUDEMONT, F., BENAMARA, F., MARI, A., MORICEAU, V., AND BOUMADANE, A. **A three-level classification of french tweets in ecological crises.** *Information Processing & Management* 57, 5 (Sept. 2020), 102284.
- [163] KUO, Y.-H., CHAN, N. B., LEUNG, J. M., MENG, H., SO, A. M.-C., TSOI, K. K., AND GRAHAM, C. A. **An integrated approach of machine learning and systems thinking for waiting time prediction in an emergency department.** *International Journal of Medical Informatics* 139 (July 2020), 104143.
- [164] LI, C. **Optimization of human resources allocation for airport security check business based on genetic algorithm.** In *2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)* (2020), pp. 983–986.
- [165] LI, Q., GUAN, X., WU, P., WANG, X., ZHOU, L., TONG, Y., REN, R., LEUNG, K. S., LAU, E. H., WONG, J. Y., XING, X., XIANG, N., WU, Y., LI, C., CHEN, Q., LI, D., LIU, T., ZHAO, J., LIU, M., TU, W., CHEN, C., JIN, L., YANG, R., WANG, Q., ZHOU, S., WANG, R., LIU, H., LUO, Y., LIU, Y., SHAO, G., LI, H., TAO, Z., YANG, Y., DENG, Z., LIU, B., MA, Z., ZHANG, Y., SHI, G., LAM, T. T., WU, J. T., GAO, G. F., COWLING, B. J., YANG, B., LEUNG, G. M., AND FENG, Z. **Early transmission dynamics in wuhan, china, of novel coronavirus–infected pneumonia.** *New England Journal of Medicine* 382, 13 (Mar. 2020), 1199–1207.
- [166] LIN, A. X., HO, A. F. W., CHEONG, K. H., LI, Z., CAI, W., CHEE, M. L., NG, Y. Y., XIAO, X., AND ONG, M. E. H. **Leveraging machine learning techniques and engineering of multi-nature features for national daily regional ambulance demand prediction.** *International Journal of Environmental Research and Public Health* 17, 11 (June 2020), 4179.
- [167] LO, H.-Y., CHAOU, C.-H., CHANG, Y.-C., NG, C.-J., AND CHEN, S.-Y. **Prediction of emergency department volume and severity during a novel virus pandemic: Experience from the COVID-19 pandemic.** *The American Journal of Emergency Medicine* (Aug. 2020).
- [168] MORELLO, T. F., RAMOS, R. M., ANDERSON, L. O., OWEN, N., ROSAN, T. M., AND STEIL, L. **Predicting fires for policy making: Improving accuracy of fire brigade allocation in the brazilian amazon.** *Ecological Economics* 169 (Mar. 2020), 106501.

- [169] RASU, A. K. M. **Hands-On Python Natural Language Processing: Explore tools and techniques to analyze and process text with a view to building real-world NLP applications.** Packt Publishing Ltd, 2020.
- [170] RIBEIRO, M. H. D. M., DA SILVA, R. G., MARIANI, V. C., AND DOS SANTOS COELHO, L. **Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for brazil.** *Chaos, Solitons & Fractals* 135 (June 2020), 109853.
- [171] SOHRABI, C., ALSAFI, Z., O'NEILL, N., KHAN, M., KERWAN, A., AL-JABIR, A., IOSIFIDIS, C., AND AGHA, R. **World health organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19).** *International Journal of Surgery* 76 (Apr. 2020), 71–76.
- [172] STOECKLIN, S. B., ROLLAND, P., SILUE, Y., MAILLES, A., CAMPESE, C., SIMONDON, A., MECHAIN, M., MEURICE, L., NGUYEN, M., BASSI, C., YAMANI, E., BEHILLIL, S., ISMAEL, S., NGUYEN, D., MALVY, D., LESCURE, F. X., GEORGES, S., LAZARUS, C., TABAÏ, A., STEMPFELET, M., ENOUF, V., COIGNARD, B., AND AND, D. L.-B. **First cases of coronavirus disease 2019 (COVID-19) in france: surveillance, investigations and control measures, january 2020.** *Eurosurveillance* 25, 6 (Feb. 2020).
- [173] WAJID, S., NEZAMUDDIN, N., AND UNNIKRISHNAN, A. **Optimizing ambulance locations for coverage enhancement of accident sites in south delhi.** *Transportation Research Procedia* 48 (2020), 280–289. Recent Advances and Emerging Issues in Transport Research – An Editorial Note for the Selected Proceedings of WCTR 2019 Mumbai.
- [174] WANG, C., HORBY, P. W., HAYDEN, F. G., AND GAO, G. F. **A novel coronavirus outbreak of global health concern.** *The Lancet* 395, 10223 (Feb. 2020), 470–473.
- [175] WU, J. T., LEUNG, K., AND LEUNG, G. M. **Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in wuhan, china: a modelling study.** *The Lancet* 395, 10225 (Feb. 2020), 689–697.
- [176] ARCOLEZI, H., CERNA, S., GUYEUX, C., AND COUCHOT, J.-F. **Preserving geoindistinguishability of the emergency scene to predict ambulance response time.** *Mathematical and Computational Applications* 26, 3 (Aug. 2021), 56.
- [177] ARUNKUMAR, K., KALAGA, D. V., KUMAR, C. M. S., CHILKOOR, G., KAWAJI, M., AND BRENZA, T. M. **Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-regressive integrated moving average (ARIMA)**

