Estimation of Road Accident Risk with Machine Learning

Antoine Hébert

A Thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Master of Computer Science at
Concordia University
Montréal, Québec, Canada

May 2020

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Antoine Hébert**
Entitled: **Estimation of Road Accident Risk with Machine Learning**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Essam Mansour

_____ Examiner
Dr. Adam Krzyzak

_____ Examiner
Dr. Essam Mansour

_____ Thesis Supervisor
Dr. Tristan Glatard

_____ Thesis Supervisor
Dr. Brigitte Jaumard

Approved by _____
Dr. Leila Kosseim, Graduate Program Director

May 15, 2020 _____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Estimation of Road Accident Risk with Machine Learning

Antoine Hébert

Road accidents are an important issue for our societies, responsible for millions of deaths and injuries every year representing a very high cost for society. In this thesis, we evaluate how machine learning can be used to estimate the risk of accidents in order to help address this issue.

Previous studies have shown that machine learning can be used to identify the times and areas of a road network with increased risk of road accidents using road characteristics, weather statistics, and date-based features. In the first part of this thesis, we evaluate whether more precise models estimating the risk for smaller areas can still reach interesting performances. We assemble several public datasets and build a relatively accurate model estimating the risk of accidents within an hour on a road segment defined by intersections.

In the second part, we evaluate whether data collected by vehicle sensors during driving can be used to estimate the risk of accidents of a driver. We explore two different approaches. With the first approach, we extract features from the time series and attempt to estimate the risk based on these features using classical algorithms. With the second approach, we design a neural network directly using the time series data to estimate the risk. After extensively tuning our models, we managed to reach encouraging performances on the validation set, however, the performances of our two models on the test set were disappointing. This led us to believe that this task might not be feasible, at least with the dataset used.

Acknowledgments

First, I would like to express my deepest gratitude to my supervisors Drs. Brigitte Jaumard and Tristan Glatard for guiding me at the beginning of my master's and suggesting that I switch to the thesis-based program, I enjoyed the process of doing my thesis and might have missed this opportunity had they not reached out to me. I would also like to thank them for their guidance and their help which allowed me to present a paper at IEEE Big Data 2019. Finally, I would like to thank them for ensuring that I receive good funding for the whole duration of my Master's.

Second, I would like to thank Groupe Robert for being at the initiative of the main research project of this thesis, and for providing initial funding. In particular, I would like to thank Gilles Gervais and Ian Marineau, our correspondents from Groupe Robert, for always making themselves available for any question we might have. I also want to thank David Brillon and David Pinson our contact person from Isaac Instruments, the company designing the telemetric system used by Groupe Robert, who facilitated the transfer of the massive amount of data.

Third, I would like to thank Timothée Guédon for our successful collaboration on the first research project. I would also like to express my appreciation to Denis Ergashbaev for the many interesting and stimulating discussions we have had on the advances in machine learning and reinforcement learning.

Finally, I would like to thank Diep Nguyen and my parents for their support and encouragements.

Contribution of Authors

This thesis includes two research papers. The first paper corresponds to a research project started as a course project for the Big Data Analytics course taught by Dr. Tristan Glatard and was conducted with another student Timothée Guédon. The second project corresponds to the main research project of my thesis and was realized by myself alone under the supervision of my supervisors Drs. Brigitte Jaumard and Tristan Glatard.

The idea and the datasets for the first project were found together. Personally, I implemented the algorithm efficiently matching each accident to its closest road segment, the querying of the weather API, the interpolation of weather information at different road segments based on weather station information, and the extraction of features based on the road characteristics. I created the solar elevation feature and the weather feature based on atmospheric events. I performed the hyper-parameter tuning of the random forest and balanced random forest models. The research paper was written in collaboration with Timothée Guédon. I wrote the related work and model development sections, as well as the discussion section with the exception of the sub-section on reproducibility. Other parts were either written by Timothée alone or in close collaboration. My supervisors proof-read the paper and made suggestions and small adjustments before the submission. I presented the paper at the conference.

Contents

List of Figures **ix**

List of Tables **x**

1 Introduction **1**

1.1 Context of the thesis 1

1.2 High-Resolution Road Vehicle Collision Prediction for the City of Montreal 2

1.3 Outline 3

2 High-Resolution Road Vehicle Collision Prediction for the City of Montreal **5**

2.1 Introduction 6

2.1.1 Open Data 6

2.1.2 High-Resolution Road Vehicle Collision Prediction 7

2.1.3 The Data Imbalance Issue 8

2.1.4 Our Contributions 9

2.2 Related Work 10

2.2.1 Road Accident Prediction 10

2.2.2 Dealing with Data Imbalance 14

2.3 Datasets Integration 15

2.3.1 Open Datasets 15

2.3.2 Positive and Negative Examples Generation 16

2.4 Model Development 18
2.4.1 Implementation of Balanced Random Forest 18
2.4.2 Feature Engineering 19
2.4.3 Identifying the most Important Features 21
2.4.4 Hyper-Parameter Tuning 21
2.5 Results 22
2.5.1 Balanced Random Forest Performances 22
2.5.2 Vehicle Collision Prediction 24
2.5.3 Vehicle Collision Feature Importance 27
2.6 Discussion 27
2.6.1 Test of our Implementation of BRF on the Mammography Dataset . 28
2.6.2 Comparison of the Different Models for Road Vehicle Collisions Prediction 29
2.6.3 Real-world Performances of our Road Vehicle Collision Prediction Model 29
2.6.4 Reproducibility of the study 30
2.6.5 Future Work 31
2.7 Conclusions 32

3 Can we Estimate Truck Accident Risk from Telemetric Data using Machine Learning ? 33

3.1 Introduction 34
3.2 Related work 36
3.2.1 Road Accident Prediction 36
3.2.2 Time Series Classification 36
3.3 Datasets 39
3.4 Method 41

3.4.1 Data preprocessing 41
3.4.2 Instance creation 43
3.4.3 Labeling 44
3.4.4 Creation of training and test sets 46
3.4.5 Feature-based approach 47
3.4.6 Representation-based approach 48
3.5 Experiments and Results 51
3.6 Discussion 53
3.7 Conclusion 58

4 Conclusion and future work **59**

Bibliography **62**

List of Figures

1 Comparison of implementations: Precision-recall curves 23

2 Comparison of implementations: ROC curves 24

3 Vehicle Collision Prediction: Precision-recall curves 25

4 Vehicle Collision Prediction: ROC curves 26

5 Vehicle Collision Prediction: Precision and Recall as a Function of the Threshold Values 26

6 Feature importance computed by the Balanced Random Forest excluding the accident count feature. 28

7 Visualization of the distribution of accidents in time 41

8 Illustration of example creation from raw data with 1-hour windows and 3 windows per example. 45

9 Window of one hour of driving data 46

10 ROC curves of the feature-based model and of the neural network on the test set and on the validation sets 52

11 Visualization of the risk of accidents estimated by the representation-based model 53

List of Tables

1 Comparison of our BRF implementation with imbalanced-learn 23

2 Result Summary 25

3 Some of the parameters collected 40

4 Types of accident 42

Chapter 1

Introduction

1.1 Context of the thesis

Road accidents are an important issue representing a very high cost for society. Despite improvements in road safety, road accidents remain one of the leading causes of death, for young people between 5 and 29 years old, it is the leading cause of death [51]. The World Health Organization estimates that road accidents cause 1.35 million death and more than 20 million injuries every year in the world [51, 26]. In 2010, the cost of transport-related injuries in Canada was estimated at 3.2 billion US dollars [30].

Trucking companies are an important user of the road network, by promoting safe driving and offering training to their employees they can help to reduce significantly the number of road accidents. Indeed, in the US in 2017, large trucks were involved in 13% of fatal crashes [24]. Some of these fatalities could have been prevented, indeed according to the US Federal Motor Carrier Safety Administration, for 32% of the large truck drivers involved in fatal crashes, at least one driver-related factor was identified. The two most common driver-related factors identified were “Speeding of Any Kind” and “Distraction/Inattention”. Although already regrettable, it can be noted that this percentage remains better than for passenger vehicles for which a driver-related factor was recorded for more

than half of fatal crashes [24].

Advances in telecommunications and electronics, as well as the increasing number of sensors already installed in trucks, make it possible and cost-efficient to collect massive amounts of data from vehicles during driving. Telemetric solutions on the market offer trucking companies the opportunity to improve the management of their truck fleet by collecting some of these data and providing real-time information to fleet managers [38].

This massive amount of data represents a new opportunity to gain a better understanding of road accidents in the trucking industry. The main research project of my thesis consisted in exploring how this data can be used to identify patterns leading to road accidents. This research project was conducted in collaboration with Groupe Robert. Groupe Robert is a logistics, distribution, and transport company founded in Quebec in 1946. Groupe Robert is a North American leader in the transportation industry employing 3,500 persons and operating a fleet of 1,400 tractors and 3,000 trailers. For many years, Groupe Robert has been monitoring road accidents and infractions in which their fleet was involved. In 2017, it equipped its truck fleet with a telemetric system collecting most of the data generated by vehicle sensors during driving. We used the data collected by this system since February 2018 to explore the potential of driving telemetric data for the prediction of road accidents.

1.2 High-Resolution Road Vehicle Collision Prediction for the City of Montreal

The process of acquiring the massive amount of data collected on all the trucks of the partner company for one year took a few months. While waiting to obtain the data, after reviewing the literature in the field of road accident prediction, I initiated an additional research project linked to the subject of my main project. This project was started as a course project for the Big Data Analytics course taught by Dr. Tristan Glatard and was

conducted with another student Timothée Guédon. Our initial goal was to reproduce the state of the art in road accident prediction and to use open data provided by the city of Montreal and the Government of Canada in order to build a model providing an estimation of the risk of accidents for each area of Montreal and for each date. After looking at available datasets, we decided to experiment with building a prediction model predicting at a higher resolution than previous studies [8, 9, 42, 57, 11, 49, 62], that is to say, a model capable of providing an estimation of the risk of accidents within one hour on a road segment defined by road intersections. We also evaluated in this study a variation of the Random Forests algorithm [10] designed to help with the severe class imbalance issue inherent to accident prediction problems. This project led to the publication of a conference paper at IEEE Big Data 2019. The first chapter of this thesis entitled "High-Resolution Road Vehicle Collision Prediction for the City of Montreal" corresponds to the content of this paper.

1.3 Outline

The first chapter of this thesis presents our first study on road accident prediction and corresponds to my first paper published at IEEE Big Data 2019. For this study, we assembled three publicly available datasets: a dataset containing road vehicle collisions, a dataset describing the Canadian road network, and a dataset containing historical weather information. Using these datasets, we created meaningful features to build a high spatial and temporal resolution road accident prediction model for the island of Montreal. In this study, we also compare different machine learning algorithms, including the Balanced Random Forest algorithm [10] which we implemented ourselves in Apache Spark [63].

The second chapter of this thesis presents the main research project I performed during my Master's and corresponds to my second paper which we will submit shortly to IEEE Transactions on Intelligent Transportation Systems. In this study, we evaluate whether

driving telemetric data of a driver can be used to estimate its risk of road accidents. We experiment with two different machine learning approaches: a feature-based approach for which we extract features from the time series data using the FRESH algorithm [15] and then use the random forest algorithm to estimate the risk; and a representation-based approach for which we use a convolutional neural network learning a representation of the data in order to directly estimate the risk from the time series data.

Chapter 2

High-Resolution Road Vehicle Collision Prediction for the City of Montreal

Road accidents are an important issue of our modern societies, responsible for millions of deaths and injuries every year in the world. In Quebec only, in 2018, road accidents are responsible for 359 deaths and 33 thousands of injuries. In this paper, we show how one can leverage open datasets of a city like Montreal, Canada, to create high-resolution accident prediction models, using big data analytics. Compared to other studies in road accident prediction, we have a much higher prediction resolution, i.e., our models predict the occurrence of an accident within an hour, on road segments defined by intersections. Such models could be used in the context of road accident prevention, but also to identify key factors that can lead to a road accident, and consequently, help elaborate new policies.

We tested various machine learning methods to deal with the severe class imbalance inherent to accident prediction problems. In particular, we implemented the Balanced Random Forest algorithm, a variant of the Random Forest machine learning algorithm in Apache Spark. Interestingly, we found that in our case, Balanced Random Forest does not perform significantly better than Random Forest.

Experimental results show that 85% of road vehicle collisions are detected by our model

with a false positive rate of 13%. The examples identified as positive are likely to correspond to high risk situations. In addition, we identify the most important predictors of vehicle collisions for the area of Montreal: the count of accidents on the same road segment during previous years, the temperature, the day of the year, the hour and the visibility.

This chapter was published in the proceedings of the 7th IEEE International Conference on Big Data [37].

2.1 Introduction

The World Health Organization describes the road traffic system as the most complex and the most dangerous system with which people have to deal every day [53]. In the last few years, the number of road traffic deaths in the world climbed, reaching 1.35 million in 2016 [51]. More particularly in Quebec, Canada, 359 people were killed in 2018, more than a thousand were seriously injured and tens of thousands have suffered small injuries[54].

Meanwhile, Big Data Analytics has emerged in the last decade as a set of techniques allowing data scientists to extract meaningful information from large amounts of complex and heterogeneous data [27]. In the context of accident prediction, such techniques provide insights on the conditions leading to an increased risk of road accidents, which in return, can be used to develop traffic-related policies and prevention operations.

