POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Predicting Bus End-Trip Delays Using Different Machine Learning Algorithms to Model Planning Effectiveness

VICTOR HANNOTHIAUX
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques appliquées

Juin 2019

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Predicting Bus End-Trip Delays Using Different Machine Learning Algorithms to Model Planning Effectiveness

présenté par **Victor HANNOTHIAUX**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Guy DESAULNIERS, Ph.D., président
Andrea LODI, Ph.D., membre et directeur de recherche
Louis-Martin ROUSSEAU, Ph.D., membre et codirecteur de recherche
Charles FLEURENT, Ph.D., membre

DEDICATION

“Study the past if you would define the future.” . . .
Confucius

ACKNOWLEDGEMENTS

I would like to thank my research director Professor Andrea Lodi and my co-director Professor Louis-Martin Rousseau for guiding me and for giving me their confidence throughout the project. They helped me to distinguish the essential aspects of the project, and they always made time for me when necessary.

Also, I would like to thank Charles Fleurent, Loïc Bodart, Catherine Bédard, David Johannès and Madjid Aoudia from Giro who gave me access to all the data I needed and their interest for the project during the numerous meetings. They were always quick to answer any of my interrogations.

I would like to thank the chair of "data science for real-time decision making" and its partners ; GERAD, IVADO and CIRRELT, and the people who work here and helps me a lot : Mehdi, Koladé and Khalid.

I would like to thank the department of Mathématiques Appliquées et Génie Industriel (MAGI) of Polytechnique Montreal.

I would like to thank Ludovic, Igor, Defeng, Antoine, Mike and all the other postdocs, Ph.D. students and masters from Polytechnique for their help during the project.

I would like to thank James Moran, my squash partner, who took the time to read my thesis and help me to correct it.

Finally, I would like to acknowledge my family and my friends for their support, and especially Vincent, Jordan, Morgane, and Valentine.

RÉSUMÉ

Le transport public existe presque partout dans le monde. Cela permet à toutes les personnes le désirant de se déplacer d'un endroit à un autre d'une ville de façon économique et écologique. De plus, de plus en plus de données sont disponibles de nos jours grâce aux systèmes embarqués à l'intérieur des véhicules. Ces données pourraient être utilisées dans une optique de prévision des retards, qui permettraient par la suite de les anticiper. Ainsi la fiabilité des horaires serait améliorée et plus de gens seraient suceptibles d'employer ce mode de transport. Des travaux ont été réalisés afin de prédire les retards en utilisant différentes données, cependant aucune d'elle ne l'a fait dans l'idée d'intégrer ces prévisions dans les procédures de création de planification de trajet.

Au cours de ce mémoire, divers modèles de prédicition de retard pour les fins de trajet sont essayés. Il ne s'agit pas de prédire le retard exact, mais de classifier les retards des fins de trajet. Afin d'être utile aux planificateurs d'horaires, ces modèles n'utilisent que des données qui peuvent se trouver en amont de la planification. Les données exploitées pour les modèles sont des observations historiques de la ville de Montréal. Deux problèmes de classification sont abordés au cours de ce mémoire. Le premier est un modèle de classification binaire qui prédit si un bus va finir son trajet en retard ou à l'heure. Le second est un modèle qui prévoit dans quel créneau de retard le bus va finir son trajet. Pour chacun des problèmes, trois algorithmes de machine learning pour l'estimation des retards sont testés : réseau de neurones, forêt aléatoire et arbre stimulé par gradient. De plus, une régression logistique est également testée afin de comparer les résultats par rapport à une méthode plus standard. Les modèles sont optimisés selon différentes méthodes et sont comparés en terme de précision et de temps d'entraînement.

Les modèles sont par la suite entraînés sur une période et testés sur d'autres afin d'étudier la possibilité d'intégrer ces modèles dans le processus de création de lignes. Par la suite, les prédictions sont utilisées afin de créer des distributions de probabilité pour les différents crénaux de retard pour les fins de trajet des bus. Les différents algorithmes sont testés afin de distinguer ceux qui reproduisent au mieux la réalité.

Le projet conclut sur la possibilité d'utiliser les données de planning pour prédire le retard des fins de trajet des bus. Une classification sur plusieurs classes peut être améliorée en intégrant de l'apprentissage non supervisée afin de déterminer les classes de retard. Il est également possible d'entraîner un modèle sur des périodes passées afin de prédire sur de futures périodes, mais cette méthode doit être encore améliorée.

ABSTRACT

Public transportation services are provided in almost all the cities of the world. They allow people to move through the cities in an economical and eco-friendly way. The buses are one of the possible solutions for public transportation. Moreover buses are interesting to study because more data are available from onboard systems and can be used to optimize service quality.

Indeed, preventing delays could improve service reliability and thus make people more likely to use public transport instead of their cars, which are currently more comfortable and more reliable. The first step in this process would be to forecast the delays. A lot of factors are linked to delays: peak-hour traffic, weather or accidents, etc. Some studies were conducted to predict end trips delay using real-time input which does not allow improvement to schedule reliability because these data are not available during planning.

This research focuses on modeling end-trip arrival time for each bus trip based only on offline input available to public transport planner. The models do not intend to predict the exact delays, but rather to classify them. The delays used to train and test the models are historical observations from the city of Montreal in autumn 2017. Two different classification problems were treated. The first one estimates the probability for a trip to end on-time or late. The second one estimates the slot of delay. For each problem, three different machine learning models were built and optimized: random forest, gradient boosted tree and artificial neural network. Also, logistic regression was tested in order to compare the results. Several optimization methods were tried. The models are compared in term of accuracy, recall, f1 score and training time.

The data from another period (autumn 2016) were then added to the database, and the model tested on the aggregated database. The model accuracy remained constant after the addition of the new period. The models were then fit on a single period (autumn 2016) and tested on the other one (autumn 2017) in order to check the possibility to use the model to forecast future schedules. The prediction is then used to generate a probability distribution for the different trips to end late to assess service reliability. The probability distributions are then compared with reality by comparing the distance between them and the frequencies of delays for the different trips. Normal distribution was also tested and obtained better results than the machine learning models.

The project concluded that it is possible to model end trip delays using offline data. Multi-label classification can be improved by using unsupervised learning to determine classes.

There is also a potential of training the models on some periods in order to predict for future ones.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

EU	European Union
GPS	Global Positioning System
STM	Société de transport de Montréal
OTN	Original Trip Number
HLP	Haut-le-pied
AI	Artficial Intelligence
ML	Machine Learning
RF	Random Forest
GBT	Gradient Boosted Tree
MLP	Multilayer Perceptron
SVM	Support Vector Machine
OD	Origin-Destination
APTS	Advanced Public Transportation Systems
APC	Automatic Passenger Count
AVL	Automatic Vehicle Location
ETA	Estimated Time of Arrival
RFID	Radio Frequency Identification
OTN	Original Trip Number
UOTN	Unique Original Trip Number
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ROC	Receiver operating characteristic
TS	Time Slot

CHAPTER 1 INTRODUCTION

Public transit is a mass transportation system available to all and operated on defined lines and schedule, often managed by private companies under contracts of public instances. Public transport is from an economic and environmental point of view, the most effective means of transportation. Trains, subway, buses, and ferries are examples of public transport.