- and seasonal auto-regressive integrated moving average (SARIMA).** *Applied Soft Computing* 103 (May 2021), 107161.
- [178] CERNA, S., ARCOLEZI, H. H., GUYEUX, C., ROYER-FEY, G., AND CHEVALLIER, C. **Machine learning-based forecasting of firemen ambulances' turnaround time in hospitals, considering the COVID-19 impact.** *Applied Soft Computing* 109 (Sept. 2021), 107561.
- [179] DAIRI, A., HARROU, F., ZEROUAL, A., HITTAWA, M. M., AND SUN, Y. **Comparative study of machine learning methods for COVID-19 transmission forecasting.** *Journal of Biomedical Informatics* 118 (June 2021), 103791.
- [180] DEVARAJ, J., ELAVARASAN, R. M., PUGAZHENDHI, R., SHAFIULLAH, G., GANESAN, S., JEYSREE, A. K., KHAN, I. A., AND HOSSAIN, E. **Forecasting of COVID-19 cases using deep learning models: Is it reliable and practically significant?** *Results in Physics* 21 (Feb. 2021), 103817.
- [181] JIAO, Y. **Human resource allocation method based on multi objective optimization.** In *2021 International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS)* (2021), pp. 825–828.
- [182] MCCANDLESS, D., EVANS, T., QUICK, M., HOLLOWOOD, E., MILES, C., HAMPSON, D., AND GEERE, D. **World's biggest data breaches & hacks,** jan 2021. Available online: <https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/> (accessed on 18 February 2021).
- [183] MINAEE, S., KALCHBRENNER, N., CAMBRIA, E., NIKZAD, N., CHENAGHLU, M., AND GAO, J. **Deep learning based text classification: A comprehensive review,** 2021.
- [184] MOURTZIS, D., SIATRAS, V., ANGELOPOULOS, J., AND PANOPOULOS, N. **An intelligent model for workforce allocation taking into consideration the operator skills.** *Procedia CIRP* 97 (2021), 196–201. 8th CIRP Conference of Assembly Technology and Systems.
- [185] RAHIMI, I., CHEN, F., AND GANDOMI, A. H. **A review on COVID-19 forecasting models.** *Neural Computing and Applications* (Feb. 2021).
- [186] SILVA, A., ALOISE, D., COELHO, L. C., AND ROCHA, C. **Heuristics for the dynamic facility location problem with modular capacities.** *European Journal of Operational Research* 290, 2 (Apr. 2021), 435–452.
- [187] STEWART, J., AND OTHERS. **Applications of machine learning to undifferentiated chest pain in the emergency department: A systematic review.** *PLOS ONE* 16, 8 (Aug. 2021), e0252612.