2.1.1 Open Data

Governments, states, provinces and municipalities collect and manage data for their internal operations. In the last decade, an open data movement has emerged that encourages governments to make the data they collect available to the public as “open data". Open data is defined as “structured data that is machine-readable, freely shared, used and built on without restrictions" [29]. Open data should be easily accessible and published under

terms that permit re-use and redistribution by anyone and for any purpose.

Open data is made possible by the progress of information technology which allows the sharing of large amounts of data. In 2009, Canada, USA, UK and New Zealand, announced new initiatives towards opening up public information. It is in this spirit that the Government of Canada launched its first-generation of the Open Data Portal in 2011 [29], giving access to several public datasets. In 2012, the city of Montreal launched its own open data portal.

2.1.2 High-Resolution Road Vehicle Collision Prediction

With the emergence of open data, governments and municipalities are publishing more and more data. At the same time, the recent progresses in Big Data Analytics have facilitated the processing of large data volumes. This makes it possible to build efficient data models for the study of road accidents.

Accident prediction has been extensively studied in the last decade. The goal of accident prediction is usually to provide a measure of the risk of accidents at different points in time and space. The occurrence of an accident is the label used to train the model, and the proposed model can be used to identify where and when the risk of accidents is significantly higher than average in order to take actions to reduce that risk. Note that the model cannot be used to predict whether an accident will occur or not. Indeed, in order to accurately predict the occurrence of an accident, additional data would be needed: the occurrence of an accident depends on many factors, including driver behavior, that cannot be easily measured.

Several studies used relatively small datasets and performed accident prediction only on a few selected roads [8, 9, 42, 57]. More recently, other studies performed accident prediction at a larger scale, such as cities or states, using deep learning[11, 49, 62]. However, unlike previous studies, they only provide an estimation of the risk of accidents for large

areas, i.e., at a coarse spatial resolution. An online article[60] presents a study of high resolution road accident prediction in the state of Utah with good performances. This article has inspired us to build a machine-learning model for high-resolution road vehicle collision prediction using public datasets. We used datasets provided by the city of Montreal and the government of Canada as part of their open data initiative. Compared to [60], we have a smaller study area, the island of Montreal, but a much higher prediction resolution. Indeed, the size and precision of our datasets made it possible to predict the occurrence of an accident within an hour on road segments defined by road intersections.

Road vehicle collision prediction can be seen as: (1) a regression problem: predicting the risk of accidents, which can be translated into different ways, or (2) a binary classification problem: predicting whether an accident will occur. We choose to approach it as a classification problem because this simpler approach facilitates the interpretation and comparison of results. In addition, classification models also provide a measure of probability that can be considered as the risk of an accident.

2.1.3 The Data Imbalance Issue

Like many real-world binary classification problems such as medical diagnosis or fraud prediction, vehicle collision prediction suffers from the data imbalance issue. This issue arises when we are interested in the prediction of a rare event. In this case, the dataset contains much less examples of the class corresponding to the rare event, the positive class. When dealing with severe data imbalance, most machine learning algorithms do not perform well. Indeed, they try to minimize the overall error rate instead of focusing on the detection of the positive class [10].

2.1.4 Our Contributions

In this study, we assembled a dataset containing road vehicle collisions, a dataset describing the Canadian road network, and a dataset containing historical weather information. Using these datasets, we created positive examples, corresponding to the occurrence of a collision, and negative examples, corresponding to the non-occurrence of a collision. For each example, we extracted from the datasets relevant features for accident prediction. Then, we built several prediction models using these examples using various machine learning algorithms. We focused on tree-based machine-learning algorithms because they have already proven their effectiveness compared to classical statistical methods [8, 9]. In addition, they allow for easier interpretation than deep learning algorithms. We first used the Random Forest algorithm[7]. We then used the Balanced Random Forest (BRF) algorithm[10], a variation of Random Forest specifically designed to better manage data imbalance. As BRF was not yet implemented in Apache Spark, we implemented it ourselves. Finally, we considered the XGBoost algorithm[13], a gradient tree boosting algorithm which has been used successfully for many machine learning problems and can handle data imbalance[12].

The contributions of this paper include:

- A demonstration of how open datasets can be combined to obtain meaningful features for road accident prediction,
- A high spatial and temporal resolution road accident prediction model for the island of Montreal,
- A comparison of three algorithms dealing with data imbalance in the context of road accident prediction,
- An implementation of Balanced Random Forest [10] in Apache Spark for efficient distributed training.

All the source code used is publicly available on Github under MIT license.

Compared to other studies in accident prediction, our study is original by the size of the datasets used and the spatial resolution of the predictions of our models. Previous studies did either use a large dataset (millions of records in total including hundreds of thousands of positive samples [11]) or predict at a high resolution on one particular road, but no study combines both aspects, which is the hallmark of our study. In terms of prediction resolution, some studies worked on only one road [8] [9] [42] while some others worked on regions (for example 5km by 5km [11] or 500m by 500m [62]). The road accident dataset we used also covers a wider time range than some studies and is about the maximum time range encountered in the related papers we studied: 7 years [62] (against 6 years in our case). For example, other studies have worked on accidents occurring during one year [8] [9] [11] [42]. In our opinion, predicting at a higher resolution yields more useful results.

The rest of this paper is organized as follows: Section 2.2 presents the related work on accident prediction and on learning with imbalanced data, Section 2.3 presents the datasets we used and how we combined them to create positive and negative examples for road accident prediction, Section 2.4 presents how we performed feature engineering, feature selection and hyper-parameter tuning, Section 2.5 presents our results and Section 2.6 discusses them. Conclusions are drawn in the last section.

2.2 Related Work

2.2.1 Road Accident Prediction

Accident prediction has been extensively studied in the last decades. Historically, variations of the Poisson regression such as the negative binomial regression were used to predict the number of accidents that occurred on a given road segment [48]. During the last decade, machine learning algorithms such as decision trees, artificial neural networks and Bayesian

networks have been used successfully to predict road accidents [8, 9, 42, 57]. Data features usually include information about the road such as number of lanes, average daily traffic, and road curvature, as well as weather information such as average precipitation and temperature.

In 2005, Chang [8] compared the performances of a negative binomial regression with that of an Artificial Neural Network (ANN) to predict the number of accidents during a year on road segments of a major freeway in Taiwan. The dataset contained data from the years 1997 and 1998, which resulted in 1,338 accidents. The ANN achieved slightly better results than negative binomial regression, with an accuracy of 61.4%. On the same dataset, Chang *et al.* [9] also used decision trees for accident prediction, to get more insights on the important variables for accident prediction. It appeared that the average daily traffic and the number of days with precipitation were the most relevant features. The decision tree reached an accuracy of 52.6%.

Lin *et al.* [42] compared the performances of Frequent Pattern trees[33] with that of Random Forest for feature selection. They used k-nearest-neighbor and Bayesian networks for real-time accident prediction on a segment of a highway. Using the mean and sometimes the standard deviation of the weather condition, the visibility, the traffic volume, the traffic speed, and the occupancy measured during the last few minutes their models predict the occurrence of an accident. They obtained the best results using the Frequent Pattern trees feature selection and achieved an accuracy of 61.7%. It should be noted that they used only a small sample of the possible negative examples, to deal with data imbalance.

Theofilatos[57] also used real-time data on two urban arterials of the city of Athens to study road accident likelihood and severity. Random Forest were used for feature selection and a Bayesian logistic regression for accident likelihood prediction. The most important features identified were the coefficients of variation of the flow per lane, the speed, and the occupancy.

In addition, many studies aim at predicting the severity of an accident using various information from the accident in order to understand what causes an accident to be fatal. Chong *et al.* [47] used decision trees, neural networks and a hybrid model using a decision tree and a neural network. They obtained the best performances with the hybrid model which reached an accuracy of 90% for the prediction of fatal injuries. They identified that the seat belt usage, the light conditions and the alcohol usage of the driver are the most important features. Abellán *et al.* [2] also studied traffic accident severity by looking at the decision rules of a decision tree using a dataset of 1,801 highway accidents. They found that the type and cause of the accident, the light condition, the sex of the driver and the weather were the most important features.

All of these studies use relatively small datasets using data from only a few years or only a few roads. Indeed, it can be hard to collect all the necessary information to perform road accident prediction on a larger scale, and dealing with big datasets is more difficult. However, more recent studies [11, 49, 62] performed accident prediction at a much larger scale, usually using deep learning models. Deep learning models can be trained online so that the whole dataset does not need to stay in memory. This makes it easier to deal with big datasets.

Chen *et al.* [11] used human mobility information coming from mobile phone GPS data and historical accident records to build a model for real-time prediction of traffic accident risk in areas of 500 by 500 meters. The risk level of an area is defined as the sum of the severity of accidents that occurred in the area during the hour. Their model achieves a Root Mean-Square Error (RMSE) of 1.0 accident severity. They compared the performance of their deep learning model with the performances of a few classical machine learning algorithms: Decision Tree, Logistic Regression and Support Vector Machine (SVM), which all got worse RMSE values of respectively 1.41, 1.41 and 1.73. We note that they have not tried the Random Forest algorithm while it usually has good prediction performances.

Najjar et al. [49], trained a convolutional neural network using historical accident data and satellite images to predict the risk of accidents on an intersection using the satellite image of the intersection. Their best model reaches an accuracy of 73%. Yuan *et al.* [62] used an ensemble of Convolutional Long Short-Term Memory (LSTM) neural networks for road accident prediction in the state of Iowa. Each neural network of the ensemble is predicting on a different spatial zone so that each neural network learns the patterns corresponding to its zone, which might be a rural zone with highways or an urban zone. They used a high-resolution rainfall dataset, a weather dataset, a road network dataset, a satellite image and the data from traffic cameras. Their model reaches an RMSE of 0.116 for the prediction of the number of accidents during a day in an area of 25 square kilometers.

These more recent studies are particularly interesting because they achieve good results for the prediction of road accidents in time and space in larger areas than previous studies which focused on a few roads. But unlike previous studies, they only provide an estimation of the risk of accidents for large areas, i.e., at a coarse spatial resolution. In our study, we decided to focus on urban accidents occurring in the island of Montreal, a 500-km^2 urban area, but with a much higher prediction resolution. We used a time resolution of one hour and a spatial resolution defined by the road segments delimited by road intersections. The road segments used have an average length of 124 meters, and 82% of the road segments are less than 200 meters long.

Some of these studies define the road accident prediction problem as a classification problem, while others define it as a regression problem. Most of the studies performing classification only report the accuracy metric which is not well suited for problems with data imbalance such as road accident prediction[35]. The studies performing regression use different definitions for the risk of accidents, which makes comparisons difficult.

2.2.2 Dealing with Data Imbalance

Road accident prediction suffers from a data imbalance issue. Indeed, a road accident is a very rare event so we have much more examples without accident, than examples with accidents available. Machine learning algorithms usually have difficulty learning from imbalanced datasets [6]. There are two main types of approaches to deal with data imbalance. The sampling approaches consist in re-sampling the dataset to make it balanced either by over sampling the minority class, by under-sampling the majority class or by doing both. Random under-sampling of the majority class usually performs better than more advanced methods like SMOTE or NearMiss [6]. The cost-based approach consists in adding weights on the examples. The negative examples receive a lower weight in order to compensate for their higher number. These weights are used differently depending on the machine learning algorithm.

Chen, Liaw, and Breiman[10] proposed two methods to deal with class imbalance when using Random Forest: Weighted Random Forest and Balanced Random Forest. Weighted Random Forest (WRF) belongs to the class of cost-based approaches. It consists in giving more weight to the minority class when building a tree: during split selection and during class prediction of each terminal node. Balanced Random Forest belongs to the class of sampling approaches. It is similar to Random Forest, but with a difference during the bootstrapping phase: for each tree of the forest, a random under-sampling of the majority class is performed in order to obtain a balanced sample. Intuitively, Balanced Random Forest is an adaptation of random under-sampling of the majority class making use of the fact that Random Forest is an ensemble method. While none of the methods is clearly better than the other in terms of predictive power, BRF has an advantage in terms of training speed because of the under-sampling. Interestingly, Wallace *et al.* [59] present a theoretical analysis of the data imbalance problem and suggest to use methods similar to Balanced Random Forest.

2.3 Datasets Integration

2.3.1 Open Datasets

We used three public datasets[16, 28, 32] provided by the city of Montreal and the government of Canada:

Montreal Vehicle Collisions[16] This dataset, provided by the city of Montreal, contains all the road collisions reported by the police occurring from 2012 to 2018 on the island of Montreal. For each accident, the dataset contains the date and localization of the accident, information on the number of injuries and deaths, the number of vehicles involved, and information on the road conditions. The dataset contains 150,000 collisions, among which 134,489 contain the date, the hour and the location of the accident. We used only these three variables since we do not have other information when no accident happened. Another dataset with all vehicle collisions in Canada is available but without the location of the accident, therefore we restrained our analysis to the city of Montreal.

National Road Network[28] This dataset, provided by the government of Canada, contains the geometry of all roads in Canada. For each road segment, a few meta-data are given. For roads in Québec, only the name of the road and the name of the location are provided. The data was available in various formats, we chose to use the Keyhole Markup Language, which is a standard of the Open Geospatial Consortium since 2008[41], This format is based on the Extensible Markup Language (XML), which makes it easier to read using existing implementations of XML parsers. From this dataset, we selected the $44,111$ road segments belonging to the island of Montreal (the dataset is separated into regions and cities).