Several features are essential for a satisfactory public transport service, such as price, comfort and travel time. One of the most important features is the reliability of the scheduling, as stated by Peek and Van Hagen (2002) [1]. With the deployment of Advanced Public Transportation Systems (APTS), it is getting easier to gather information about the trip and thus to monitor the efficiency of the schedules.

Moreover, the cost of components giving access to real-time information about bus trips has decreased, which has led to a global deployment of these systems. The data extracted could help to improve the public transport service and reduce their costs.

1.1 Framework of the project

A definition of the basic concepts is essential to understand the project and to avoid misinterpretations. To characterize the subject, the following sections will present the Montreal bus network and of the contractor in charge of managing the planning : GIRO.

1.1.1 Basic concepts and definitions

A bus is a transport facility which can transfer a limited number of people at the same time ; large buses can accommodate up to one hundred persons. A stop is a place where the bus halts along its way and where people can get in or out of the bus. A line is a collection of stops which determines a path from one terminus to another. A trip is a line covered by a bus, leaving a terminus at a determined time. A block is a sequence of trips that a bus covers between its start at a depot and its return. A layover is a buffer time during two trips of a block.

The figure 1.1 is an example of the schedule of a block. In this example, a bus is covering a sequence of 3 trips. The layover at the different terminuses serves the purpose of preventing the propagation of delays. When a bus finishes the last trip of a block, it comes back to the depot.

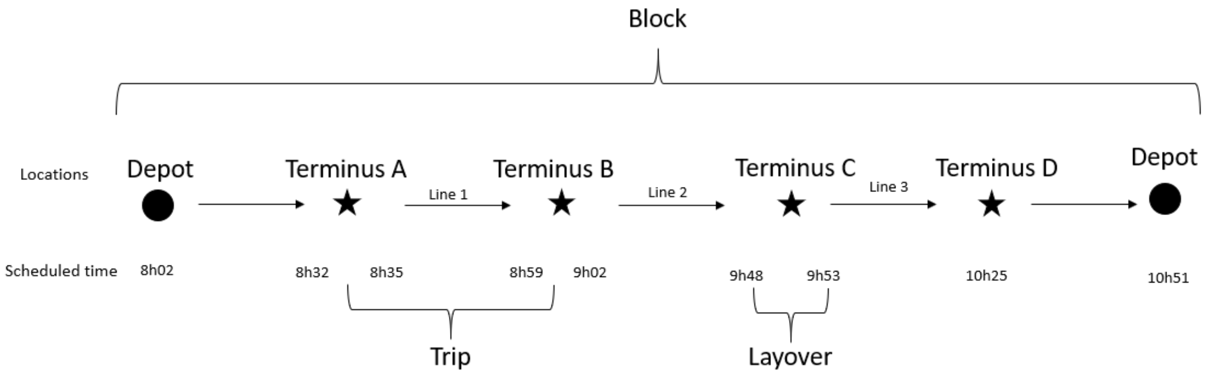


Figure 1.1 Concepts schematic

At the end of a trip, there are several possibilities : the bus can either do the same line in the return way, or stay at the same terminus and cover another trip, or go to another terminus and cover another trip. The trip intra terminus without covering any trip in the last option is called an Haut-le-pied (HLP) or deadhead.

1.1.2 Case study : the Montreal Bus network

The population of the island of Montreal is about 1,705,000 inhabitants in an area of 431,5 km^2. The Montreal urban design is the superposition of an old urban division which comes from the French seigneurial period and the classic checkerboard pattern from North-American cities [2]. One of the characteristics of the road organization in Montreal is the few numbers of left-turn.

The Société de transport de Montréal (STM) administrates Montreal public transportation. In 2016, 221 bus lines were in activity, which represented about 4300 trips a day. Within the four subway lines, it is more than 429,5 million of travels during this year. The busiest bus lines could transport 31 777 people on average each day [3]. A new metro line is actually under construction and could be active for 2025.

The continental climate of the island of Montreal complicates public transport planning. Indeed, the cold winters damage the roads and make road works necessary during the rest of the year. It is an additional difficulty for the schedules of public transport.

More than a third of the vehicles in Montreal are equipped with APTS devices, mostly for passenger counting and GPS location over time. For each trip covered by a bus equipped with APTS devices, data are available at each stop, such as date, line number, direction, block

ID, trip ID, vehicle ID and the number of people in the vehicle. At each trip covering a line at a specific time is given an Original Trip Number (OTN). Furthermore, for each record, the scheduled time and the real departure time are available.

1.1.3 GIRO

The project has been done with the partnership of GIRO. GIRO is a Canadian company founded in 1979 which provides software solutions for Public transport and postal organizations to plan, optimize, and manage their operations. Its clients are all over the world (the STM in Montréal, the SNCF in France, MTA in New-York, SBS in Singapore...). GIRO is providing the software Hastus to the STM in order to help in the scheduling and the operations. This software could manage a fleet of more than six thousand vehicles.

One of the current GIRO challenges is to integrate machine learning tools in their solutions for improving the reliability of their schedule, predicting the delay or the absence rate among the drivers. For that purpose, it has developed a partnership with universities and Research Chair, such as the Canada Excellence Research Chair in Data Science for Real-Time Decision-Making.

1.1.4 Machine Learning

Articial Intelligence (AI) is a field of computer science and was first introduced in 1956. AI includes all the theory and practices about creating things able to reproduce intelligence. Machine Learning (ML) is one of the study fields of AI. ML is based on a statistical approach in order to allow computers to improve their way of solving problems by themselves. The implementation of such methods is an essential part of ML.

In the first part of a ML implementation, the objective is to model an existing database. The model uses features as inputs and targets as outputs. This phase is called the learning phase. The second part is to try to predict future outputs using a combination of features. The learning can be supervised if the desired predictions are known : for example classification or regression. However, on the contrary, the learning is qualified of unsupervised if the objective of the learning is to determine and define the structure of the data [4].

There are several ML kinds of models : decision trees (which include Random Forest (RF) and Gradient Boosted Tree (GBT)), Support Vector Machine (SVM) or Artificial Neural-Network (ANN), just to quote some.

1.2 Problem

Deciding of the problem of the thesis was one of the main difficulties of the project. Indeed, at the beginning of the project, the framework was not specified. The first objective was to get ideas of various projects and determine their potential in term of machine-learning and their feasibility.

As the service quality is one of the main issues for public transportation planners and because GIRO is one of the main actors in this sector, it was decided that the project would aim to improve service quality.

An analysis of the available data led to the first outline of a problem : forecasting the delay at each stop using real-time data. However, the literature review showed that this problem was addressed by several before. Moreover, GIRO acts upstream in the public transportation system, and no real application could be found. Working with off-line data was essential for the project. Then, the idea of working with the layover appeared.

The analysis of the data shows that the scheduled time is not always respected, and the delay could be propagated inside a block. The layover, which is a buffer time, could prevent the propagation of delay. Thus it was thought that a new method to find adequate layover could be developed.

The layover is the main leverage a public transport planner could use in order to prevent delay propagation. Predicting the departure status of the trip was then the new objective of the thesis. However, once the situation is modeled, it just described the adequacy of the layover in this particular situation, and the importance of the feature layover was too significant. It was too difficult to transpose the models to other periods. The final idea was to forecast the delay of the end-trip, in order for public transport to have an idea of the necessary buffer time to prevent delay propagation.

1.3 Objectives

The first objective of this project was to determine the potential of the available data in machine learning processes. To decide it, an analysis of the different features was necessary. This analysis led to decide the other objectives of the projects.