- [188] TANG, K. J. W., ANG, C. K. E., CONSTANTINIDES, T., RAJINIKANTH, V., ACHARYA, U. R., AND CHEONG, K. H. **Artificial intelligence and machine learning in emergency medicine.** *Biocybernetics and Biomedical Engineering* 41, 1 (Jan. 2021), 156–172.
- [189] ABOUELJINANE, L., AND FRICHI, Y. **A simulation optimization approach to investigate resource planning and coordination mechanisms in emergency systems.** *Simulation Modelling Practice and Theory* 119 (2022), 102586.
- [190] ARCOLEZI, H. H., CERNA, S., COUCHOT, J.-F., GUYEUX, C., AND MAKHOUL, A. **Privacy-preserving prediction of victim’s mortality and their need for transportation to health facilities.** *IEEE Transactions on Industrial Informatics* 18, 8 (2022), 5592–5599.
- [191] CERNA, S., GUYEUX, C., AND LAIYMANI, D. **The usefulness of NLP techniques for predicting peaks in firefighter interventions due to rare events.** *Neural Computing and Applications* 34, 12 (Feb. 2022), 10117–10132.
- [192] GRILLO, H., ALEMANY, M., AND CALDWELL, E. **Human resource allocation problem in the industry 4.0: A reference framework.** *Computers and Industrial Engineering* 169 (2022), 108110.
- [193] NITHIN BUDUMA, NIKHIL BUDUMA, J. P. **Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms**, 2 ed. O’Reilly Media, 2022.
- [194] WU, T., HUANG, L., LIANG, Z., ZHANG, X., AND ZHANG, C. **A supervised learning-driven heuristic for solving the facility location and production planning problem.** *European Journal of Operational Research* 301, 2 (Sept. 2022), 785–796.
- [195] ATMO-BFC. **Référent de la qualité de l’air, de l’énergie et du climat.** Available online: <https://www.atmo-bfc.org/> (accessed on 08 June 2022).
- [196] BISON-FUTÉ. **Les prévisions de trafic.** Available online: <https://www.bison-fute.gouv.fr> (accessed on 08 June 2022).
- [197] BLEU, F. **60% d’interventions en plus pour les pompiers du doubs pendant la canicule.** Available online: <https://www.francebleu.fr/infos/societe/60-d-interventions-en-plus-pour-les-pompiers-du-doubs-pendant-la-canicule-1498142401> (accessed on 08 June 2022).
- [198] COLLECTIVITÉS-LOCALES. **Collectivité territoriale.** Available online: <https://www.collectivites-locales.gouv.fr/> (accessed on 08 June 2022).

- [199] DEEPMIND. **Alphago - the movie**. Available online: <https://youtu.be/WXuK6gekU1Y> (accessed on 08 June 2022).
- [200] DOW, A., AND CUNNINGHAM, M. **Patient deaths after wait in ambulances lead to hospital probe**. <https://www.theage.com.au/national/victoria/patient-deaths-after-wait-in-ambulances-lead-to-hospital-probe-20200701-p557xg.html>. (Online; accessed 01 November 2020).
- [201] EAU-FRANCE. **Données hydrométriques de la banque nationale des données quantitatives relatives aux eaux de surface**. Available online: <https://www.hydro.eaufrance.fr/> (accessed on 08 June 2022).
- [202] FUNARTECH. **Hybridization of ml and or**. <https://funartech.com/approach/hybridization>. (Online; accessed 02 June 2022).
- [203] GEOPY. **Geopy**. <https://geopy.readthedocs.io/en/stable/>. (Online; accessed 02 June 2022).
- [204] HENDRICKS, D. **With hospitals full of coronavirus patients, ambulances wait hours for beds to become available**. <https://www.krgv.com/news/with-hospitals-full-of-coronavirus-patients-ambulances-wait-hours-for-beds-to-become-available>. (Online; accessed 01 November 2020).
- [205] HOMERY, T. **Quimper. l'hôpital, le calvaire des ambulanciers**. <https://www.ouest-france.fr/bretagne/quimper-29000/quimper-l-hopital-le-calvaire-des-ambulanciers-6857042>. (Online; accessed 01 November 2020).
- [206] HONNIBAL, M., AND MONTANI, I. **spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing**. Available online: <https://spacy.io/> (accessed on 08 June 2022).
- [207] INSEE. **Données statistiques**. Available online: <https://www.insee.fr/fr/statistiques?categorie=1> (accessed on 08 June 2022).
- [208] MARKETING, S. **Predictops**. Available online: <https://predictops.io/> (accessed on 08 June 2022).
- [209] MÉTÉO-FRANCE. **Service météorologique national français**. Available online: [https://donneespubliques.meteofrance.fr/?fond=produit&id\\_produit=90&id\\_rubrique=32](https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=90&id_rubrique=32) (accessed on 08 June 2022).
- [210] MÉTÉO-FRANCE. **Vigilance cards and bulletins archive**. Available online: <http://vigilance-public.meteo.fr/index.php> (accessed on 08 June 2022).