Historical Climate Dataset[32] This dataset, provided by the government of Canada, contains hourly weather information measured at different weather stations across Canada. For each station and every hour, the dataset provides the temperature, the dew point temperature (a measure of humidity), the humidity percentage, the wind direction, the wind speed, the visibility, the atmospheric pressure, the Hmdx index (a measure of felt temperature) and the wind chill (another measure of felt temperature using wind information). This dataset also contains the observations of atmospheric phenomena such as snow, fog, rain, etc.

2.3.2 Positive and Negative Examples Generation

The accident prediction problem can be stated as a binary classification problem, where the positive class is the occurrence of an accident and the negative class is the non-occurrence of an accident on a given road at a given date and hour. For each accident, we identified the corresponding road segment using its GPS coordinates. Such time-road segment pairs are used as positive examples. For the negative examples, we generated a uniform random sample of 0.1% of the 2.3 billions possible combinations of time and road segments in order to obtain 2.3 million examples. We removed from these examples the few ones corresponding to a collision in the collision dataset in order to obtain the negative examples.

The identification of the road segments for each collision and the estimation of the weather information for each road segment made our dataset generation expensive in resources and time. We used the big data framework Apache Spark [63] to implement these dataset combination operations. Inspired by the Map Reduce programming model [19], Apache Spark's programming model introduced a new distributed collection called Resilient Distributed Dataset (RDD), which provides the "same optimization as specialized Big Data engines but using it as libraries" through a unified API. After its release in 2010, Apache Spark rapidly became the most active open-source project for Big Data [63]. As a

consequence, it benefits from a wide community and offers its Application Programming Interface (API) in the Java, Scala, R and Python programming languages.

Apache Spark's dataframe API, a collection based on RDDs and optimized for structured data processing, is particularly adequate for combining several datasets. Still, our first implementation had impractical time and memory space requirements to generate the dataset. Indeed, it was querying the Historical Climate Data API in real-time with a cache mechanism. Collecting only the weather stations and hours necessary for our sample of negative examples resulted in bad performances. We got a performance increase by first building a Spark dataframe with all the Historical Climate Data for weather stations around Montreal and then merging the two datasets. We conducted a detailed analysis of our algorithm to improve its performances. We notably obtained a good performance increase by not keeping intermediate results of the road segment identification for accidents. As opposed to what we initially thought, recomputing these results was faster than writing and reading them in the cache. Finally, the identification of the road segment corresponding to accidents was very memory intensive, we modified this step to be executed by batches of one month. With these improvements and a few other implementation improvements including re-partitioning the data frame at key points in our algorithm, we managed to reduce the processing times to a reasonable time of a few hours.

We also used clusters from Compute Canada to take maximum advantage of the Apache Spark distributed nature for the generation of examples and the hyper-parameter tuning of our models. We started with the Cedar cluster provided by West Grid and we continued with the new Béluga cluster provided by Calcul Québec.

To facilitate tests and development, our pre-processing program saves intermediate results to disk in the Parquet format. During later execution of the algorithm, if the intermediate results exists on disk, they will be read instead of being recomputed. This made it possible to quickly test new features and different parameters by recomputing only the

required parts of the dataset.

2.4 Model Development

2.4.1 Implementation of Balanced Random Forest

The Balanced Random Forest algorithm was not available in Apache Spark. An implementation is available in the Python library imbalanced-learn[40] which implements many algorithms to deal with data imbalance using an API inspired by scikit-learn[40], but the size of our dataset made it impossible for us to use this library. Therefore, we implemented Balanced Random Forest in Apache Spark.

In the Apache Spark implementation of Random Forest, the bootstrap step is made before starting to grow any tree. For each sample, an array contains the number of times it will appear in each tree. When doing sampling with replacement, values in this array are sampled from a Poisson distribution. The parameter of the Poisson distribution corresponds to the sub-sampling rate hyper-parameter of the Random Forest, which specifies the size of the sample used for training each tree as a fraction of the total size of the dataset. Indeed, if for example we want each tree to use a sample of the same size as the whole dataset, the sub-sampling ratio will be set to 1.0, which is indeed the average number of times a given example will appear in a tree.

To implement Balanced Random Forest, we modified the parameter of the Poisson distribution to use the class weight multiplied by the sub-sampling ratio. Hence, a negative sample with a weight of, say, 0.25 has 4 times less chance to be chosen to appear in a given tree. This implementation has the advantage that it did not require a big code change and is easy to test. However, it also has the drawback that users probably expect linearly correlated weights to be equivalent, which is not the case in our implementation since multiplying all the weights by n is like multiplying the sub-sampling ratio by n.

To be compatible with other possible use cases, the weights are actually applied per samples and not per class. This is a choice made by Apache Spark developers that we respected. To support sample weights, we create a new Poisson distribution for each sample. To make sure the random number generator is not reseeded for each sample, we use the same underlying random number generator for all Poisson distributions, this also helps reducing the cost of creating a new Poisson distribution object. Like with other estimators accepting weights, our Balanced Random Forest implementation reads weights from a weight column in the samples data frame. We adapted the Python wrapper of the Random Forest classifier to accept and forward weights to the algorithm in Scala.

2.4.2 Feature Engineering

For each example, we created three types of features: weather features, features from the road segment, and features from the date and time.

For weather features, we used data from the Historical Climate Dataset (see Section 2.3.1). To estimate the weather information at the location of the road segment, we used the mean of the weather information from all the surrounding weather stations at the date and hour of the example, weighted by the inverse squared distance between the station and the road segment. We initially used the inverse of the distance, but we obtained a small performance improvement when squaring the inverse of the distance. We tried higher exponents, but the results were not as good. We used all the continuous weather information provided by the Historical Climate Dataset. In addition, we created a feature to use the observations of atmospheric phenomenon provided by the dataset. To create this feature, we first created a binary variable set to 1 if the following phenomena are observed during the hour at a given station: freezing rain, freezing drizzle, snow, snow grains, ice crystals, ice pellets, ice pellet showers, snow showers, snow pellets, ice fog, blowing snow, freezing fog. We selected

these phenomena because they are likely to increase the risk of accidents. Then we computed the exponential moving average of this binary variable over time for each station in order to model the fact that these phenomena have an impact after they stop being observed and a greater impact when they are observed for a longer period of time. We used the same method as for other weather information to get a value for a given GPS position from the values of the weather stations.

For the features from the road segments, we were restricted by the limited metadata provided on the road segments. From the shape of the road segment, we computed the length of the road segment, and from the name of the street, we identified the type of road (highway, street, boulevard, etc.). In addition, road segments are classified into three different levels in the dataset depending on their importance in the road network: we created a categorical feature from this information. For these two categorical features, we encoded them as suggested in The Elements of Statistical Learning [34] in Section 9.2.4. Indeed, instead of using one-hot encoding which would create an exponential number of possible splits, we indexed the categorical variable ordered by the proportion of the examples belonging to the given category, which are positive samples. This encoding guarantees optimal splits on these categorical variables. Lastly, we added a feature giving the number of accidents that occurred previously on this road segment.

For the date features, we took the day of the year, the hour of the day, and the day of the week. We decided to make the features “day of the year" and “hour of the day" cyclic. Cyclic features are used when the extreme values of a variable have a similar meaning. For example, the value 23 and 0 for the variable hour of the day have a close meaning because there is only one hour difference between these two values. Cyclical encoding allows this fact to be expressed. With cyclical encoding, we compute two features, the first one is the cosine of the original feature scaled between 0 and 2π, and the second one is the sine of the original feature scaled between 0 and 2π. In addition to these basic date features, we

computed an approximation of the solar elevation using the hour of the day, the day of the year and the GPS coordinates. The solar elevation is the angle between the horizon and the sun. Note that it is of interest, because it is linked to the luminosity which is relevant for road accident prediction.

2.4.3 Identifying the most Important Features

Random Forest measures feature importance by computing the total decrease in impurity of all splits that use the feature, weighted by the number of samples. This feature importance measure is not perfect for interpretability since it is biased toward non-correlated variables, but it helps selecting the most useful features for the prediction. Random Forest usually performs better when irrelevant features are removed. Therefore, we removed the features wind direction, wind speed, dew point temperature, wind chill, hmdx index and day of month which had a much lower feature importance. This improved the performances of the model.

2.4.4 Hyper-Parameter Tuning

To determine the optimized hyper parameters, we first performed automatic hyper-parameter tuning by performing a grid search with cross-validation. Because the processing times on the whole dataset would have been too high, we took a small sample of the dataset. Still, we could not test many parameter combinations using this method.

Once we got a first result with grid search we continued manually by following a plan, do, check, adjust method. We plotted the precision-recall and ROC curves on the test and training set to understand how the performances of our model could be improved. These curves are obtained by computing the precision, the recall and the false positive rate metrics when varying the threshold used to classify an example as positive. Most classification algorithms provide a measure of the confidence with which an example belongs to a class.

We can reduce the threshold on the confidence beyond which we classify the example as positive in order to obtain a higher recall but a lower precision and a higher false positive rate. In order to obtain a general measure of the performances of a classifier at all thresholds, we can use the area under the ROC curve. The area under the Precision-Recall curve, however, should not be used [25].

Interestingly, despite using many trees, our Random Forest classifiers tended to over-fit very quickly as soon as the maximum depth parameter went above 18. We eventually used only 100 trees, because adding more trees did not increase performances. We have not tried more than 200 trees, maybe many more trees would have been necessary to increase the maximum depth without over-fitting, but then the memory requirement would become unreasonable. Our final parametrization used a total of 550 gigabytes of memory per training of the Balanced Random Forest model on the cluster.

2.5 Results

2.5.1 Balanced Random Forest Performances

To test our implementation of Balanced Random Forest (BRF) in Apache Spark, we performed an experiment on an imbalanced dataset provided by the imbalanced-learn library. We chose to use the mammography dataset[61] which is a small dataset with 11,183 instances and 6 features. It has an imbalance ratio of 42, i.e., there are 42 times more negative samples than positive samples. We compared the performances obtained with the implementation of BRF in the library imbalanced-learn with those obtained with our implementation of BRF in Apache Spark. We also compared these performances with the performances obtained with both implementations of the classical Random Forest algorithm. Results are summarized in Table 1. We observe that we obtain similar results with both implementations of BRF.

Table 1: Comparison of our BRF implementation with imbalanced-learn

	Area under ROC
imbalanced-learn RF	0.932
Spark RF	0.951
imbalanced-learn BRF	0.956
Spark BRF	0.960

Figure 1 shows the precision-recall curves obtained with both implementations of the Balanced Random Forest (BRF) and Random Forest (RF) algorithms on the mammography dataset. We can see that, with a low recall, BRF implementations perform worse, and with a high recall, all the models have similar performances except the Random Forest model from Apache Spark which has a lower precision.

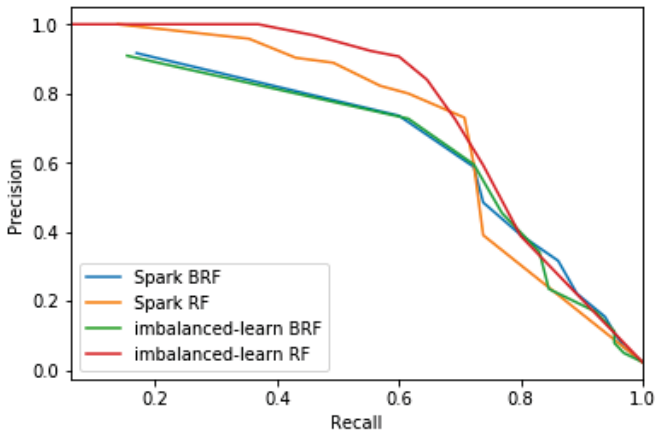

Figure 1: Comparison of implementations: Precision-recall curves

Figure 2 shows the Receiver operating characteristic (ROC) curves obtained with both implementations of the Balanced Random Forest (BRF) and Random Forest (RF) algorithms.

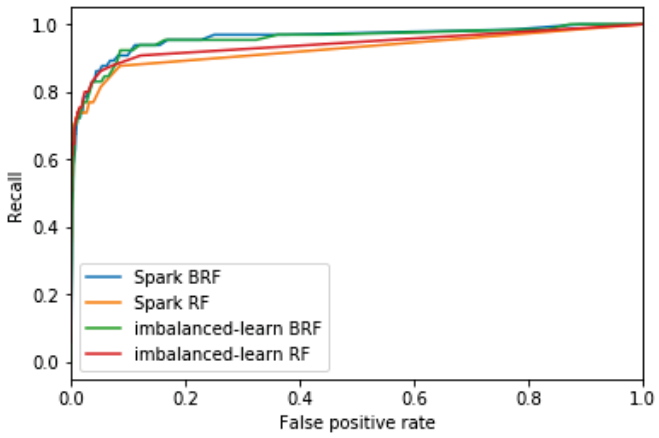

Figure 2: Comparison of implementations: ROC curves

2.5.2 Vehicle Collision Prediction

Results were obtained by training the algorithms on the whole dataset of positive samples and with a sub-sample of 0.1% of the 2 billion possible negative examples. This corresponds to a total of 2.3 million examples with a data imbalance reduced to a factor of 17. To evaluate our models, we used a test set containing the last two years of our dataset. The model was trained on the 4 previous years and used only data from these years. For instance, the “count_accident" feature contains only the count of accidents occurring from 2012 to 2016 on the road segment. In addition to the three models built using tree-based machine learning algorithms, we created a simple baseline model. This model is very basic in the sense that it uses only the count of accidents of the road segment. The probability of accidents given by this model for an example whose road segment has a count of accidents of n, is the percentage of positive examples among the examples with a count of accidents higher than n.