The second objective was to find out if it was possible to model the end trip delays using offline data in Montreal. The data used come from Autumn 2017. Different machine learning models were candidates for this. Both binary and multi-label classifiers models were tested.

Then the next step was to try if it was possible to train the different models on one period

(Autumn 2016) and test it on another (Autumn 2017). The objective of this part was to assess the possibility to predict end trip delays for future periods.

The ultimate objective of this thesis was to prepare a final deliverable for GIRO, predicting the probability distributions of delays for the different trips to help to create more reliable schedules. These distributions were compared to a normal distribution.

1.4 Work overview

This dissertation displays the work done to estimate the bus end trip delays. It also shows examples of how these models could be used practically. The thesis which presents this work is composed of different parts.

The first section presents a literature overview of the several topics of this thesis : the scheduling, arrival time prediction models, service reliability and APTS technologies. The analysis of arrival time prediction models shows that the use of only scheduling data had not been done yet, and this paper offers to fill this gap.

The second chapter describes the methodology and the several sources used to create the database. An analysis of the database is presented, as the correlation between the different features.

The third chapter describes the building and the optimization of the machine learning models which were used to estimate the delays for the various trips of the schedule. This chapter first presents binary classifying models and the second part describes multi-label ones. The last section shows the results of a model fits on one period and tested on another one. The algorithms are compared with a metric combining accuracy and recall : f1 score.

The fourth chapter presents the proposed method to assess schedule reliability. The trips are aggregated by time slot, line, and direction, and the probability distribution for the different delay classes are proposed. An analysis of the distance between the different probability distributions and reality described by frequencies in each category is presented.

Finally, the thesis concludes with the best of the algorithms presented in this work which could be used to predict bus end trip arrival times and proposes ideas for future work.

CHAPTER 2 LITERATURE REVIEW

2.1 Public transport planning

The public transport planning decomposes in several sub-problems [5]. The first one is estimating the demand, the second one is deciding the network design, the third one is creating the vehicle scheduling and the last one is the driver assignment. The following section will describe currently operated algorithms used to answer these different problems, such as the various factors essential for quality service in public transportation.

2.1.1 Planning methodology

The demand is often estimated by analyzing the previous data and estimating the habits of people. Origin-Destination (OD) matrices are used to model the demand, by estimating the number of people susceptible to go from the various origins to the different destinations. The OD matrices are created depending on the time of the days and are the best ways of estimating the needs of the public transportation network. Such matrices could either be created in static or dynamic way [6]. External factors such as fares or quality of the service should be taken into account to improve their accuracy [7].

Designing a network is a problem which aims at deciding the topology and the frequency of the operated lines to fulfill the demand under constraints of budget [8]. The routes and their recurrences can be assigned either separately using a first route set generation and genetic algorithm [9] or simultaneously [10]. The genetic algorithm also enables multi-objective and creates routes from scratches and produces results near to the Pareto optimum [11].

The vehicle scheduling problem aims at assigning buses to trips to cover a given timetable, taking into account the bus fleet and other practical constraints, such as the number of depots [12]. This problem is more complex than a routing problem because it has time windows as constraints [13]. The layover, or buffer time, is thus one of the leverages for having robust schedules. The algorithm produces sequences of trips which should be covered by buses, also called block. It is in this aspect of public transport planning that the project took place.

The last phase of the problem is driver assignment. Two elements composed this question : the first one is the creation of the different work days and the second one is to assign drivers to these work days, under some constraints such as collective agreements or workers availability. This problem is called the bus driver scheduling problem [14].

2.1.2 Quality factor for planning

Several factors can lead people to use public transportation [15]. These factors are :

- the fare of the public transport ticket, which supports the service provided. However the price is not the factor of the decision in itself, but the difference of expense in comparison to other transport solutions such as the bike or the car.
- the accessibility, in space which is the distance minimum to reach access to the public transport network, and in time which is the waiting time before accessing the facility
- the mean travel time from one origin to a destination. It also has to be compared to the travel time with other transport facilities
- the comfort of the passenger during the journey. The fact that some transports are overcrowded or outdated can have a fatal impact on the way people perceive public transportation.
- the safety
- the service reliability, which is how close the journey is to what expected the passenger. The thesis will develop this topic.

These various quality factors were prioritized in a Maslow pyramid by Peek and Van Hagen in 2002 as the figure 2.1 shows. Some quality factors could be qualified either of "satisfier", which means that their presence could encourage more people to use public transport, or "unsatisfier", which means their absence could prevent people from using public transport.

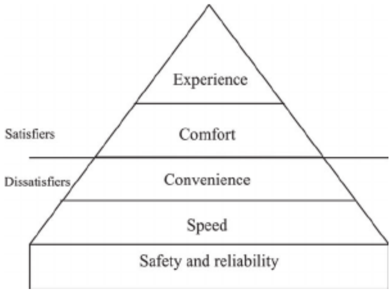


Figure 2.1 Quality factors in public transport presented in pyramid of Maslow (source : Peek and Van Hagen 2002)

Most people still uphold to use the private motor vehicle because of its convenience, despite the environmental degradation. Studies have shown that four major study fields will help to understand how to make people shift from private car to public transportation : improvement in service reliability, creating new evaluation methods, studying specificities of car users and how to make people perceived the benefits of public transportation [16].

2.2 Advanced Public Transportation Systems

APTS are a sub-class of Intelligent Transportation Systems which were defined by the European Union (EU) Directive 2010/40/EU as systems where communication and information manage users, traffic flows, infrastructures and transports [17].

There are different types of APTS, also called intelligent public transportation Systems. The main advantage of using APTS technology is that the data could be collected automatically, and thus not having manual data collection efforts. APTS aims at enhancing efficiency and effectiveness of public transportation infrastructure, using archived data [18] [19]. The following sections will present different APTS systems and how the analysis of the data they provided have improved service quality.

2.2.1 Automatic Passenger Count

For instance, many transit authorities throughout Canada and the United States used Automatic Passenger Count (APC) technology [20]. APC systems can count the number of people using a set of infrared beams which cross stairwells at waist level. Passengers who are boarding or leaving break the beam and so close the switches : systems can count and record the passenger activity. Treadle mats can also be used to count the number of people entering and leaving public transports [21].

More recent counting passengers systems are developed, and they include tracking and validation to people detection. These methods use various image recognition methods, detecting clothes and physical characteristics. They perform well on dense and sparse crowds and can identify more than 20% accurately than other competitive methods [22]. Stereovision can also be used to count the number of passengers in buses and subway [23].

There are multiples objectives in counting the number of people in public transports : monitoring user trends, analyzing performance and identify problematic locations [21].

APC systems have been assessed, and measurement errors exist. However, the percentage of errors with APC systems is not relevant compared to mistakes made during manual data collection techniques [20].

2.2.2 Automatic Vehicle Location

Automatic Vehicle Location (AVL) use Global Positioning System (GPS) devices to track in real-time a whole fleet of buses. These AVL systems were designed at first for monitoring buses in real-time [24], but now it also provides data for offline analysis. AVL systems offer a large amount of data which enable to track the percentage of schedule adherence at each stop and combine with APC data could even connect to the number of customers affected [25].

The schedule adherence is the percentage of buses which arrives within a time window near the schedule one and acceptable for the user. A poor quality schedule adherence can prevent passengers from using public transportation. AVL data can determine the statistical distribution of arrival time at each stop, and so update the transportation times in the timetables, and thus improving transit on-time performance [26]. Another example is the use of AVL data to detect bottleneck in public transportation networks [27].