- [211] RÉPUBLIQUE-FRANÇAISE. **Plateforme ouverte des données publiques françaises.** Available online: <https://www.data.gouv.fr/fr/datasets/donnees-hospitalieres-relatives-a-lepidemie-de-covid-19/> (accessed on 08 June 2022).
- [212] RÉSEAU-SENTINELLES. **Réseau de recherche et de veille sanitaire en soins de premiers recours.** Available online: <https://www.sentiweb.fr/?page=table> (accessed on 08 June 2022).
- [213] UC-BERKELEY-RISE-LAB. **Ray - universal distributed compute framework.** Available online: <https://docs.ray.io/en/latest/> (accessed on 08 June 2022).
- [214] ULANDER, A. **Optimization based decision support tools for fire and rescue resource planning.** Available online: <http://liu.diva-portal.org/smash/get/diva2:789239/FULLTEXT01> (accessed on 08 June 2022).
- [215] WIKIPEDIA. **Covid-19 pandemic in france.** [https://en.wikipedia.org/wiki/COVID-19\\_pandemic\\_in\\_France#Lockdown\\_measures](https://en.wikipedia.org/wiki/COVID-19_pandemic_in_France#Lockdown_measures). (Online; accessed 02 June 2022).
- [216] WIKIPEDIA. **Great circle distance.** [https://en.wikipedia.org/wiki/Great-circle\\_distance](https://en.wikipedia.org/wiki/Great-circle_distance). (Online; accessed 02 June 2022).
- [217] WORLD-HEALTH-ORGANIZATION. **Naming the coronavirus disease (COVID-19) and the virus that causes it.** Available online: [https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-\(covid-2019\)-and-the-virus-that-causes-it#:~:text=The%20International%20Committee%20on%20Taxonomy,two%20viruses%20are%20different](https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it#:~:text=The%20International%20Committee%20on%20Taxonomy,two%20viruses%20are%20different) (accessed on 07 September 2020).
- [218] WORLD-HEALTH-ORGANIZATION. **Novel coronavirus – china.** Available online: <https://www.who.int/csr/don/12-january-2020-novel-coronavirus-china/en/> (accessed on 07 September 2020).
- [219] WORLD-HEALTH-ORGANIZATION. **WHO announces covid-19 outbreak a pandemic.** Available online: <https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/news/news/2020/3/who-announces-covid-19-outbreak-a-pandemic> (accessed on 07 September 2020).

# LIST OF FIGURES

1.1	The image on the left illustrates by type of operation the increase in the number of interventions over the years, from 2006 to 2021. The image on the right depicts the distribution of the operational load by type and for the last year, 2021. The data analyzed were provided by SDIS25. . . . .	7
1.2	The image on the left shows the cumulative number of interventions per hour of the day. The image on the right illustrates the cumulative number of interventions per day of the year. Both images consider data of the last 16 years. February 29th were removed for a better visualization of the pattern. . . . .	8
2.1	Scheme of the operations research process [89]. . . . .	25
2.2	Illustration of the search and objective space. . . . .	28
2.3	A schematic representation of the Bayesian optimization process [126]. . .	31
2.4	Illustration of the confusion matrix for binary classification. . . . .	32
3.1	Overview of operational process of SDIS25 when dealing with an intervention. In green, we can see the times related to the emergency call received at the call center. In orange, the main times related to the displacement of the available armament. And in blue, the times related to a possible displacement of the armament taking the victim to a hospital, clinic, morgue or other. . . . .	39
3.2	The breakage calculation is the indicator of the service quality. In green, we can see the cases that do not generate breaks. In yellow, the cases that generate simple breaks such as arriving late at the scene. In orange, the complex cases, where the engine is late to the scene and there are no other resources available in the two principal centers from the defense list to attend a new simultaneous intervention. . . . .	41