Table 2 presents the results obtained on the test set with the classical Random Forest algorithm with further under-sampling (RF), with the Balanced Random Forest algorithm (BRF), with the XGBoost algorithm (XGB), and with the baseline model (base). The values of the hyper-parameters we used and more details about the results are available on the

Github repository of the project.

Table 2: Result Summary

	BRF	RF	XGB	base
Area under the ROC curve	0.916	0.918	0.909	0.874

As we can see, the three machine learning models obtain similar performances and perform much better than the baseline model. The XGBoost model has slightly worse performances than the two others.

Figure 3 shows the precision-recall curves of the three models.

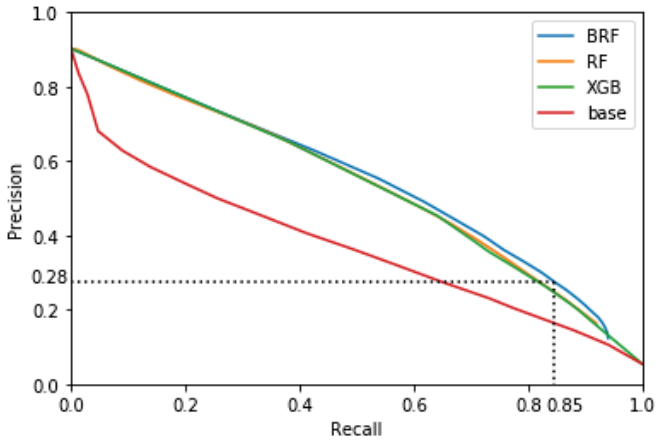

Figure 3: Vehicle Collision Prediction: Precision-recall curves

Figure 4 shows the Receiver operating characteristic (ROC) curves of the three models.

Figure 5 shows the precision and the recall as a function of the threshold values for BRF and RF algorithms. It shows that despite BRF and RF having similar results on the PR and ROC curves, they have different behaviors. For an identical threshold value, BRF has a higher recall but a lower precision than RF.

As we can see, the Balanced Random Forest model surprisingly does not perform better than the other models. It achieves a recall of 85% with a precision of 28%, and a false positive rate (FPR) of 13% on the test set.

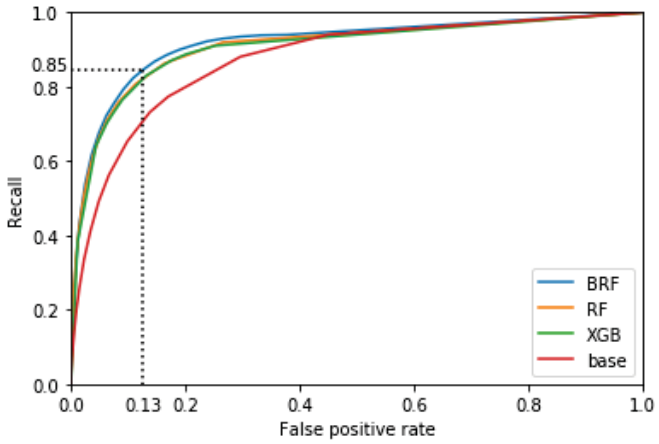

Figure 4: Vehicle Collision Prediction: ROC curves

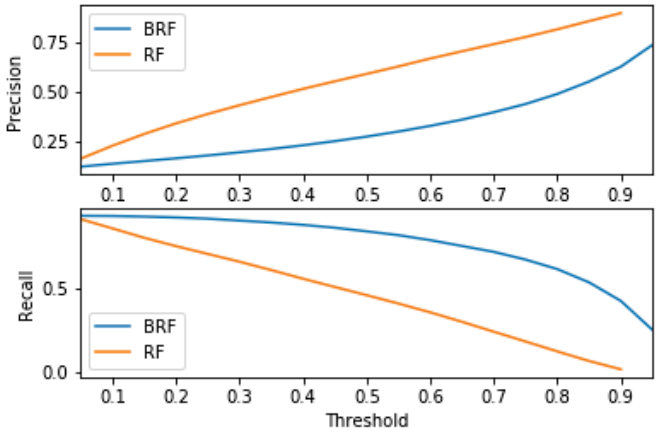

Figure 5: Vehicle Collision Prediction: Precision and Recall as a Function of the Threshold Values

2.5.3 Vehicle Collision Feature Importance

With a feature importance of 67%, the number of accidents which occurred on the road segment during the previous years is clearly the most useful feature. This shows that accidents are concentrated on specific roads. Figure 6 presents the importance of the other features as reported by the Balanced Random Forest algorithm. As we can see, the next most important feature is the temperature. Then, the day of the year, the cosine of the hour of the day, which separates day from night, and the visibility follow. The solar elevation and the humidity are the following features of importance. The remaining features have almost the same importance, except the street type which is significantly less important.

We believe that the road features like the street length, the street level and the street type have a lower importance because the accident count already provides a lot of information on the dangerousness of a road segment. Surprisingly, the risky weather feature is one of the least important ones. This suggests that our definition of risky weather may need to be revisited.

As compared to the count of accidents, the other features seem to have almost no importance, however the performance of the model decreases significantly if we remove one of them.

2.6 Discussion

With areas under the ROC curve of more than 90%, the performances of our models are good. However, they mostly rely on the count of previous accidents on the road segment as we can see from the feature importance of the accident count feature and the performance of the base model. This is not an issue for accident prediction, but it does not help to understand why these roads are particularly dangerous. We believe that this feature is even more useful because we do not have information about the average traffic volume for each

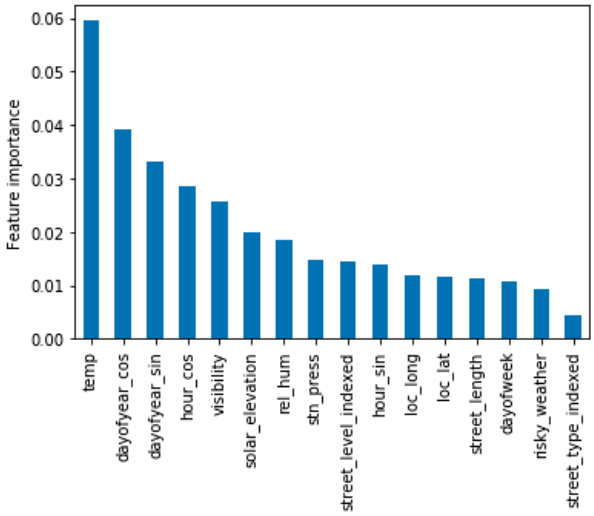

Figure 6: Feature importance computed by the Balanced Random Forest excluding the accident count feature.

road. Therefore, this feature does not only inform the machine learning algorithm about the dangerousness of a road segment but also indirectly about the number of vehicles using this road. Nonetheless, the performance of our models does not only rely on this feature. As we can see from the curves, the performances of our models are significantly better than those of the base model that exclusively relies on the count of accidents.

2.6.1 Test of our Implementation of BRF on the Mammography Dataset

As expected, we obtained similar results to the imbalanced-learn library with our implementation of the BRF algorithm. The precision-recall curve shows that the BRF algorithm had a better precision with high recall values, but a much lower precision with low recall values. For medical diagnosis and road vehicle collision prediction, we usually prefer to have a higher recall with a lower precision, so BRF is more suitable for these use cases.

2.6.2 Comparison of the Different Models for Road Vehicle Collisions Prediction

For the road vehicle collision prediction, the Balance Random Forest algorithm obtained slightly better results than the classical Random Forest algorithm. However, the gain in prediction performance is very small. We believe this is caused by the fact that negative examples are not so different from each other and the information they contain is well captured by a single random sub-sample. We observe that the BRF algorithm achieved better performances than Random Forest with high recall values. With lower recall values, both Random Forest algorithms had similar performances. The XGBoost algorithm obtained worse results than the two other algorithms. However, it is still interesting because it was much faster to train than Random Forest algorithms. This made the hyper-parameter tuning of the XGBoost algorithm easier and much faster.

2.6.3 Real-world Performances of our Road Vehicle Collision Prediction Model

As stated previously, the accuracy measure is not a good metric for road accident prediction. Indeed, since most examples belong to the negative class, the model which obtains the best accuracy is usually the one with the lowest false positive rate. But for rare event prediction, we usually want a model with a high recall even if it implies a higher false positive rate. This is especially true in accident prediction, because false positives can correspond to high-risk situations that we probably want to detect too. For these reasons, we decided not to use the accuracy measure. Instead we used the precision-recall curve to compare the performances of our models. However, we should be careful when using the precision measure on a dataset using a sample of the possible negative examples like it is usually the case in accident prediction. Indeed, the precision computed on the test set does

not correspond to the precision we would obtain in production. If the sample of negative examples is representative of the population in production, the model will achieve the same false positive rate. Because we used a sample of the possible negative examples but all the positive examples in the test set, there will be more cases of false positive in production for the same number of positives. As a consequence, the precision will be much lower.

Since we know the proportion of positive examples in the real world, if we assume that the sample of negative examples is representative of the population in production, we can provide an estimation of the precision that the model could achieve. There are on average $22,414$ collisions each year and during a year there are a total of $386,412,360$ combinations of hour and road segments. Therefore, in the real-world approximately 0.0058% of examples are positive. With a recall of 85%, approximately 0.00493% of examples are true positives and 0.00087% are false negatives. With a false positive rate of 13%, approximately 12.99925% of examples are false positives and 86.99495% are true negatives. Therefore, with the real world distribution, our model would likely obtain a precision of 0.04%. If the goal of our model was to actually predict accidents, this would not be a satisfying precision, but the real goal of accident prediction is to identify when and where the risk of accidents is significantly higher than average in order to take measures. With this precision, the probability of a collision to occur is 6 times higher than average for examples detected as positive. By varying the threshold used by the model, we can choose when to take actions.

2.6.4 Reproducibility of the study

The results from this study can be reproduced using the Github repository of the project. The 'readme' file provides more information on how to create training examples from the datasets and how to train the models. All the figures can be reproduced with the Jupyter notebooks available on the same repository.

The National Road Network and the Historical Climate datasets used in this study are open datasets from the government of Canada. One can potentially reproduce the study for any other Canadian city as long as the the city provides open data on vehicle collisions including the date, time and localization of such collisions in sufficient amount. For example, the city of Toronto seems to be a good candidate with 11 years of vehicle collisions open data available through the “Automobile" dataset provided by the Toronto Police Service. The latter dataset contains the date, time and localization of the accidents. National road network information and historical climate information tends to be easily found for many countries which would allow this study to also be reproducible in other countries. For example historical climate information for the United States can be found in the U.S. Historical Climatology Network dataset and road network information seems to be available in the USGS National Transportation Dataset.

2.6.5 Future Work

We believe that a better performance could be reached by adding more features from other datasets. For the city of Montreal, we identified two particularly interesting datasets: a dataset with the location and dates of construction work on roads, and a dataset with the population density. In addition, Transport Québec gives access to cameras monitoring the main roads of Montreal. The videos from these cameras could be useful to get an estimation of the traffic in the roads of the island. These datasets could be used to improve prediction performances. However, this type of dataset might not be available for other geographical areas. The current model use datasets that can easily be made available for most cities.

The most important feature is the number of accidents which happened during the previous year. While this feature helps a lot to reach useful prediction performances, it does not help in understanding the characteristics of a road segment which makes it dangerous. A human analysis of these particularly risky road segments could detect patterns that could

help to take measure to reduce the number of accidents in Montreal. This can also be useful to improve our current accident prediction model, if the detected patterns can be used by merging other datasets.

Lastly, it would be interesting to analyze why BRF did not perform better for this problem in order to understand under which conditions it helps to deal with data imbalance.

2.7 Conclusions

In this study, we conducted an analysis of road vehicle collisions in the city of Montreal using open data provided by Montreal city and the Government of Canada. Using three different datasets, we built road vehicle collision prediction models using tree-based algorithms. Our best model can predict 85% of road accidents in the area of Montreal with a false positive rate of 13%. Our models predict the occurrence of a collision at high space resolution and hourly precision. In other words, it means our models can be used to identify the most dangerous road segments every hour, in order to take actions to reduce the risk of accidents. Moreover, we believe that our work can easily be reproduced for other cities under the condition that similar datasets are available. One can freely use our source code on Github for reference. Finally, our study shows that open data initiatives are useful to society because they make it possible to study critical issues like road accidents.

Chapter 3

Can we Estimate Truck Accident Risk from Telemetric Data using Machine Learning ?

Road accidents have a high societal cost that could be reduced through improved risk predictions using machine learning. This study investigates whether telemetric data collected on long-distance trucks can be used to predict the risk of accident associated with a driver. We use a dataset provided by a truck transportation company containing the driving data of 1,141 drivers for 18 months. We evaluate two different machine learning approaches to perform this task. In the first approach, features are extracted from the time series data using the FRESH algorithm and then used to estimate the risk using Random Forests. In the second approach, we use a convolutional neural network to directly estimate the risk from the time series data. We find that neither approaches is able to successfully estimate the risk of accident on this dataset, in spite of many methodological attempts. We discuss the difficulties of using telemetric data for the estimation of the risk of accident that could explain this negative result.