Even if AVL systems are used to monitor the adherence to a schedule, they can have impacts directly on the willingness of the operator to keep on schedule, and thus on the schedule adherence directly [28].

2.2.3 Smart card

A smart card in public transportation is an electronic device which serves as a proof of subscription or automatizes fare collection systems. The predominant technology used actually in the smart card is Radio Frequency Identification (RFID). Transit agencies used them increasingly because of their efficiency and effectiveness. While collecting payments, they produce large amounts of data which could compare planned and real trips arrival time or count the number of people onboard. The biggest challenge for this technology is to link the different trips made by a user to map their whole journey [29].

Smart card data coupled with cellphone call detail can provide patterns for urban mobility and transport mode choices, and thus find problematic locations which could prevent public transport usage. This method of data mining to determine temporal and spatial variability in transport mode preference has been tested on Singapore [30]. Moreover, mobile phone data give the ability to transit operator to improve the management of their service by monitoring the real demand with the one planned. Travel patterns can be found using mobile phone location data, and then provide accurate OD demand. New routes and schedules which could reduce both travel time and waiting time can be identified [31].

2.3 Service reliability

Improving service reliability could lead more people to use public transports. Moreover, public transport planners try to maximize it. This section describes the service reliability and focuses on schedule reliability.

2.3.1 Definition of service reliability

Service reliability can be defined as the difference between the expected service in time and comfort relative to the one perceived by the user. The time reliability can be assessed by the waiting time compared to the scheduled one. The comfort reliability can be evaluated by the possibility of finding a seat during peak hours and by the percentage of bus overcrowded. The degree of reliability relies mainly on the passenger assumption of the variability.

The planning robustness has been defined as the capacity of a schedule to respond to disruptions without impacting the rest of the network [32]. The leading cause of disturbance in schedules is the presence of delays within the schedules, and thus planning robustness is directly linked to service reliability. Methods had been established to improve planning robustness, which mainly consists of deciding the trade-off between the service quality, service reliability and operating costs [33].

2.3.2 Improving service reliability

One way of improving service reliability is by establishing holding points alongside the lines. Holding points are specific stops where the driver has to wait for a determined departure time to leave the particular stop if he is in advance. When holding points are employed, the additional travel time decreases [34].

Other factors can help in improving service reliability. For that, the STM has introduced smart card, has restricted some lane to bus-only use, has added articulated buses to its fleet and had integrated transit signal priority. Except for the transit signal priority, all the other solutions had positive impacts on service variation, a component of transit service reliability [35].

During the network design, transit planner can also improve service reliability by, for example, choosing line length and stop spacing. The waiting time can be decreased by 65% during the timetable design by optimizing the choice of the percentile value used. A too high value can improve the robustness of a model but can lead to extra waiting time for passenger between different stops [36].

2.3.3 Measuring service reliability

A measure of the service reliability can be done by counting the number of buses arriving within a time window near to the schedule one. However, for some tests, service reliability can be assessed as the number of buses arriving at the destination, regardless of the delay [37].

In the transit field, the ratio volume/capacity has been used to label critical points. A modern method to evaluate network performance has been developed, studying link capacity and network design [38]. This method had enabled more benefits than with the volume/capacity methods, by also quantifying the time saves.

Problematic links labeling is essential to evaluate the vulnerability of a public transport network. A robustness index has been proposed, using the capacity of the network for assessing the service performance [39]. This method locates critical links by using the change in all the network locations and can be used when the network robustness index is not significant.

2.3.4 Delay propagation and bus bunching

Several types of delay exist, such as the current delay, the primary delay (a bus ends a trip late) and the secondary delay (which occurs when a bus ends late a trip because the delay propagated). Cumulative distribution functions could model all of these delays. Thus, it has been possible to represent the delay propagation with an activity graph [40].

Using layover as buffer times could prevent a primary delay from propagating, and thus avoid further problems. Indeed, because of a primary delay, the number of people at each stop waiting for a bus will increase, and thus the probability of stopping at each stop and the dwell time will extend, generating more delays, causing the headway with the next bus to shrink. This phenomenon is called bus bunching. Some researches have tried to use APTS data to determine the causes of bus bunching [41].

Due to delay propagation, passengers could experience several negative aspects : changes in expected travel times, augmentation of the probability to find a vehicle crowed and the impossibility of finding a seat [15].

2.4 Predicting Delay in Public transportation

The main objective of the project was to estimate the end trip delay of trips. The following sections will review different methods used to predict travel time and forecast end trip delays, using various types of data.

The data collected through APTS have enabled the building of accurate models for bus travel times predictions and thus have improved the robustness of schedules. There are various factors which impact on bus travel times : some can be anticipated when designing networks such as the hour of the day and the route, but others cannot, such as weather conditions, traffic accidents or special events. Arrival time has been estimated using various methods, which mainly use real-time data.

2.4.1 Models using percentile methods

The most straightforward method to predict the travel time in public transportation is the use of the percentile travel time method. Using the 95th percentile travel time method indicates that 95 percent of the trips have a travel time shorter than the 95th percentile travel time. However this method is not accurate because generally the trips are not distinguished, and thus this method does not take into account exterior information such as the peak hours.

2.4.2 Models using historical data

Historical data models analyze travel time on the previous trip at the same period to forecast the bus travel time, assuming the traffic journey to remain stationary. Patterns could be found for traffic conditions on a daily and weekly basis, and the forecast could be done accurately for buses arrival time for part of the day, for each day of the week [42]. However, these models cannot take into account the subtleties of traffic.

Models could also use previous average travel time to predict further bus arrival time. This method was mainly used as a reference for other studies. This model was almost every time outperformed by the algorithm proposed by these researches [43]. Other inputs can be taken into account to adjust the results and improve accuracy. The main other independent feature which could be added to the model is the weather condition. Previous travel times can also be combined with dynamic data sources such as AVL systems to develop real-time prediction algorithm [44].

Another historical data model uses the average speed of vehicles over certain links of a route to forecast the Estimated Time of Arrival (ETA) of buses. Different algorithms have been proposed to estimate bus arrival time extracting the average speed from GPS data [45]. A variation of this model has been proposed latter, combining real-time location, real-time traffic, historical travel time, temporal and spatial variations of the traffic conditions [46]. The main factor for the ETA of a bus is its current speed.

The previous models use historical data to make direct forecasts depending on the part of the day. The results could be made more accurate using a Double-Seasonal Holt-Winter's Exponential Smoothing approach. This method allows the model to update itself with four different parameters and thus can take into account seasonality and time-trend in data series. The results show that estimation is approximately 10% better than elementary statistical models [47].

Models using historical data require a considerable set of data which cannot be available. Therefore, these models are not suitable for analysis with too significant variances in the differences in speed or travel times.

2.4.3 Models using statistics

The different factors which impact on bus arrival time are the diver behavior, signal, number of people on the bus. These variables are used as independent ones in the majority of the studies, and the accuracy of the methods rely on their independence [48]. Most of the literature about statistical models have been published before the 1990s.

One of the statistical model used to forecast the ETA of buses is the use of time series. This model uses arrival time from historical periods to predict future arrival times. It is assumed that mathematical functions can provide patterns to determine arrival times, and these patterns will not change in the future. Precision for this model relies mostly on the likeness between previous and actual patterns [49]. Time series were not tested during the project because they need real-time data and we used offline data.