- 3.3 Breakdown calculation for case R3. In the image on the left, we present the calculation of the breaking time for the urban zone (Z1) and in the image on the right, for the semi-urban (Z2) and rural (Z3) zones. For both images, the calculation is applied to the engaged adapted engine. . . . . 41
- 3.4 Breakdown calculation for cases C2, C2BIS, and C2TER. On the left, the definitions for the urban zone (Z1) and on the right, for the semi-urban (Z2) and rural (Z3) zones. These are the cases when another adapted engine becomes available in centers 1 or 2 of the defense list and before the arrival at the scene of the engaged adapted engine. . . . . 42
- 3.5 Breakdown calculation for cases C3, C3BIS, and C3TER. On the left, the definitions for the urban zone (Z1) and on the right, for the semi-urban (Z2) and rural (Z3) zones. These are the cases when another adapted engine becomes available in centers 1 or 2 of the defense list and before the end of the intervention. . . . . 42
- 3.6 Breakdown calculation for cases C1, C1BIS, and C1TER, which are the most critical cases. On the left, the definitions for the urban zone (Z1) and on the right, for the semi-urban (Z2) and rural (Z3) zones. These are cases when there is no other adapted engine available in centers 1 or 2 of the defense list even after the end of the intervention. . . . . 43
- 4.1 Predictions for 2010. . . . . 52
- 4.2 Predictions for 2011. . . . . 52
- 4.3 Predictions for 2016. . . . . 52
- 4.4 Predictions for 2017. . . . . 52
- 4.5 Predictions for 1h - 2017 . . . . . 54
- 4.6 Exact predictions for 1h - 2017 . . . . . 54
- 4.7 Predictions for 1h - 2018 . . . . . 54
- 4.8 Exact predictions for 1h - 2018 . . . . . 54

5.1 Statistical analysis for the TT of ambulances: each row illustrates a histogram with bins of 1 minute to the TT frequency distribution (left side) and the cumulative sum of TT in hours for each day of the week and hour in the day (right side), considering each hospital. On the left side, the frequencies for the three hospitals follow a positively skewed distribution, where more than 80% of cases were less than one hour. On the right side, we can deduce that between 8h and 22h the accumulated time follows the intense human activity while awake, which is translated as more workload and longer TTs. . . . . 63

5.2 Statistical analysis for the AvTT per day considering the COVID-19 impact. The plot on the top illustrates statistics on the average turnaround time in minutes per day for 2020 and each hospital. The plot on the bottom illustrates the COVID-19 related statistics provided by Data Gouv and the SDIS25. . . . . 65

5.3 Average breakage time in seconds per day for 2020, considering service public breakdowns of the type “rescue people”. . . . . 67

5.4 Causes of public service breakdowns by month in 2019. . . . . 67

5.5 Causes of public service breakdowns by month in 2020. . . . . 67

5.6 The proposed methodology is based on ML techniques to predict the TT of an ambulance in a given hour and hospital. The first stage (dashed line flow) is based on a multivariate model of regularly spaced time series to predict the AvTT of the next hour in each hospital. The second stage (solid line flow) is a multivariate irregularly spaced time series model, which considers temporal variables of each ambulance departure, type of intervention, external variables and the previously predicted AvTT as inputs. For example, to predict the TT of ambulances departing to a hospital at 2:25h, 13:47h and 18:15h, the predicted AvTTs for 2h, 13h and 18h (highlighted) are used, respectively. . . . . 68

5.7 Illustrated example of the input data structure used for the AvTT model. In gray the hour identifier (Hour ID), in yellow the explanatory variables such as time, traffic, weather, statistics of COVID-19 cases from Data Gouv, trends in Google, number of assisted interventions, AvTT of the previous 24 hours, etc. And in orange, the AvTT to predict. . . . . 70