This chapter will be submitted shortly to IEEE Transactions on Intelligent Transportation Systems.

3.1 Introduction

Despite improvements in road safety, road accidents remain an important issue worldwide: they lead to an estimated 1.35 million deaths and more than 20 million injuries every year, and are the leading cause of death for people aged between 5 and 29 [51, 26]. Road accidents also represent a high economic cost for society. In Canada, the yearly economic cost of transport-related injuries is estimated to US$3.2 billions [30].

Road accidents are an important issue for truck transportation companies. Each accident can cause driver injuries, truck repair costs and the loss of transported goods. The US Federal Motor Carrier Safety Administration (FMCSA) estimated at US$148,279 the average cost of a truck crash for society [22]. To minimize road accidents, most truck transportation companies analyze accidents to understand their causes and how they might be prevented. Some companies also offer regular training to their drivers to promote safe driving. According to the FMCSA, 5.5% of fatal truck crashes are caused by driver fatigue and could have been prevented [23].

In the United States and Canada, it is now mandatory for motor carriers to equip their trucks with electronic logging devices (ELD) directly connected to the vehicle to track service hours [50, 31]. This is an opportunity for transportation companies to go beyond the compliance requirements and install telemetric systems to collect a variety of sensor data from the vehicle. Many such telemetric solutions are available on the market to improve truck fleet management by providing real-time information to fleet managers [38]. Telemetric systems produce huge amounts of data, generated by an ever-increasing number of sensors on the vehicle.

The availability of big amounts of telemetric data generated by vehicles is a great opportunity to try to predict accidents by characterizing dangerous driving behaviour. Indeed, it is likely that the style of driving greatly influences the risk of accidents. In this study, we design a machine learning model using such telemetric data to estimate the risk of accident associated with a driver.

Telemetric data generated by vehicles is in the form of time series. During driving, vehicle sensors record various parameters at regular intervals and store them in the telemetric system. We will therefore design machine learning models which can provide a measure of the risk of accidents of a given driver by looking at times series containing the evolution of various parameters during its driving. If we define the risk of accidents as the probability that this driver has an accident, then estimating the risk of accident is equivalent to classifying examples as leading to an accident or not. Therefore, the problem is a time series classification one.

Road accident prediction has been studied, but never using this type of data. Most studies predict the risk of accidents at different points in time and space using characteristics of the road network and weather information. Instead, we are interested in predicting the risk of accidents for a given driver based on information about their driving. Such a model could help truck transportation companies identify drivers with riskier driving styles, and offer them additional safe driving trainings. It could also be useful to insurance companies.

The rest of this paper is organized as follows: Section 3.2 presents the related work on road accident prediction and on time series classification, Section 3.3 and Section 3.4 present our datasets and model creation methods, Section 3.5 presents experiments and results, and Section 3.6 discusses these results. Conclusions are drawn in the last section.

3.2 Related work

3.2.1 Road Accident Prediction

Many studies consider road accident prediction and aim at predicting the risk of an accident at a given place and time. These studies would for example predict which segments of a road are most dangerous[8], or what times and areas of a city are most dangerous[11]. They usually use information about the road such as the average daily traffic or the road curvature, as well as weather information such as the temperature or the precipitation.

Early work on road accident prediction used classical statistical modelling, usually variants of Poisson Regression. In 2005, Chang[8] compared an artificial neural network with a negative binomial regression for the prediction of the number of accidents on road segments of a Taiwanese freeway: it was the first work to show that machine learning methods could achieve better performances than classical statistical modelling for road accident prediction. Later studies performed road accident prediction with various machine learning algorithms, usually only focusing on a few roads [9, 42, 57]. More recently, other studies performed road accident prediction at a larger scale covering larger areas or predicting at a higher-resolution [11, 49, 62, 37]. These studies showed that weather and road characteristics influence the risk of accident, and that it is possible to successfully identify places and times where accident are much more likely to happen. Instead, our goal is to identify the accident risk associated with a particular truck driver, regardless of location or weather conditions.

3.2.2 Time Series Classification

The literature on time series classification is very diverse in terms of methods and models. We identify four broad classes of methods: feature-based, model-based, distance-based[1], and representation based[20]. Feature-based methods first derive features from the time

series data and then apply classical classification algorithms. The model-based approach is a generative approach that trains, for each class, a generative model learning the characteristics of the class. To predict the class of a new example, each model is asked how likely it is that this example belong to its class, and the predicted class is the class with the highest probability. Distance-based methods define a relevant distance metric between two time series and then use a k-nearest neighbor classifier (k-NN) or a support vector machine (SVM). Finally, representation-based methods use deep neural networks to learn a representation of time series and classify accordingly.

The performance of different methods highly depends on the type of time series and problems. The distance-based approach and the use of elastic distance measures were historically the most popular approach [4]. Dynamic time warping (DTW) is a commonly used distance measure. Many variants have been proposed but Lines and Bagnall [44] have shown that none of them is significantly better than DTW. In 2016, Bagnall et al. [3] compared the performances of different time series classification methods from the feature-based, model-based and distance-based approaches on the datasets of the UCR time series classification repository[18]. The best-performing algorithm was COTE [4], an ensemble of classifiers applied on various time series transformations. COTE combines 11 distance-based classifiers and 24 feature-based ones. The same year, COTE was improved with HIVE-COTE[45] which introduces two additional sets of classifiers and a hierarchical voting system improving the aggregation of the different classifier results. An important limitation of both COTE and HIVE-COTE is their very high computational requirements as they combine many classifiers and complex transformations with complexities as high as $O(n^2t^4)$ with n the number of time series and t their length. This limitation makes it impractical to use these algorithms with big datasets or long time series.

When using feature-based or distance-based methods, it is hard to know which distance or which features to use without expertise on the data used. In 2016, Christ et al. [15]

introduced an algorithm called FRESH (FeatuRe Extraction based on Scalable Hypothesis test) that automatically selects relevant time series features for binary classification. The algorithm has three main steps. First, it computes many possible features from the time series, simple features such as the mean, the standard deviation or the kurtosis, but also more advanced features such as the number of peaks or the spectral centroid. Then, for each feature, it uses a statistical test to check if the feature is relevant to predict the class, and finally selects the best features using the Benjamini-Yekutieli procedure[5]. The resulting features can then be used with any classical machine learning algorithm. The authors evaluate the performances of this method when combined with an Adaboost classifier on the UCR time series classification repository[18]. It achieves results comparable to the DTW algorithm, with a lower computational cost as FRESH scales linearly with the number of samples and the length of the time series. In 2019, Fawaz et al.[20] evaluated the performances of representation-based methods and compared the performance of several deep neural networks. They found that a ResNet deep neural network competes with HIVE-COTE while being much more computationally efficient. More recently, Fawaz et al.[21] introduced a new deep neural network architecture for time series classification slightly outperforming HIVE-COTE on the UCR time series classification repository with a win/draw/loss of 40/6/39. This new architecture named InceptionTime was inspired by the Inception-v4 architecture [56] used in computer vision.

In summary, time series classification made significant progress in recent years. HIVE-COTE offers state of the art performances but has impractical computational cost. For big datasets, deep neural networks or the FRESH algorithm coupled with classical machine learning seems to be the two most promising approaches. We will use both approaches to build our models.

3.3 Datasets

The datasets used in this study were collected by Groupe Robert Inc, a transportation company based in Quebec, Canada. For many years, Groupe Robert Inc. has been monitoring road accidents and infractions involving their truck drivers to better understand how to reduce the number of accidents. In 2017, it equipped its truck fleet with a telemetric system collecting most of the data generated by vehicle sensors during driving.

We used two datasets provided by the company: (1) the data from the sensors of the vehicle collected using the telemetric system onboard the trucks, and (2) the list of accidents involving drivers of the company, extracted from the records of the company. These datasets contain data collected for 18 months between February 2018 and June 2019.

The telemetric system records the values measured by the vehicle sensors whenever the engine is on. Different sensors are recorded at different time intervals, every half a second, every second, every 10 seconds or every minute. The values measured by the sensors are collected on the CAN BUS of the vehicle using the Society of Automotive Engineers J1939 communication protocol. This protocol defines identifiers for each sensor on the vehicle (see Table 3). We have not used 24 other recorded parameters which we identified as not relevant for our study in agreement with the domain experts at Groupe Robert Inc.

The truck fleet of the company is not homogeneous, it is composed of different types of trucks used for different transportation needs. The company identifies 3 different types of trucks: long-distance trucks, short-distance trucks, and specialized trucks like container and bulk trucks. These trucks are not equipped with the same sensors and follow different driving patterns. In our first model, we will focus on long-distance trucks, as it is likely that the other classes will require different models.

The company keeps track of all accidents involving their trucks, amounting to 1,434 accidents during the study period. For each accident, the date of the accident, the identifier of the driver, and the type of accident are recorded. Table 4 shows the 30 types of accidents

Table 3: Some of the parameters collected

Sensor identifier	Description
Acc_Lat	Acceleration on the lateral axis
Acc_Long	Acceleration on the longitudinal axis
Acc_Long_WBVS	Acceleration on the longitudinal axis as measured on the wheels
Acc_Vert	Acceleration on the vertical axis
AccelPedalPos1	Use of the acceleration pedal
ActualEngPercentTorque	Engine torque in percentage
ActualEnginePower	Engine power
ActualEngineTorque	Engine torque
ActualRetarderPercentTorque	Retarder torque in percentage
AmbientAirTemp	Ambient air temperature
BarometricPress	Barometric pressure
BrakeSwitch	Status of brake switch
CruiseCtrlActive	Status of cruise control
EcoMode	Status of economy mode
EngCoolantTemp	Temperature of engine coolant
EngFuelRate	Fuel rate
EngReferenceTorque	Engine reference torque
EngSpeed	Engine rotation speed
EngTurboBoostPress_PSI	Engine turbocharger boost pressure
EstEngPrsticLossesPercentTorque	Estimated torque loss due to engine parasitics
NominalFrictionPercentTorque	Nominal friction torque in percentage
Top_Gear_State	Whether the top gear is used
WheelBasedVehicleSpeed	Vehicle speed as measured on the wheels
gps_Altitude	GPS altitude
gps_Lat	GPS latitude
gps_Long	GPS longitude
gps_Speed	Vehicle speed as reported by the GPS

that were identified. The four most frequent types of accidents are types 1, 2, 3 and 4, representing 61% of accidents. They correspond to non-severe accidents occurring mostly during maneuvers.

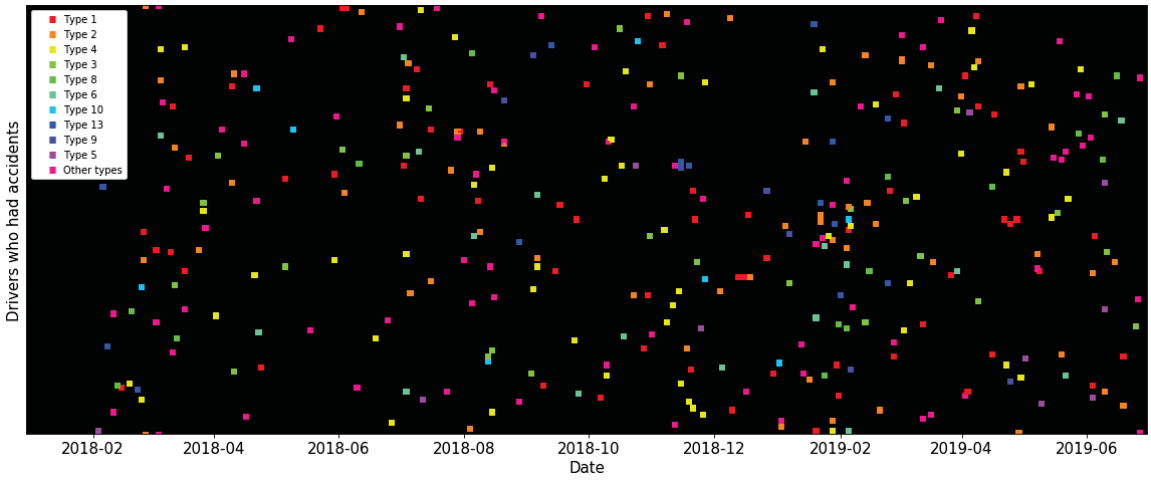

Figure 7: Visualization of the distribution of accidents in time

Figure 7 presents the distribution of accidents in time and across drivers. Each row corresponds to a different driver and the x-axis represents time. Each colored square correspond to an accident and the color of the square correspond to the type of accident. Only the 48% of drivers who had an accident during the study period are included in this visualization. We observe that most drivers who had an accident had more than one during the study period.

3.4 Method

3.4.1 Data preprocessing

The data obtained from Groupe Robert Inc required formatting to be usable for model training. The data was initially in the form of 14 million files with each file containing the data collected on one truck during a driving period lasting between less than a minute to an hour.