Another statistical model is using a regression model, and it is more likely to work under uncertain traffic condition. These models measure simultaneously the effects of various inputs in order to predict bus arrival times. These inputs can come from APC systems. Distance, boarding passengers, number of stops and weather information were used in various multilinear regression to estimate bus arrival time at terminus [50]. One of the main advantages of regression is revealing the impact of each feature in the model. However, these models are usually outperformed by other models because the input data are highly inter-correlated [49]. This method is relatively easy to implement and was thus used as a comparison to the different machine learning models during the project.

2.4.4 Models using Machine Learning

There are several advantages of using ML models over statistical models. It is easier to deal with interdependent features in input and noisy data can be processed [51]. ML models are used without processing traffic data. However, results for one stop are not assignable to another location because of location specificities. The different machine learning algorithms implemented during the project are ANN, GBT and RF, and are presented in the following sections. SVM models are also described.

ANN have been recently more used for estimating bus travel time because of their ability to model any kind of function, even non-linear one. ANN models predicting end-trip arrival time has been proposed, using only GPS data, trained with historical data and test with real-time information. The ANN model outperformed the historical average model in term of prediction accuracy and robustness [52]. Jeong and Rilett also proposed an ANN model which performed better than multilinear regression models [53]. Furthermore, a model using ANN to predict bus travel time with GPS data and other features such as the number of passengers boarding had been proven efficient on a line of the Delhi public transport network [54].

The gradient boosting tree algorithm is a method based on decision trees, which creates new decision trees based on the errors made on the previous models. These new decision trees are therefore used to adjust the weight of data in the training set, to minimize a loss function. GBT algorithms have outperformed simpler model for travel time prediction [55]. However, the training time could be too considerable. GBT were also used in a method based on predicting the travel time of the different journey segments of a trip. This method also uses queueing theory estimators [56].

SVM is a recent neural network algorithm which tries to find the separating border between different output in order to classify them. The main advantage of SVM is the model find without specific function the relation between input and output. SVM models have been tested using three features : the segment, the travel time of the ongoing section and the latest travel time of the next section. The results showed that SVM model could be used to predict bus arrival time [57]. However, significant problems can cause computation time issues. Bus travel time has also been estimated using real-time road traffic with a SVM model, and this model has outperformed the ANN model [58].

RF algorithms create a specified number of decision trees during the training phase, and for multi-label classification, output the class represented the more in all the decision trees generated. RF models can also be operated for labeling bottleneck in public transport network by estimating a probability distribution for congestion in time and space [59]. This algorithm manages to reduce the prediction error on congested days by 38% compared to a genetic algorithm. Another use of a RF algorithm for public transportation is to predict bus travel time using the near neighbor method [60]. This article demonstrates that even the computation time is slow, the model obtains a high accuracy. RF models have also been used to predict the service reliability of a bus line directly in Dalian (China) by estimating the punctuality, frequency and load factors of the buses [61].

Other machine learning algorithms exist and could have been tested during the project. For example recurrent neural networks, which are a class of artificial neural networks. However, this model is more specific for problems with temporal dynamic behavior and was not well suited for a model with offline data.

2.4.5 Models using Kalman Filter

Kalman filters were adopted along the years for many technologies using treatment signals, such as radar or communication. A Kalman filter uses the dynamic of an object to deduce its static characteristics, hence eliminating the noise of signals. These data can be used for studying the past, analyzing the present or predicting the future.

In some models Kalman filters had been incorporated to predict the expected bus arrival time at individual bus stops, using the filter to make the data more reliable and a neural network for the robustness [62]. Kalman filters can also integrate data from social networks in which people posted news about events they witnessed [63]. This method had been tested on a traffic simulator.

Kalman filters are used for preprocessing the data in models which also combine machine learning. Indeed, Kalman filters clean the data and make them more accurate, which help many machine learning models, especially with real-time data.

2.4.6 Conclusion

Different machine learning models can use APTS data in order to estimate bus arrival time, and they outperformed other models. Techniques such as Kalman filters can improve more the accuracy of the models. However, it has only been used for real-time prediction and not for the estimation of planning overall. The main advantage of machine learning models is that they find some patterns that human could not distinguish.

In this project, machine learning models will be trained on historical data on a planning level, in order to predict future periods, and will be compared to real-time data coming from APTS systems.

CHAPTER 3 DATA ANALYSIS

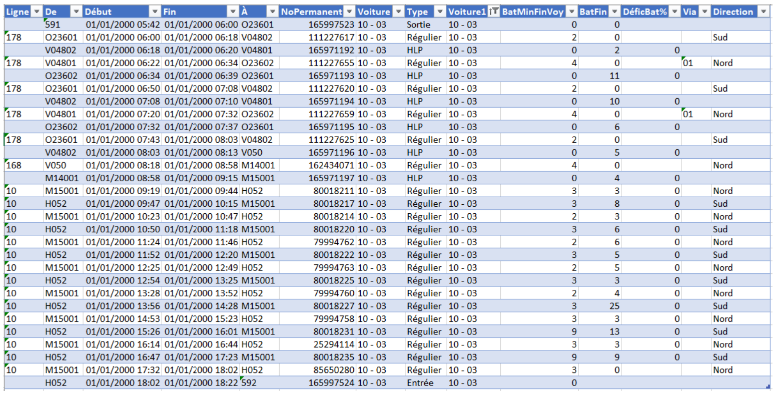
The database is an essential aspect when creating a machine learning model. Data could be collected from one or multiple sources, linked altogether, cleaned and then preprocessed before being operated in the models. This chapter presents the procedure which has led to the creation of the database. Therefore the database is analyzed, and the importance of the features is discussed. GIRO provided the data used in this project, which were extracted from the APTS systems installed by the STM in their fleet. They concern the period of the Autumn 2017 and Autumn 2016 for the buses in Montreal. As the data from Autumn 2016 were only used for a specific analysis, it was not investigated during the data analysis. However, the methodology to build it was the same as for Autumn 2017.

3.1 Planning Data

In public transportation, the planning is a document which displays the details about the trips scheduled, which means the departure time for the different lines. It also arranges the trips in sequence, called blocks. Moreover, it specifies the time of departure from the depot, the return to the depot and the HLP.

The schedule for Autumn 2017 was operated from the 28th of August until the 29th October, for the weekdays and except the 4th of September and 9th October which were public holidays. In total, data from 43 days composes this database. Furthermore, 19 355 trips were scheduled for each weekday in Montreal during the described period. Hence the Autumn 2017 schedule can be used to describe 831 405 unique trips.

Figure 3.1 shows an example of data from the planning provided by GIRO.