5.8	Illustrated example of the input data structure used for the TT model. In gray the depart identifier, in yellow the explanatory variables such as the type of intervention, the predicted AvTT for the present hour and the AvTT of the last 48h, time variables, Google trends, traffic, weather, etc. And in orange, the TT of the ambulance to predict. . . . .	71
5.9	Comparison between the real and the predicted AvTT for ambulances per hour in each hospital, namely CHRU, CHIH, and HNFC, respectively, by each ML technique and the baseline BSAvTT. . . . .	75
5.10	Comparison between the real hourly AvTT, the predictions of the best model obtained with the LGBM technique and the predictions of the baseline model BSAvTT, for each hospital (CHRU, CHIH and HNFC). It is observed that the AvTT predicted by LGBM better recognizes the patterns of the mean times and their variation through time, while the BSAvTT maintains a more constant pattern. . . . .	76
5.11	Comparison between the real TT of each ambulance for each hospital (CHRU, CHIH and HNFC), the predictions of the proposed methodology and the predictions of the baseline model BSTT. . . . .	78
6.1	Distribution of the ART variable for zones Z1, Z2, and Z3, respectively. . . .	87
6.2	Histogram of the total number of dispatched ambulances per hour in the day (left-hand plot) and cumulative ART in hours per day of the week and hour in the day (right-hand plot). . . . .	87
6.3	Illustration of predictions made by LASSO, XGBoost, LGBM, and MLP for 80 samples. . . . .	90
7.1	Ratio of victims' death ( $Nb. Deaths/Nb. Calls \times 100$ ) according to call center operators' delay time to answer the phone, the total call duration time, and the delay time taken to broadcast the alert, respectively, by time intervals in seconds. . . . .	99
7.2	Proposed ML-based methodology developed in 2 stages. Stage 1 predicts the victim's mortality, which will be used as a feature in stage 2 when predicting the need to transport the victim to an HF. . . . .	100
7.3	Pearson correlation between all predictors and targets. . . . .	103
7.4	Model performances. ROC curve of the best classifiers by technique. . . .	104
7.5	Feature importance from the best XGBoost models, considering the type <i>Gain</i> as score and the first 20 variables. . . . .	104

8.1	Number of interventions per type, early 2012 . . . . .	112
8.2	Example of possible consequences section (in French) . . . . .	113
8.3	Number of samples for each category . . . . .	114
8.4	Total Person Rescue interventions and its mean per hour series, early 2012	118
8.5	Auto-correlation graph for Total Person Rescue . . . . .	118
8.6	Illustration example of the tabular data for multilabel classification, consid- ering calendar variables . . . . .	122
9.1	Breakage Calculation Input and Output Data. . . . .	132
9.2	Number of BSC by month. . . . .	137
9.3	Number of BPS by month. . . . .	137
9.4	Causes of breakdowns of public service by month in 2019. . . . .	137
9.5	Breakdowns of public service of SAP type and by sector (Map of Doubs 2019). . . . .	138
9.6	Breakdowns of public service of INC type and by sector (Map of Doubs 2019). . . . .	138
9.7	Predictions of public service breakdowns using the best XGBoost model by month for 2019. . . . .	140
10.1	Variables with more impact on the simulation of the operational load. . . . .	144
10.2	Model M2. . . . .	146
10.3	Model M3. . . . .	146
10.4	Simulator scheme. . . . .	148
10.5	On the left, the progress of the hyperparameter optimization with HyperOpt for the depart model. On the right, the prediction results of the best depart model with XGBoost. . . . .	152
10.6	Feature importance of the best depart model with XGBoost, generated by counting the number of times a feature appears in a tree. . . . .	152
10.7	On the left, the progress of the hyperparameter optimization with HyperOpt for the arrival model. On the right, the prediction results of the best arrival model with XGBoost. . . . .	153
10.8	Feature importance of the best arrival model with XGBoost, generated by counting the number of times a feature appears in a tree. . . . .	153

10.9 Comparison of real and simulated departure times of first aid engines by 5-minute intervals. . . . .	155
10.10 Comparison of real and simulated arrival times of first aid engines by 5-minute intervals. . . . .	155
10.11 Comparison of real and simulated response times of first aid engines by 5-minute intervals. . . . .	155
10.12 Comparison of real and simulated departure times of adapted engines by 5-minute intervals. . . . .	156
10.13 Comparison of real and simulated arrival times of adapted engines by 5-minute intervals. . . . .	156
10.14 Comparison of real and simulated response times of adapted engines by 5-minute intervals. . . . .	156
10.15 Comparison of the distribution of BPS between M1, M2, and M3, over the Doubs territory. . . . .	157
11.1 Optimization scheme for engine distribution. . . . .	160
11.2 The best solution with the minimum number of RSP. On the left side, the red points indicate the RSP distribution in the Doubs for M3 and the solution. On the right side, it is shown the increment, reduction or no change of VSAV by center. . . . .	163
11.3 The best solution using the original quantity of VSAV. On the left side, the red points indicate the RSP distribution in the Doubs for M3 and the solution. On the right side, it is shown the increment, reduction or no change of VSAV by center. . . . .	163
11.4 The best solution using less VSAV than the original number. On the left side, the red points indicate the RSP distribution in the Doubs for M3 and the solution. On the right side, it is shown the increment, reduction or no change of VSAV by center. . . . .	164
12.1 Schema of the search for the best position for the new center (stage 1). . . . .	168
12.2 Schema of the search for the center configuration (stage 2). . . . .	171
12.3 Map of Doubs, highlighted in the west zone. The automated search located the new center in red, while the SDIS25 proposed the location in blue. . . . .	174
13.1 Illustration of a feasible solution for agent sizing. . . . .	178