Table 4: Types of accident

ID	Proportion	Description
1	26%	Accident while driving backwards
2	17.6%	Hit a stationary object (except wall)
3	6.6%	Accident while changing dock
4	11%	Hit a stationary vehicle
5	2.2%	Hit an animal
6	4.7%	Rear collision
7	1.5%	Damaged equipment during loading
8	4.9%	Miscellaneous
9	2.3%	Hit a cable
10	3.5%	Rubbing
11	1%	Accident while turning right at intersection because a third party was overtaking on the right
12	0.2%	Accident while going straight through the intersection
13	3%	Loss of control
14	2%	Accident or fined because the truck cut off
15	1.9%	Accident caused by trailer not properly coupled with truck
16	0.9%	Truck stuck (in snow for example), towing necessary
17	2.1%	Hit a wall or building
18	0.7%	Mechanical Breakdown
19	1.7%	Fined because of leaking truck
20	1.2%	Improper maneuvering in tight turns
21	0.3%	Fined because of improper snow clearance of the truck (for example ice remaining on the truck roof)
22	0.5%	Accident caused by vehicle wheel ignition
23	0.7%	Hit a bridge
24	1.8%	Equipment damaged during unloading
25	0.6%	Cargo
26	0.1%	Vehicle wheel loss
27	0.3%	Accident while turning left at intersection because a third party was overtaking on the left
28	0.1%	Truck cargo theft
29	0.7%	Truck cargo fell out of the truck
30	0.1%	Equipment damaged without reported accident

These files were in a proprietary format used by the telemetric system. Two MS Windows utilities were provided to convert a file from a proprietary format to another proprietary format and then to the CSV format. In addition, a separate CSV file identified which truck each driver was driving at different times. We used custom Python scripts and a virtual RAM drive to efficiently convert each file to the Apache Parquet format, using the provided utility to read files. The Apache Parquet format is a format from the Apache Hadoop ecosystem providing efficient data compression. This conversion was a data-intensive process that took several days. Once all files were converted to the Apache Parquet format, we identified the driver corresponding to each file and merged files corresponding to contiguous driving periods by the same driver on the same truck. As a result, we obtained 3.2 million Parquet files representing 890 GB of data.

We were informed that some of the accelerometer sensors might not be properly configured, and that the reported acceleration on the lateral axis, on the longitudinal axis and on the vertical axis might be permuted and in the wrong direction. We attempted to fix these issues by permuting and changing the sign of these parameters so that the acceleration on the longitudinal axis is positively correlated to the acceleration on the longitudinal axis as measured on the wheels for each truck and each month. This correction is not perfect since the accelerometers have been reconfigured at different dates for each truck and not necessarily at the beginning of the month.

3.4.2 Instance creation

Trucks make frequent stops which results in gaps in the time series. To alleviate this problem, we extracted non-overlapping windows of continuous driving from the raw data (Figure 8). After a few trials, we chose a window size of one hour, meaning that 3 windows of data could be extracted from a trip with a duration between 3 and 4 hours. A smaller window size would discard low-frequency patterns, while a too long window size would

make it necessary to discard more data since driving periods shorter than the window size cannot be used. Since one hour of driving might not be enough to access the driving style of a driver, we aggregated 60 sequential but not necessarily contiguous windows to form each example. Therefore, our machine learning models look at 60 hours of driving to estimate the risk of accident.

When performing statistical learning, we need to assume that examples are independent and identically distributed. In this study, we use the data from one driver to generate several examples, which means that examples are not all independent from each other. There is probably some correlation between examples corresponding to the same driver. This could affect learning, but it allows us to extract a reasonable number of examples from the limited data available. In the next subsection, we will show how we defined our test sets carefully so that they remain valid despite examples not being independent.

As presented in Section 3.3, a total of 51 parameters are recorded during driving, 24 of these parameters were identified as non-relevant by the domain experts from the company. We experimented with using various subsets of the 27 parameters left and found that the best results on the validation set were obtained when using only 6 parameters: the acceleration in the three dimensions, the position of the accelerator pedal, the engine torque and the retarder torque. The acceleration parameters and the engine torque were recorded every half a second while the other parameters were recorded every second. We downsampled the acceleration parameters and the engine torque to obtain the same sampling frequency for all parameters and reduce the computational requirement of further processings. Figure 9 presents an example of the data corresponding to a one hour window.

3.4.3 Labeling

Our goal was to obtain a model to estimate the accident risk. To train such a model in a supervised way, we needed for each example a “ground truth" value of the risk of accidents.

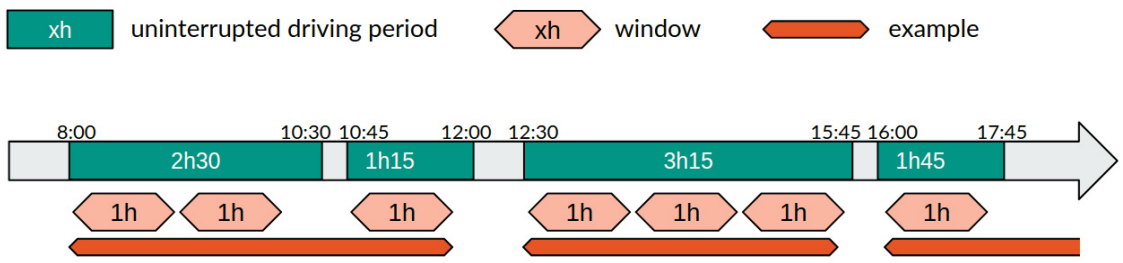

Figure 8: Illustration of example creation from raw data with 1-hour windows and 3 windows per example.

We used our second dataset, containing the list of accidents, to evaluate the accident risk associated with each example. By defining the risk of accident as the probability of having an accident, a model estimating the accident risk can be considered as a binary classification model. Driving data generated by a driver who had an accident belongs to the positive class, while data generated by a driver who did not belongs to the negative class. By training the model to classify driving data in this way, we obtained a model estimating the probability that new driving data belongs to the positive class, this probability is the accident risk according to our definition.

More precisely, we considered as positive the examples generated by a driver who had an accident in the year following the date of the example. We decided not to consider as positive the examples that followed an accident because we assumed that drivers might adjust their driving after they have an accident. We used a duration of one year because accidents are rare, and an incautious driving will not result in an accident right away. We experimented with shorter durations ranging from a week to a year.

As explained in Section 3.3, there are different types of accidents in the dataset. It is likely that some of these accident types are not related to the driving data, for example drivers are probably not responsible for accidents of type 5 when the truck hit an animal. Therefore, we decided to ignore some accident types, based on how well they are predicted on the validation set. We only used the accident types 1, 2, 7, 8, 9, 11, 15, 16, 17, 22, 23.

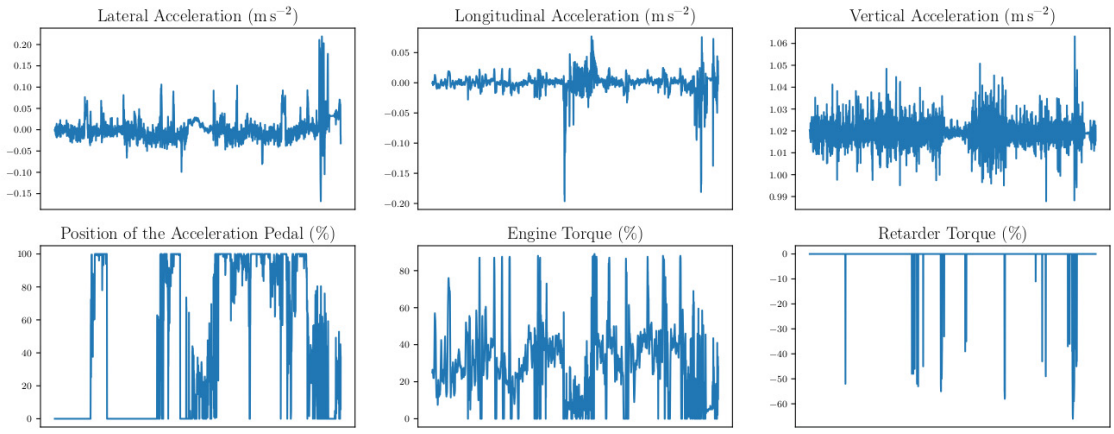

Figure 9: Window of one hour of driving data

3.4.4 Creation of training and test sets

As mentioned previously, there is a risk of shared information between the training and test sets due to the way we create examples. This has consequences on how to correctly split the examples into a training set and a testing set for performance evaluation.

If we simply take a random sample of examples to create the test set, we will be measuring the ability of the model to recognise drivers, and not its ability to measure the accident risk of a new driver. Indeed, examples from the same drivers would be present in both the training and the testing set. In addition, most of the examples generated from the data of one driver have the same label: if the driver never had an accident during the study period, then all the examples will be negative; if the driver had an accident toward the end of the study period, then almost all their examples will be positive. Therefore, the model could correctly classify an example simply by recognizing the driver and retrieving from the training data whether this driver had an accident after the example occurred.

We split the training and test sets by driver rather than by example, to make sure that we evaluate the ability of the model to estimate the accident risk on unknown drivers. In

addition, to ensure that the test set is a representative sample of the examples, we performed a stratified split: we ensured that the percentage of positive examples in the test set is approximately the same as the percentage of positive examples in the data. We used approximately 30% of the examples to create the test set.

We did not use the test set for the tuning of preprocessing and for model selection. Instead, we used validation sets created from the training data with the same procedure as for the test set, i.e., by making a stratified split by driver. We performed early tuning with one validation set containing 30% of the training examples. We noticed that reported performance metrics could significantly change depending on which random validation set was used, so we later used k-fold cross-validation to obtain a more stable estimation of performances. Like for the test set, we made sure that examples corresponding to one driver were either in the training set or the validation set and that the proportion of positive examples in the validation set was representative of the proportion of positive examples in the dataset.

3.4.5 Feature-based approach

We built a first model using a feature-based approach and the FRESH algorithm [15] for feature extraction and selection. Indeed, as discussed in Section 3.2, the FRESH algorithm seemed a promising approach for time-series classification when using big datasets.

We used the TSFRESH library [14] (version 0.14.1), a Python library implementing the FRESH algorithm to extract features from time series and select the most promising ones. The extraction of these features for all the examples was a long process that took several days. To speed up the process, we excluded features labeled by the library as having a high computational cost. A total of $4,488$ features were extracted for each example, and $1,728$ of them have been considered relevant by the FRESH algorithm. We used the Random Forest algorithm [7] to perform classification based on these features. Hyperparameters of

the Random Forest algorithm were tuned using 5-fold cross-validation.

3.4.6 Representation-based approach

As discussed in Section 3.2, deep neural networks have obtained state-of-the-art performances on some TSC datasets and offer a much lower computational cost than competing methods.

We started with the neural network architecture which obtained the best average performance in [20]: a ResNet neural network[36] adapted for TSC. This architecture is composed of 3 residual blocks followed by a global averaging pooling averaging feature maps over time and a final fully-connected layer. Each residual block is composed of 3 convolutional layers using batch normalization and a residual connection adding the input of the block to the pre-activation of the last layer. This residual connection is the main characteristic of this architecture and gave it its name which stands for Residual Network. Fawaz et al. provide an implementation of this neural network using TensorFlow, which we reimplemented in PyTorch [52] for convenience.

The original architecture takes as input matrices of dimension (C, L) with C the number of channels and L the the length of the time series. We adapted the neural network to be able to use tensors of dimension (N, C, L) with N the number of windows. As indicated in 3.4.2, we used $N = 60$ windows for each examples. We adapted the architecture by removing the last layer and applying the rest of the neural network to each window. We added a head combining extracted features. We initially used a few fully-connected layers to form the head, but later found that a global average over windows followed by one final fully-connected layer seemed to perform best.

Our initial adapted ResNet obtained very bad performances on the validation sets. We made a lot of changes to the neural network architecture and its training procedure to obtain better performances.

We quickly noticed that our model was subject to overfitting, indeed, while it obtained very good results on the training set, results on the validation sets were very bad. We therefore added spatial dropout [58] after each convolutional layer to regularize the model. Spatial dropout consists in randomly dropping out the activation of some feature maps during training. With convolutional layers, it is recommended to use spatial dropout instead of regular dropout, indeed neurons from the same feature map are usually correlated and randomly dropping neurons independently does not affect much the learning process. We used a high dropout rate for all layers in order to regularize our model as much as possible. We found using automatic hyperparameter tuning that a dropout rate of 57% seemed to perform best on the validation sets.

Even after adding heavy dropout to the neural network, and reducing the number of feature maps, the neural network was still overfitting the training data. In order to further reduce its capacity, we tried reducing its depth. We found that the neural network was performing best on the validation set with only one residual block. This is surprising because deeper networks trained for less epochs usually generalize better than shallower network trained for longer. With such a shallow network, one could wonder if the residual connection is still useful, after experimenting without we found it was indeed not useful. We also removed zero-padding which became no longer necessary.

To further reduce the capacity of the model, we tried making use of strided convolutions. By using a convolutional layer with a stride greater than 1, the following convolutional layer can achieve the same receptive field with a smaller kernel. We found better results when using a stride of 2 for the first two convolutional layers while adapting the kernel sizes accordingly.

We experimented with different activation functions. The Exponential Linear Unit (ELU)[17] seemed to perform best, so we replaced the ReLU activations initially used by ELU activations.