Ligne	De	Début	Fin	À	NoPermanent	Voiture	Type	Voiture1	BatMinFinVoy	BatFin	DéficBat%	Via	Direction
	591	01/01/2000 05:42	01/01/2000 06:00	O23601	165997523	10 - 03	Sortie	10 - 03	0	0			
178	O23601	01/01/2000 06:00	01/01/2000 06:18	V04802	111227617	10 - 03	Régulier	10 - 03	2	0			Sud
	V04802	01/01/2000 06:18	01/01/2000 06:20	V04801	165971192	10 - 03	HLP	10 - 03	0	2	0		
178	V04801	01/01/2000 06:22	01/01/2000 06:34	O23602	111227655	10 - 03	Régulier	10 - 03	4	0		01	Nord
	O23602	01/01/2000 06:34	01/01/2000 06:39	O23601	165971193	10 - 03	HLP	10 - 03	0	11	0		
178	O23601	01/01/2000 06:50	01/01/2000 07:08	V04802	111227620	10 - 03	Régulier	10 - 03	2	0			Sud
	V04802	01/01/2000 07:08	01/01/2000 07:10	V04801	165971194	10 - 03	HLP	10 - 03	0	10	0		
178	V04801	01/01/2000 07:20	01/01/2000 07:32	O23602	111227659	10 - 03	Régulier	10 - 03	4	0		01	Nord
	O23602	01/01/2000 07:32	01/01/2000 07:37	O23601	165971195	10 - 03	HLP	10 - 03	0	6	0		
178	O23601	01/01/2000 07:43	01/01/2000 08:03	V04802	111227625	10 - 03	Régulier	10 - 03	2	0			Sud
	V04802	01/01/2000 08:03	01/01/2000 08:13	V050	165971196	10 - 03	HLP	10 - 03	0	5	0		
168	V050	01/01/2000 08:18	01/01/2000 08:58	M14001	162434071	10 - 03	Régulier	10 - 03	4	0			Nord
	M14001	01/01/2000 08:58	01/01/2000 09:15	M15001	165971197	10 - 03	HLP	10 - 03	0	4	0		
10	M15001	01/01/2000 09:19	01/01/2000 09:44	H052	80018211	10 - 03	Régulier	10 - 03	3	3	0		Nord
10	H052	01/01/2000 09:47	01/01/2000 10:15	M15001	80018217	10 - 03	Régulier	10 - 03	3	8	0		Sud
10	M15001	01/01/2000 10:23	01/01/2000 10:47	H052	80018214	10 - 03	Régulier	10 - 03	2	3	0		Nord
10	H052	01/01/2000 10:50	01/01/2000 11:18	M15001	80018220	10 - 03	Régulier	10 - 03	3	6	0		Sud
10	M15001	01/01/2000 11:24	01/01/2000 11:46	H052	79994762	10 - 03	Régulier	10 - 03	2	6	0		Nord
10	H052	01/01/2000 11:52	01/01/2000 12:20	M15001	80018222	10 - 03	Régulier	10 - 03	3	5	0		Sud
10	M15001	01/01/2000 12:25	01/01/2000 12:49	H052	79994763	10 - 03	Régulier	10 - 03	2	5	0		Nord
10	H052	01/01/2000 12:54	01/01/2000 13:25	M15001	80018225	10 - 03	Régulier	10 - 03	3	3	0		Sud
10	M15001	01/01/2000 13:28	01/01/2000 13:52	H052	79994760	10 - 03	Régulier	10 - 03	2	4	0		Nord
10	H052	01/01/2000 13:56	01/01/2000 14:28	M15001	80018227	10 - 03	Régulier	10 - 03	3	25	0		Sud
10	M15001	01/01/2000 14:53	01/01/2000 15:23	H052	79994758	10 - 03	Régulier	10 - 03	3	3	0		Nord
10	H052	01/01/2000 15:26	01/01/2000 16:01	M15001	80018231	10 - 03	Régulier	10 - 03	9	13	0		Sud
10	M15001	01/01/2000 16:14	01/01/2000 16:44	H052	25294114	10 - 03	Régulier	10 - 03	3	3	0		Nord
10	H052	01/01/2000 16:47	01/01/2000 17:23	M15001	80018235	10 - 03	Régulier	10 - 03	9	9	0		Sud
10	M15001	01/01/2000 17:32	01/01/2000 18:02	H052	85650280	10 - 03	Régulier	10 - 03	3	0			Nord
	H052	01/01/2000 18:02	01/01/2000 18:22	592	165997524	10 - 03	Entrée	10 - 03	0				

Figure 3.1 Example of Planning file

The documents were available under several formats, and the format Comma-separated values (CSV) was used and transformed under XLSX files. The different elements available in the planning files are :

- the id of the line operated. The data can be missing if the bus is operating a return to the depot or an HLP for example. For the second record of the previous figure, the line ID is "178".
- the id of the origin of the trip. For the second record of the previous figure, the origin is "O23601".
- the scheduled start time of the trip. For the second record of the previous figure, the start time is "06 :00".
- the scheduled end time of the trip. For the second record of the previous figure, the scheduled end time ID is "06 :18".
- the id of the destination of the trip. For the second record of the previous figure, the Id of the destination is "V04802".
- the permanent number. This number is specific to each trip, and thus be considered as a primary key because it identifies uniquely every trip. For the second record of the previous figure, the permanent number is "111227617".
- the block id. For the second record of the previous figure, the block ID is "10 - 03".
- the type of the trip. It can be either "regular" when it the bus operates a line, or "exit" when the bus leaves the depot, or "return" when the bus comes back to the depot or "HLP" which are the bobtails. For the second record of the previous figure, the trip is "Regular".
- the minimum layover time in minutes. For the second record of the previous figure, the minimum layover time is "2".
- the actual layover. For the second record of the previous figure, the minimum layover time is "0".
- the shortfall layover, which is different from 0 if the actual layover is lower than the minimum layover. However, this situation does not exist in our database. For the second record of the previous figure, the shortfall layover time is not indicated.
- the deviation, because buses can follow different paths for the same line, for example for a special time of the day. For the second record of the previous figure, the deviation is not indicated, which means that the bus follows the classic path.
- the direction, which can be one of the four cardinal directions. For the second record of the previous figure, the deviation "South".

The layover is not always indicated because if an HLP follows a regular trip, the layover will be transferred at the end of the HLP. The minimum layover time comes from the 95% percentile method, which means that if the minimum layover is satisfied, at least 95% of the trips would start on-time. However, this method is not very robust.

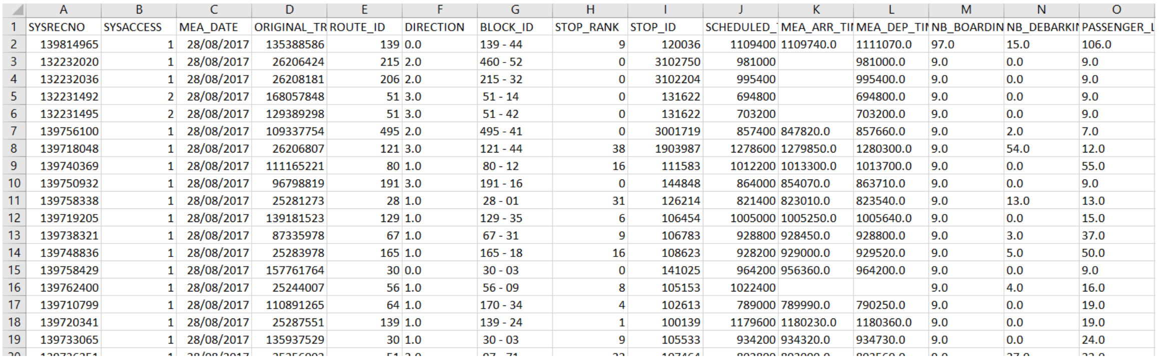
One trip is unique for each day, and its primary key is the permanent number. During the project, it was slightly modified and called OTN. A primary key composed of the day number aggregated to the permanent number was created to identify each trip in our database uniquely. The id created was named Unique Original Trip Number (UOTN).