13.2 Methodology for optimizing the personnel sizing in the new center, considering only new agents. . . . . 180

13.3 Methodology for optimizing the personnel sizing in the new center, considering new and existing agents. . . . . 187

13.4 Methodology for optimizing the personnel sizing in the new center, considering only existing agents. . . . . 189

13.5 Optimization results of new agents sizing. The figure represents all the minimized targets. In the upper left, in red, and in the lower right, in green, it is presented the minimization of the total number of agents and the Factor\_H, respectively. In orange, the minimization of the number of agents per skill is illustrated. . . . . 192

13.6 Optimization results of new and existing agents sizing. The top 2 images show the reduction of loss and breakdowns with Bayesian optimization. The lower 2 images illustrate the minimization of the number of agents and Factor\_H with the NSGA-II-AFO algorithm. . . . . 193

13.7 Optimization results of existing agents sizing. The top 2 images show the minimization of loss and breakdowns with Bayesian optimization. The lower 2 images illustrate the optimization of agents sizing with the NSGA-II-AFO Algorithm. . . . . 194

14.1 OptimOps System interface. . . . . 200



# LIST OF TABLES

3.1	Time frame for first aid and adapted engines according to the zone. . . . .	40
4.1	Analysis of the growth of interventions from 2006 to 2017. Data from SDIS25.	47
4.2	Prediction results on data 2006-2017 . . . . .	52
4.3	Hyperparameters . . . . .	53
4.4	Prediction results for 2017 and 2018 . . . . .	53
5.1	Data analysis, for each hospital, on the number of arrivals (Arr.) and Mean $\pm$ std TT values in minutes per semester during 2015-2020. . . . .	64
5.2	Data analysis, for each hospital, on the mean $\pm$ std TT values in minutes per month during 2020. . . . .	66
5.3	Prediction results for the AvTT of ambulances per hour and hospital by ML technique. . . . .	73
5.4	Search space for hyperparameters by technique and the best configuration obtained for predicting AvTT per hour for each hospital. . . . .	74
5.5	Prediction results for the TT using the proposed methodology and the BSTT model. . . . .	77
5.6	Search space for hyperparameters in LGBM and the best configuration obtained according to the method applied for predicting the TTs of each ambulance. . . . .	77
6.1	Mean and std values for the ART variable and the total number of dispatched ambulances (Nb. Amb.) per year in zones Z1, Z2, and Z3, respectively. For 2020, we only consider cases of the first semester. . . . .	88
6.2	Search space for hyperparameters by ML technique and the best configuration obtained. . . . .	90
6.3	ART prediction scores . . . . .	90

7.1	Summary statistics on the number (Nb.) of EMS calls, mortality, and transportation to HFs and the augmentation (Growth) in relation to the baseline year 2015. Total indicates the sum for Nb. and the average for Growth. . . .	98
7.2	Defined architectures for CNN with the number of neurons ( $nn$ ), pool_size ( $s$ ), and dropout rate ( $r$ ) used. . . . .	102
7.3	Results of the best models by ML technique for 2020. . . . .	103
8.1	Statistical description of the number of interventions per hour . . . . .	112
8.2	Example of a multilabelling of vigilance texts . . . . .	114
8.3	Defined architectures for LSTM . . . . .	116
8.4	Defined architectures for CNN . . . . .	116
8.5	Emergency Person Rescue prediction scores . . . . .	119
8.6	Total Person Rescue prediction scores . . . . .	119
8.7	Heating prediction scores . . . . .	119
8.8	Storm and Flood prediction scores . . . . .	119
8.9	Prediction scores using XGBoost, Emergency Person Rescue case . . . .	120
8.10	Prediction scores using XGBoost, Total Person Rescue case . . . . .	120
8.11	Prediction scores using XGBoost, Heating case . . . . .	120
8.12	Prediction scores using XGBoost, Storm/Flood case . . . . .	120
8.13	Hyperparameters search space and the best configuration for XGBoost and Random Forest multilabel models . . . . .	123
8.14	Prediction results of the multilabel models based on XGBoost and Random Forest techniques . . . . .	124
8.15	Hyperparameters search space and the best configuration for NLP multilabel models . . . . .	125
8.16	Prediction results of the multilabel models based on NLP techniques . . .	125
8.17	Accuracy results for each type of intervention, considering the models generated with the weather bulletins. . . . .	125
9.1	Statistical information of breakdowns of public service by month from 2017 to 2019. . . . .	137
9.2	Predictions of public service breakdowns by month for 2019. . . . .	139
9.3	Hyperparameters of the best XGBoost model by month for 2019. . . . .	139