It can be challenging to find the right set of hyperparameters for which a neural network will learn successfully. The common practice is to start with the configuration of hyperparameters used by another study on a related problem. It was not possible in this study since to the best of our knowledge, there are no other studies making use of telemetric driving data for accident prediction. To help with the search of a good configuration of hyper-parameters, we made use of automatic hyper-parameter tuning. Thanks to the limited size of our dataset and of our model, it was possible to try many different configurations. The following hyperparameters were automatically tuned: the amount of weight decay, the dropout rate, the kernel size of the three convolutional layers and the number of feature maps. We found that the amount of weight decay did not seem to matter, this might be because the use of batch normalization changes the effect of weight decay [64]. For other hyperparameters, we obtained the following values: 57% for the dropout rate, 31, 8 and 4 for the kernel sizes of the first, second and third convolutional layers and 10 for the number of feature maps.

To train the neural network, we used the Adam optimization algorithm [39] with a small amount of weight decay. We used the corrected implementation of Adam with weight decay [46]. We used a batch size of 32. To find a good learning rate, we used the method presented in [55], and we obtained a learning rate of 1.1×10^{-1}. To determine for how many epochs to train the model, we used early stopping: we evaluated the performances of the model on the validation set after each epoch and stopped training when the performance did not improve for 3 epochs in a row. Finally, we used a focal loss [43] instead of the usual cross-entropy loss. This loss is designed to help with data imbalance and we found that it improved our results.

3.5 Experiments and Results

To measure the performance of our models, we used mainly the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve shows the evolution of the True Positive Rate (TPR) as a function of the False Positive Rate (FPR) when varying the threshold used by the model to classify examples. The TPR is the proportion of examples identified as positives among actual positives, and the FPR is the proportion of examples identified as positives among actual negatives. The area under the ROC curve corresponds to the probability that the model will rank a randomly chosen positive example higher than a randomly chosen negative one, so we believe it is appropriate to evaluate a risk estimation model.

As indicated in the previous section, for both approaches, we used k-fold cross-validation for model selection. We decided to report results on both the validation sets and the test set. For the validation results we report the average of the results of the different models obtained with different splits of the training data. To obtain the average ROC curves, we average the True Positive Rate for each False Positive Rate. For the test results, with the feature-based approach, we simply retrain a model using the whole training dataset before evaluating it on the test set. With the representation-based approach, the validation set is not only used for model selection but also for early-stopping, so we cannot retrain a single model using the whole training data. Instead, we report the average results on the test set of the models obtained with different splits. We cannot simply select the model with the best validation results among the models trained with different splits, because the performances of the model on the validation set do not reflect its performances on the test set.

With the feature-based approach, we obtained an average area under the ROC curve of 58% on the validation sets, which correspond to performances slightly better than those of a random classifier. But on the test set we obtained an area under the ROC curve of 43% only.

With the representation-based approach, we were able to obtain better results on the

validation sets with an average area under the ROC curve of 65%. On the test set however, results are the same as with the feature-based approach with an area under the ROC curve of 43%.

With both approaches, we noticed a high variation of performances measured using the validation set across the different splits of the k-fold cross-validation. The standard deviation of the area under the ROC curves was 9% with the feature-based approach and 7% with the representation-based one.

Figure 10 presents the ROC curves on the test set and on the validation sets obtained with both models.

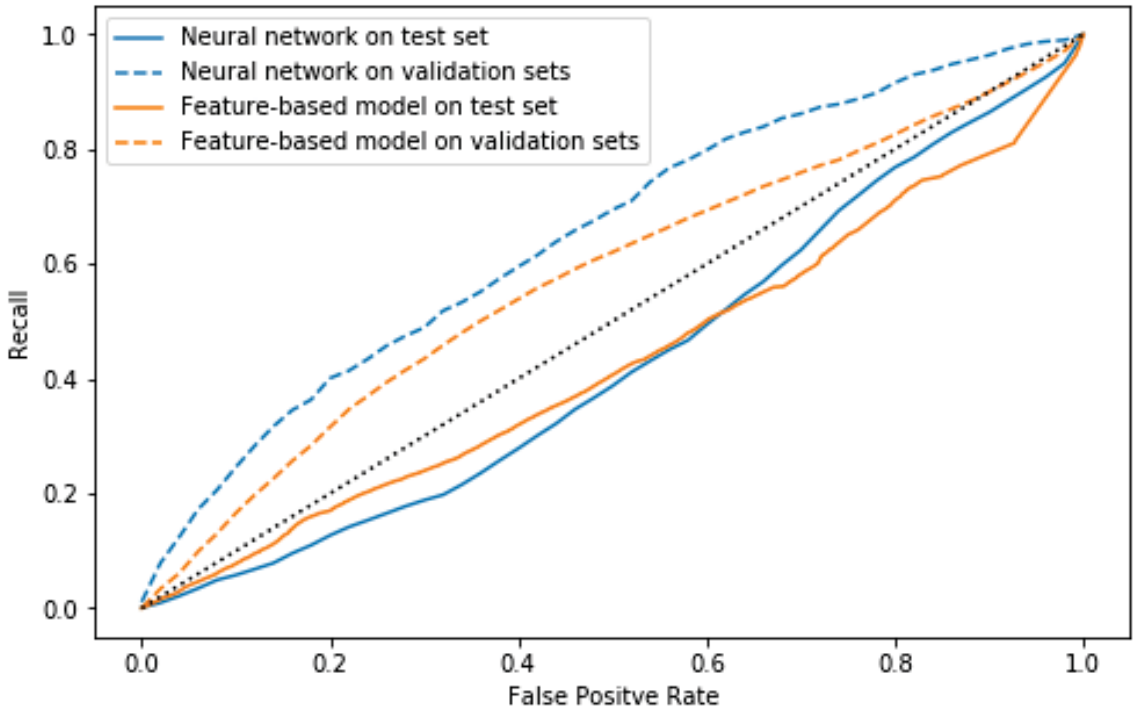

Figure 10: ROC curves of the feature-based model and of the neural network on the test set and on the validation sets

Figure 11 shows a visualization of the risk of accidents estimated by the neural network model on examples of the test set. Each row corresponds to a different driver and the x-axis represents time. Each colored rectangle correspond to an example, and its color represents

the prediction of the model. On this figure, we observe that the model usually estimates the same risk of accident for different examples corresponding to the same driver at different times during the study period. This is interesting because the model has no knowledge of which driver an example correspond to.

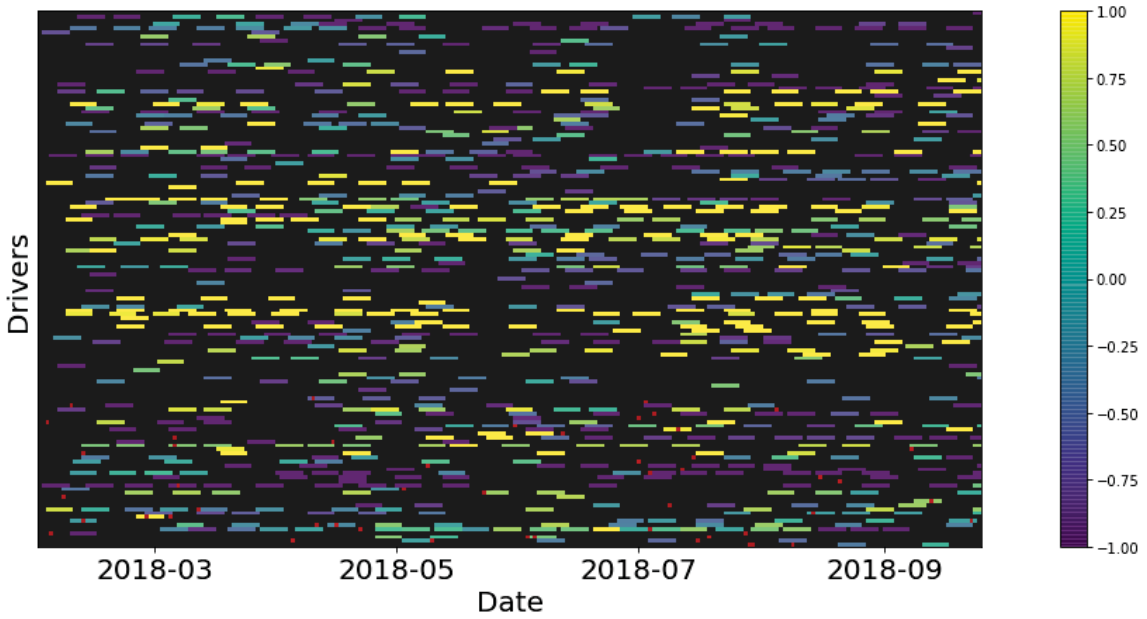

Figure 11: Visualization of the risk of accidents estimated by the representation-based model

3.6 Discussion

With performances on the test set worse than those of a random classifier, we cannot say that we were able to estimate the risk of accident accurately in this study. In this section, we discuss the reasons that could explain those results.

Road accidents are caused by a combination of many factors: how the driver drives, but also on which road they drive and under which weather and traffic conditions. In this study, we only use data describing the driving: we expected that by looking at whether a full-time driver had an accident during a long study period of 18 months, these other factors would

average out. That is to say that during the study period, drivers would have met all kind of driving conditions and that on average drivers with accidents would show a different driving style. Given the results we obtain, the study period or the number of drivers might not be long and high enough for this to happen.

In addition, the driving style of a given driver is likely to significantly vary over time. Indeed, quality of driving is likely to be affected by the hour of the day and fatigue. This means that a driver who had an accident during the study period because they were particularly tired on that day might not necessarily show dangerous driving patterns during the rest of the study period, and our labelling method would result in misclassified training examples. This would suggest to label examples as dangerous only when they occur during the few days before an accident. However, it might also happen that a driver always drives dangerously, but because accidents are very rare only has one or even no accident during the study period. This would also result in many misclassified training examples. We can also imagine cases where very careful drivers are involved in an accident due to other factors such as bad weather conditions or bad behaviors of other users of the road. These problems result in a very noisy labelling of examples. Machine learning can work with noisy labels as long as the majority of examples are correctly classified, but it requires more examples or a high inductive bias. It could be interesting to experiment with semi-supervised learning methods and different labeling methods to see if they would help to deal with these issues.

Our telemetric data might not be able to describe driving behaviors accurately enough to estimate the risk of accidents. For example, information about the use of the steering wheel is only available through the lateral accelerometer, as there is no sensor on the steering wheel. An important part of driving is the observation of everything that is happening outside the vehicle. A good driver not only drives carefully and maneuvers smoothly, but also consistently and accurately monitors everything happening on the road. The sensors

we have access to do not give information about this important part of the driving activity. This part of driving is especially important for our dataset, because as we have seen in Section 3.3, most accidents do not happen on the road, but at slow speed during maneuvers. It could be interesting to add sensors in trucks to collect data about visual checks performed by the driver. We believe that recent improvements in computer vision make it possible to use a camera aboard the vehicle to determine whether visual checks have been performed for example.

The relation between telemetric data and the risk of accident is complex. A more dangerous driver is probably not simply a driver with an higher average speed, it is likely to be a driver for which the evolution of telemetric data in specific contexts follow different patterns than safer drivers. For example, we might be able to evaluate to what extent a driver anticipates turns by looking at the evolution of the speed before a turn. With the representation-based approach, this would mean that the transformations from the raw data to a useful representation are quite complex. For this reason, we think that a neural network capable of applying such a complex transformation would require many layers and maybe more powerful structures than only convolutional layers. For example, the use of attention might make sense for our task, as it would allow the neural network to focus on windows of the time series that are particularly useful to assess a driver's driving style. However, the limited size of our dataset does not make it possible to train such networks, and for this reason we experimented mostly with relatively small networks for this study.

Another difficulty that we face when using machine learning to predict rare events like road accidents is the data imbalance issue. Indeed, machine learning algorithms tend to focus on the majority class and fail to account for other classes. It is quite easy to deal with this issue by assigning a higher weight to examples of the minority class, or by resampling the dataset. However, data imbalance sometimes hides another issue which is harder to deal with: a too small sample of examples for one class. With too few examples from one class,

it is harder for the algorithm to learn significant characteristics of the class to discriminate it. Accidents are rare, so most examples belong to the negative class. Combined with the limited size of our dataset, it makes it harder for the models to train without overfitting the positive examples of the training set.

Our results show that there is a big difference between the performances of the models on the validation sets and their performances on the test set. This might be because many hyperparameters were determined by looking at the performances on the validation sets. The validation set was used to determine how to create instances: the length of the windows and the number of window per example. It was also used to determine how to label examples, to choose which accidents are considered predictable and for how long before an accident the driving data is labelled as positive. Finally, it was also used to determine the list of sensors to use and the hyperparameters of the models. Some of these hyperparameter values might be indeed better in general, but some of them might be particularly better just for the limited training and validation datasets and artificially increase performances reported using the validation set.

Because of the limited size of the dataset and the issue of noisy labels discussed earlier, the measure of the performances on a subset of data is probably not reliable enough to take decisions. Indeed, the standard deviations of the areas under the ROC curve obtained with different splits of the k-fold cross-validation are quite high (7% and 9% respectively for the feature-based approach and the representation-based approach).

In Figure 11, we observed that the neural network model usually estimates the same risk of accidents for examples from the same driver at different dates. This suggests that the model bases its prediction on characteristics of driving that are invariant over time for a driver. This could be because the accident risk indeed does not change much over time for a driver, but it could also be simply because of the way the model is trained. During training, the model does not know that we want it to predict the risk of accident, it only

has access to pairs of driving data and labels. As discussed in part 3.4.4, most examples from the same driver have the same label. Because of this, the most simple way to learn the mapping between driving data and labels might be to simply recognise drivers. Once the model has learned to recognise drivers from the training set, it can already achieve an almost perfect score on the training data. When presented an example from a new driver, such model would try to recognise the driving of a driver from the training set and output the accident risk of this driver. This behavior would lead to the kind of results we observe, most examples from the same new driver would look like the same driver form the training set and the model would therefore output the same accident risk.