3.2 APTS Data

The APTS data provided by the STM data center are a mix of AVL and APC data. They were recorded via a system called SCAD. In the data furnished, manually collected data were present but were removed.

3.2.1 Raw data

The gathered data were transferred upon a CSV format. One record gave the information of a bus making a halt at a bus stop, and various precise details such as the actual time of arrival and the number of people boarding and leaving the bus. Figure 3.2 is an example of a file provided :



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SYSRECNO	SYSACCESS	MEA_DATE	ORIGINAL_TR	ROUTE_ID	DIRECTION	BLOCK_ID	STOP_RANK	STOP_ID	SCHEDULED_	MEA_ARR_TI	MEA_DEP_TI	NB_BOARDIN	NB_DEBARKI	PASSENGER_L
2	139814965	1	28/08/2017	135388586	139	0.0	139 - 44	9	120036	1109400	1109740.0	1111070.0	97.0	15.0	106.0
3	132232020	1	28/08/2017	26206424	215	2.0	460 - 52	0	3102750	981000		981000.0	9.0	0.0	9.0
4	132232036	1	28/08/2017	26208181	206	2.0	215 - 32	0	3102204	995400		995400.0	9.0	0.0	9.0
5	132231492	2	28/08/2017	168057848	51	3.0	51 - 14	0	131622	694800		694800.0	9.0	0.0	9.0
6	132231495	2	28/08/2017	129389298	51	3.0	51 - 42	0	131622	703200		703200.0	9.0	0.0	9.0
7	139756100	1	28/08/2017	109337754	495	2.0	495 - 41	0	3001719	857400	847820.0	857660.0	9.0	2.0	7.0
8	139718048	1	28/08/2017	26206807	121	3.0	121 - 44	38	1903987	1278600	1279850.0	1280300.0	9.0	54.0	12.0
9	139740369	1	28/08/2017	111165221	80	1.0	80 - 12	16	111583	1012200	1013300.0	1013700.0	9.0	0.0	55.0
10	139750932	1	28/08/2017	96798819	191	3.0	191 - 16	0	144848	864000	854070.0	863710.0	9.0	0.0	9.0
11	139758338	1	28/08/2017	25281273	28	1.0	28 - 01	31	126214	821400	823010.0	823540.0	9.0	13.0	13.0
12	139719205	1	28/08/2017	139181523	129	1.0	129 - 35	6	106454	1005000	1005250.0	1005640.0	9.0	0.0	15.0
13	139738321	1	28/08/2017	87335978	67	1.0	67 - 31	9	106783	928800	928450.0	928800.0	9.0	3.0	37.0
14	139748836	1	28/08/2017	25283978	165	1.0	165 - 18	16	108623	928200	929000.0	929520.0	9.0	5.0	50.0
15	139758429	1	28/08/2017	157761764	30	0.0	30 - 03	0	141025	964200	956360.0	964200.0	9.0	0.0	9.0
16	139762400	1	28/08/2017	25244007	56	1.0	56 - 09	8	105153	1022400			9.0	4.0	16.0
17	139710799	1	28/08/2017	110891265	64	1.0	170 - 34	4	102613	789000	789990.0	790250.0	9.0	0.0	19.0
18	139720341	1	28/08/2017	25287551	139	1.0	139 - 24	1	100139	1179600	1180230.0	1180360.0	9.0	0.0	19.0
19	139733065	1	28/08/2017	135937529	30	1.0	30 - 03	9	105533	934200	934320.0	934730.0	9.0	0.0	24.0

Figure 3.2 Example of APTS file

The different details present for one record are :

- System Record Number, which is the id of the record. The first record of the example has a system record number of "139814965".
- System Access. The first record of the example has an access system Id of "1".

- Date. The first record of the example has a record data of "28/08/2017".
- Original Trip Number. The first record of the example has an original trip number of "135388586".
- Route id. The first record of the example has a route id of "139".
- Direction, which is an integer between 0 and 3. North is associated with 0, South to 1, East to 2 and West to 3. If a record has no direction associated, it is because the record is related to an HLP. The first record of the example has a direction of "0", which means "North".
- Block Id. The first record of the example has a block id of "139-44".
- Stop Rank. The first record of the example has a stop rank of "9", which means that the record corresponds to the 9th stops of a line.
- Stop id. The first record of the example has a stop id of "120036".
- Scheduled time. The first record of the example has a scheduled time of "1109400" in the STM time convention.
- Arrival Time. The first record of the example has an arrival time of "1109740" in the STM time convention.
- Departure Time. The first record of the example has a departure time of "1111070" in the STM time convention.
- Number of people boarding. The first record of the example has 97 people boarding.
- Number of people debarking. The first record of the example has 15 people debarking.
- Number of the passengers on the bus. The first record of the example has 106 passengers on the bus.
- The source of the record. The first record of the example has "SCAD" as a source, which means it was recorded via the APTS system.
- The location of the record. The first record of the example has a location of "120036", which is the same as the stop id.

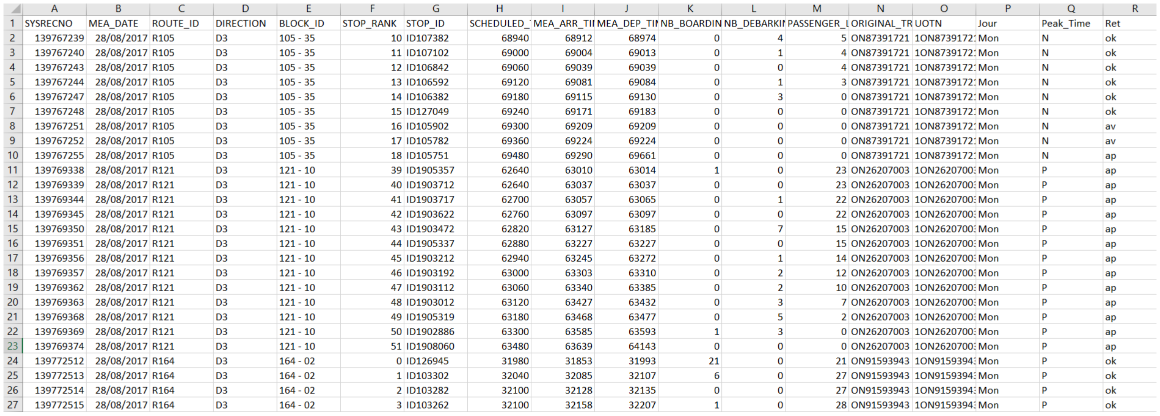
The format of some columns was modified in order to simplify further analysis. Moreover, some features were added to the database : the day of the week (from Monday to Friday), the part of the day or the UOTN associated to the record to link it later to planning data. The part of the day was decided as follow : "P" to describe peak hours, from 6 am to 9 am and from 3 pm to 6.30 pm, "T" for transition hours which are the 30 minutes before and after each peak hours, and finally "N" for the other period.

Also, the time was transformed in order to appear in a second format. The STM time convention gives the time tenth of second, with a day starting at 432 000. The first record which has a scheduled time of 1109400 could be converted in second :

$$T(s) = \frac{1109400 - 432000}{10} = 67740$$

Thus, 1109400 in STM time convention is 67 740 seconds, or 18 :49 :00, or 6.49 p.m.

The figure 3.3 is an example of the APTS file cleaned.