9.4 Metrics of the best XGBoost model with scaled breakdowns predictions by month for 2019. . . . . 139

10.1 Basic armament to attend SAP and INC emergencies. . . . . 148

10.2 Number of breakdowns per model and per case. . . . . 154

11.1 Results of the 3 best solutions for the distribution of VSAVs. . . . . 162

12.1 Illustration of the Deployment Plan. . . . . 168

12.2 Results of configurations simulated for the new center from the automated decision (NC-AU) and from human decision (NC-HU) . . . . . 173

13.1 Example of the definition of resources for a new center . . . . . 178

13.2 The 3 best solutions for sizing new agents. . . . . 191

13.3 The 3 best solutions for sizing new and existing agents. . . . . 193

13.4 The 3 best solutions for sizing existing agents. . . . . 195





**Title:** Prediction of Interventions and Optimization of Resources Based on Machine Learning and Operations Research for Fire Departments

**Keywords:** Fire Brigade, Machine Learning, Operations Research, Neural Networks, Bayesian Optimization, Artificial Intelligence.

**Abstract:**

Around the world, fire departments constantly seek to develop strategies to reduce their response time in interventions, since it is one of the most important factors when saving lives and to measure the quality of their service. The data collected over the years on their interventions during fires, rescues, traffic accidents, etc., could be used to develop data-driven systems for decision making, understand the trends of certain events, improve the efficiency of their service, and reduce operating costs.

For this reason, the main objectives of this thesis are:

- i) Predict interventions to support decision-making in the short-term deployment of resources, and ii) Develop methodologies to optimize resources in the

long-term.

For objective *i*, we will build Machine Learning (ML) based models to predict the number of interventions in the next hours, when a peak in operational load will occur due to rare events (e.g., storms and floods), the mortality of victims, the response time to an incident, and the turnaround time of ambulances in hospitals.

For objective *ii*, we will develop optimization methods based on ML and Operations Research (OR) techniques for the creation of a quality of service indicator, an operational load simulator, the redeployment of ambulances, and the implementation of a new center.

**Titre :** Prediction of Interventions and Optimization of Resources Based on Machine Learning and Operations Research for Fire Departments

**Mots-clés :** Sapeur Pompier, Apprentissage Automatique, Recherche Opérationnelle, Réseaux Neuronaux, Optimisation Bayésienne, Intelligence Artificielle.

**Résumé :**

Dans le monde entier, les Services d'Incendie et Secours (SIS) cherchent à développer des stratégies pour réduire leur temps de réponse lors des interventions, car c'est l'un des facteurs les plus importants pour sauver des vies et pour mesurer la qualité de leur service. Les données collectés au fil des ans sur leurs interventions lors d'incendies, de sauvetages, d'accidents de la route, etc., pourraient être utilisés pour développer des systèmes de prise de décision basés sur les données, comprendre les tendances de certains événements, améliorer l'efficacité de leur service et réduire les coûts d'exploitation.

Pour cette raison, les principaux objectifs de cette thèse sont: i) Prédire les interventions pour soutenir la prise de décision dans le déploiement des ressources

à court terme, et ii) Développer des méthodologies pour optimiser les ressources à long terme.

Pour l'objectif *i*, nous construisons des modèles basés sur l'apprentissage automatique (ML) pour prédire le nombre d'interventions dans les prochaines heures, quand un pic de charge opérationnelle se produira en raison d'événements rares (par exemple, tempêtes et inondations), la mortalité des victimes, le temps de réponse à un incident et le temps de rotation des ambulances dans les hôpitaux.

Pour l'objectif *ii*, nous développerons des méthodes d'optimisation basées sur des techniques de ML et de recherche opérationnelle (OR) pour la création d'un indicateur de qualité de service, un simulateur de charge opérationnelle, le redéploiement des ambulances, et la mise en place d'un nouveau centre.