In other words, this effect could be caused by the fact that most examples from the same driver in the training set have the same label and that it might be easier to identify the driver than to estimate the risk of an accident using the driving data. In order to force the model to learn to recognise safe driving as opposed to who his driving, we might need a higher number of different drivers in the training set. With more drivers, it would becomes more difficult for the model to learn what the driving data of each driver look like and become necessary for the model to start making links between the driving data of different drivers with the same labels. A different approach could also help to deal with this issue without requiring a higher number of different drivers. For example, we could frame the problem as a meta-learning problem for which each task consists in classifying driving data from one driver depending on whether it was followed by an accident or not. This would prevent the model from cheating by recognising the driver since each episode would contain only data from one driver. By using meta-learning, the meta-model could learn how to train a good accident risk estimator by putting together knowledge from different drivers.

Our results show that using telemetric data to estimate the risk of accident of a particular driver is not easy, but it does not mean that it is impossible. It might require a bigger dataset and a community of researchers and machine-learning practitioners to find the right

approaches and methods. For this reason, we think it would be useful to publish our dataset and make it accessible to anyone who wants to work on this problem. However, the publication of such a dataset raises important ethical issues, as the raw dataset contains personal information such as the GPS position and the work schedule of the drivers which cannot be published. Publishing a preprocessed version of the dataset would restrict the way one can frame the problem.

3.7 Conclusion

We can still not give a definite answer to the question: “Can we estimate truck accident risk from telemetric data using machine learning?". In our study, with the dataset we had access to, it is unlikely to be possible. Indeed, we experimented with two different approaches and many different methods without success. It would be interesting to see if this task would become possible with larger datasets, including more drivers and with data from different sensors. We believe that the estimation of the risk of accidents of a driver based on its driving data remains a very difficult machine learning problem. Indeed, because of the many factors that determine the occurrence of an accident, the number of accidents does not seem to be a good surrogate for driving quality. It might be necessary to use a different approach to teach a machine learning model what safe driving looks like.

Chapter 4

Conclusion and future work

In this thesis, two aspects of machine learning for road accident prediction were covered. With the first research project, the commonly-found problem consisting in finding times and areas with increased risk was extended with a higher prediction resolution. Historical data, weather information, and characteristics of the road network were enough to reach interesting predictive performances. I believe this analysis can easily be reproduced for other cities as long as they collect historical road accident data. Weather and road network datasets are available for all cities in Canada [32, 28]. A project extending this analysis to the state of Quebec is planned for summer 2020. In the future, it would also be interesting to see what performances can be obtained by using additional datasets to add more features. In the city of Montreal, datasets containing the history of ongoing construction work on the road network could for example be included. Finally, I believe that it would be interesting to evaluate other machine learning methods on this problem since only tree-based methods have been explored so far. Some machine learning algorithms are intractable because of the very high number of examples but some non-tree-based methods are scalable enough to be attempted. A neural network with the use of embeddings to encode categorical features, for example, could be an interesting model to experiment with and for which the very high number of examples would not be a problem. It would also be interesting to further

experiment with the Balanced Random Forest algorithm to understand why it did not lead to significantly better results on this highly imbalanced problem.

With the second research project, we experimented with a different road accident prediction problem consisting in the estimation of the risk of having an accident for a given driver based on its driving behavior. Despite experimenting with two different approaches successfully used for many time series classification problems, and making many different attempts, we were not able to reach good performances. Most of the work done towards this thesis consisted of finding and evaluating new ideas in order to reach good performances. Only the final model and the final data processing procedure are presented in the second chapter of this thesis since presenting the many attempted approaches would not fit the format of a research paper. For example, I experimented with a regression approach consisting in predicting the number of days before the next accident, I thought this approach could help by avoiding the threshold effect of classifying driving examples depending on how far in the future an accident happened. As mentioned in the paper, various ways to label examples have also been considered. After spending the first months experimenting with the feature-based approach for which I had some experience, I decided to attempt using a deep neural network approach based on the survey paper on deep learning for time series classification [20]. A lot of time was spent on this new approach, indeed the design of a neural network architecture involves many decisions and requires specific knowledge that I have acquired on my own during the most part of my thesis. A significant effort has been invested in adapting existing neural network architectures for time series classification to the needs of this specific problem. Indeed, even after implementing usual regularization techniques our model still suffered from overfitting. In hindsight, with the understanding I gained, some of the decisions taken were probably not the best ones. All decisions were based on experimentations and results on the validation set, however, given the size of our validation set it might have been best to rely more heavily on my developing understanding

and intuition. The issue of the inconsistent validation results had been identified early on, and I also spent time experimenting with various metrics and evaluation methods in order to try to obtain a more trustworthy method to compare different models. For example, I compared the stability of the precision, recall, F1 score, area under the ROC curve, area under the Precision-Recall curve and area under the Precision-Recall-Gain Curves curve metrics [25].

After having invested substantial efforts in finding an approach and a model able to perform this task without success, we can conclude that this task is much harder than initially thought. It would be incorrect to say that this task is unfeasible, but I am confident that with the available dataset, it is not possible to reach useful performances. The main challenge during this research project has been to gather useful insights into which methods and hyper-parameters are most efficient. Indeed, results obtained on the validation sets always appeared noisy and, eventually, were found not to be representative of the results on the test set. Yet, in the absence of previous studies making use of this type of data, cross-validation results and intuition are the only means to find a successful model. This difficulty to evaluate the real performances of the model might be caused by a too-small dataset compared to the complexity of the task, it could also be caused by the fact that our data examples are not independent and identically distributed (IID) and by the limited number of drivers in the dataset. It would be interesting to continue studying this task with the help of a bigger dataset, however, it is likely to be difficult to get access to bigger datasets. In the future, it would be interesting to study machine learning approaches to make better use of non-IID data. Indeed, for many problems, it is easy to gain many samples from each subject but harder to gain data from many different subjects. Approaches inspired by domain adaptation techniques and meta-learning would be interesting to explore. This task could be useful to evaluate different methods designed to help dealing with non-IID data.

Bibliography

[1] A. Abanda, U. Mori, and J. A. Lozano. A review on distance based time series classification. *Data Mining and Knowledge Discovery*, 33(2):378–412, 2019.

[2] J. Abellán, G. López, and J. de Oña. Analysis of traffic accident severity using decision rules via decision trees. *Expert Systems with Applications*, 40(15):6047–6054, 2013.

[3] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[4] A. Bagnall, J. Lines, J. Hills, and A. Bostrom. Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535, 2015.

[5] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

[6] P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):31:1–31:50, Aug. 2016.

[7] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[8] L.-Y. Chang. Analysis of freeway accident frequencies: Negative binomial regression versus artificial neural network. *Safety Science*, 43(8):541 – 557, 2005.

[9] L.-Y. Chang and W.-C. Chen. Data mining of tree-based models to analyze freeway accident frequency. *Journal of Safety Research*, 36(4):365 – 375, 2005.

[10] C. Chen and L. Breiman. Using random forest to learn imbalanced data. *Technical Report, University of California, Berkeley*, 2004.

[11] Q. Chen, X. Song, H. Yamada, and R. Shibasaki. Learning deep representation from big and heterogeneous data for traffic accident inference. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, pages 338–344, 2016.

[12] T. Chen. Notes on parameter tuning. `https://xgboost.readthedocs.io/en/latest/tutorials/param_tuning.html`, 2019.

[13] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–794, New York, NY, USA, 2016.

[14] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307:72–77, 2018.

[15] M. Christ, A. W. Kempa-Liehr, and M. Feindt. Distributed and parallel time series feature extraction for industrial big data applications. *arXiv preprint arXiv:1610.07717*, 2016.

[16] City of Montreal. Montreal vehicle collisions. `http://donnees.ville.montreal.qc.ca/dataset/collisions-routieres`, 2019.

[17] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations*, 2016.

[18] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML. The ucr time series classification archive, October 2018.

[19] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Sixth Symposium on Operating System Design and Implementation (OSDI)*, pages 137–150, San Francisco, CA, 2004.

[20] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[21] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean. Inceptiontime: Finding alexnet for time series classification. *arXiv preprint arXiv:1909.04939*, 2019.

[22] Federal Motor Carrier Safety Administration. Current FMCSA Crash Cost Figures. `https://www.fmcsa.dot.gov/sites/fmcsa.dot.gov/files/docs/FMCSACrashCostCalculationsDec08.pdf`, 2008.

[23] Federal Motor Carrier Safety Administration. Safety is good business. `https://www.fmcsa.dot.gov/safety/good-business/safety-good-business`, 2014.

[24] Federal Motor Carrier Safety Administration. Large Truck and Bus Crash Facts. `https://www.fmcsa.dot.gov/sites/fmcsa.dot.gov/files/docs/safety/data-and-statistics/461861/ltcbf-2017-final-5-6-2019.pdf`, 2017.

[25] P. Flach and M. Kull. Precision-recall-gain curves: Pr analysis done right. In *Advances in neural information processing systems*, pages 838–846, 2015.

[26] E. Fumagalli, D. Bose, P. Marquez, L. Rocco, A. Mirelman, M. Suhrcke, and A. Irvin. *The high toll of traffic injuries: unacceptable and preventable*. World Bank, 2017.

[27] A. Gandomi and M. Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management (IJIM)*, 35(2):137 – 144, 2015.

[28] Government of Canada. National road network. `https://open.canada.ca/data/en/dataset/3d282116-e556-400c-9306-ca1a3cada77f`, 2017.

[29] Government of Canada. Open data 101. `https://open.canada.ca/en/open-data-principles`, 2017.

[30] Government of Canada. The cost of injury in canada. `https://www.canada.ca/en/public-health/services/injury-prevention/cost-injury-canada.html`, 2018.

[31] The government of canada is improving safety in the commercial driving industry. `https://www.canada.ca/en/transport-canada/news/2019/06/the-government-of-canada-is-improving-safety-in-the-commercial-driving-industry.html`, 2019.

[32] Government of Canada. Historical climate dataset. `http://climate.weather.gc.ca/`, 2019.

[33] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87, 2004.

[34] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[35] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering (TKDE)*, pages 1263–1284, 2008.

[36] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[37] A. Hébert, T. Guédon, T. Glatard, and B. Jaumard. High-resolution road vehicle collision prediction for the city of montreal. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1804–1813, 2019.

[38] ISAAC Instruments. Truck fleet management solutions. https://www.isaac.ca/en/, 2020.

[39] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[40] G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research (JMLR)*, 18(1):559–563, Jan. 2017.

[41] Library of Congress. KML, version 2.2. https://www.loc.gov/preservation/digital/formats/fdd/fdd000340.shtml, 2017.

[42] L. Lin, Q. Wang, and A. W. Sadek. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. *Transportation Research Part C: Emerging Technologies*, 55:444 – 459, 2015.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2999–3007, 2018.

[44] J. Lines and A. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592, 2015.

[45] J. Lines, S. Taylor, and A. Bagnall. Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1041–1046. IEEE, 2016.

[46] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[47] M. M. Chong, A. Abraham, and M. Paprzycki. Traffic accident analysis using machine learning paradigms. *Informatica*, 29:89–98, 05 2005.

[48] J. Milton and F. Mannering. The relationship among highway geometrics, traffic-related elements and motor-vehicle accident frequencies. *Transportation*, 25(4):395–413, Nov 1998.

[49] A. Najjar, S. Kaneko, and Y. Miyanaga. Combining satellite imagery and open data to map road safety. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pages 4524–4530, 2017.

[50] O. of the Federal Register, editor. *Code of Federal Regulations, 49 CFR 395*. Office of the Federal Register, 10 2019.

[51] G. W. H. Organization, editor. *Global Status Report on Road Safety 2018*. Geneva: World Health Organization, 2018.

[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[53] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. Hyder, E. Jarawan, and C. Mathers, editors. *World Report on Road Traffic Injury Prevention*. Geneva: World Health Organization, 2004.

[54] SAAQ. Road safety record. `https://saaq.gouv.qc.ca/en/saaq/documents/road-safety-record/`, 2018.

[55] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.

[56] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[57] A. Theofilatos. Incorporating real-time traffic and weather data to explore road accident likelihood and severity in urban arterials. *Journal of Safety Research*, 61:9–21, 2017.

[58] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[59] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. Class imbalance, redux. In *IEEE 11th International Conference on Data Mining (ICDM)*, pages 754–763, Dec 2011.

[60] D. Wilson. Using machine learning to predict car accident risk. `https://medium.com/geoai/using-machine-learning-to-predict-car-accident-risk-4d92c91a7d57`, 2018.

[61] K. S. Woods, C. C. Doss, K. W. Bowyer, J. L. Solka, C. E. Priebe, and W. P. Kegelmeyer Jr. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 7(06):1417–1436, 1993.

[62] Z. Yuan, X. Zhou, and T. Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pages 984–992, New York, NY, USA, 2018.

[63] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica. Apache spark: A unified engine for big data processing. *Communications of the ACM*, 59(11):56–65, Oct. 2016.

[64] G. Zhang, C. Wang, B. Xu, and R. Grosse. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2019.