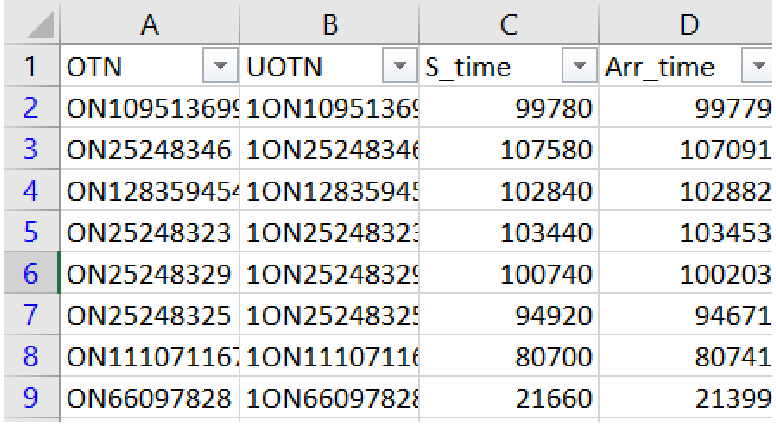


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	SYSRECNO	MEA_DATE	ROUTE_ID	DIRECTION	BLOCK_ID	STOP_RANK	STOP_ID	SCHEDULED_	MEA_ARR_TI	MEA_DEP_TI	NB_BOARDIN	NB_DEBARKI	PASSENGER_L	ORIGINAL_TR	UOTN	Jour	Peak_Time	Ret
2	139767239	28/08/2017	R105	D3	105 - 35	10	ID107382	68940	68912	68974	0	4	5	ON87391721	1ON8739172	Mon	N	ok
3	139767240	28/08/2017	R105	D3	105 - 35	11	ID107102	69000	69004	69013	0	1	4	ON87391721	1ON8739172	Mon	N	ok
4	139767243	28/08/2017	R105	D3	105 - 35	12	ID106842	69060	69039	69039	0	0	4	ON87391721	1ON8739172	Mon	N	ok
5	139767244	28/08/2017	R105	D3	105 - 35	13	ID106592	69120	69081	69084	0	1	3	ON87391721	1ON8739172	Mon	N	ok
6	139767247	28/08/2017	R105	D3	105 - 35	14	ID106382	69180	69115	69130	0	3	0	ON87391721	1ON8739172	Mon	N	ok
7	139767248	28/08/2017	R105	D3	105 - 35	15	ID127049	69240	69171	69183	0	0	0	ON87391721	1ON8739172	Mon	N	ok
8	139767251	28/08/2017	R105	D3	105 - 35	16	ID105902	69300	69209	69209	0	0	0	ON87391721	1ON8739172	Mon	N	av
9	139767252	28/08/2017	R105	D3	105 - 35	17	ID105782	69360	69224	69224	0	0	0	ON87391721	1ON8739172	Mon	N	av
10	139767255	28/08/2017	R105	D3	105 - 35	18	ID105751	69480	69290	69661	0	0	0	ON87391721	1ON8739172	Mon	N	ap
11	139769338	28/08/2017	R121	D3	121 - 10	39	ID1905357	62640	63010	63014	1	0	23	ON26207003	1ON2620700	Mon	P	ap
12	139769339	28/08/2017	R121	D3	121 - 10	40	ID1903712	62640	63037	63037	0	0	23	ON26207003	1ON2620700	Mon	P	ap
13	139769344	28/08/2017	R121	D3	121 - 10	41	ID1903717	62700	63057	63065	0	1	22	ON26207003	1ON2620700	Mon	P	ap
14	139769345	28/08/2017	R121	D3	121 - 10	42	ID1903622	62760	63097	63097	0	0	22	ON26207003	1ON2620700	Mon	P	ap
15	139769350	28/08/2017	R121	D3	121 - 10	43	ID1903472	62820	63127	63185	0	7	15	ON26207003	1ON2620700	Mon	P	ap
16	139769351	28/08/2017	R121	D3	121 - 10	44	ID1905337	62880	63227	63227	0	0	15	ON26207003	1ON2620700	Mon	P	ap
17	139769356	28/08/2017	R121	D3	121 - 10	45	ID1903212	62940	63245	63272	0	1	14	ON26207003	1ON2620700	Mon	P	ap
18	139769357	28/08/2017	R121	D3	121 - 10	46	ID1903192	63000	63303	63310	0	2	12	ON26207003	1ON2620700	Mon	P	ap
19	139769362	28/08/2017	R121	D3	121 - 10	47	ID1903112	63060	63340	63385	0	2	10	ON26207003	1ON2620700	Mon	P	ap
20	139769363	28/08/2017	R121	D3	121 - 10	48	ID1903012	63120	63427	63432	0	3	7	ON26207003	1ON2620700	Mon	P	ap
21	139769368	28/08/2017	R121	D3	121 - 10	49	ID1905319	63180	63468	63477	0	5	2	ON26207003	1ON2620700	Mon	P	ap
22	139769369	28/08/2017	R121	D3	121 - 10	50	ID1902886	63300	63585	63593	1	3	0	ON26207003	1ON2620700	Mon	P	ap
23	139769374	28/08/2017	R121	D3	121 - 10	51	ID1908060	63480	63639	64143	0	0	0	ON26207003	1ON2620700	Mon	P	ap
24	139772512	28/08/2017	R164	D3	164 - 02	0	ID126945	31980	31853	31993	21	0	21	ON91593943	1ON9159394	Mon	P	ok
25	139772513	28/08/2017	R164	D3	164 - 02	1	ID103302	32040	32085	32107	6	0	27	ON91593943	1ON9159394	Mon	P	ok
26	139772514	28/08/2017	R164	D3	164 - 02	2	ID103282	32100	32128	32135	0	0	27	ON91593943	1ON9159394	Mon	P	ok
27	139772515	28/08/2017	R164	D3	164 - 02	3	ID103262	32100	32158	32207	1	0	28	ON91593943	1ON9159394	Mon	P	ok

Figure 3.3 Example of APTS file cleaned

3.2.2 Extracting trip info

The topic of the research is to estimate end-trip delay for buses. Hence only one record for each trip was relevant : the last one. The last record of each trip was extracted via a Macro VBA. As the interest of the research is to work with planning, real-time data such as the number of people in the buses were not collected. The figure 3.4 shows a file treated with a Macro VBA. Lots of information were redundant with data present in the planning file, thus only the scheduled time and the arrival time were extracted, as well as the UOTN and the OTN in order to link the data. The times are in second.



	A	B	C	D
1	OTN	UOTN	S_time	Arr_time
2	ON109513699	1ON10951369	99780	99779
3	ON25248346	1ON25248346	107580	107091
4	ON128359454	1ON12835945	102840	102882
5	ON25248323	1ON25248323	103440	103453
6	ON25248329	1ON25248329	100740	100203
7	ON25248325	1ON25248325	94920	94671
8	ON11071167	1ON1107116	80700	80741
9	ON66097828	1ON66097828	21660	21399

Figure 3.4 Example of information extracted from APTS data

All the 43 files were treated with this method. Finally, 139 452 trips were collected. Moreover, as explained before, 19935 were scheduled each day, which implies that 831 405 trips could have been recorded. The difference is explained by the fact that only 20% of the buses were equipped with SCAD systems in 2017 [3]. 20% of the 831 405 scheduled trips is 166 281. Thus the number of trips collected is coherent.

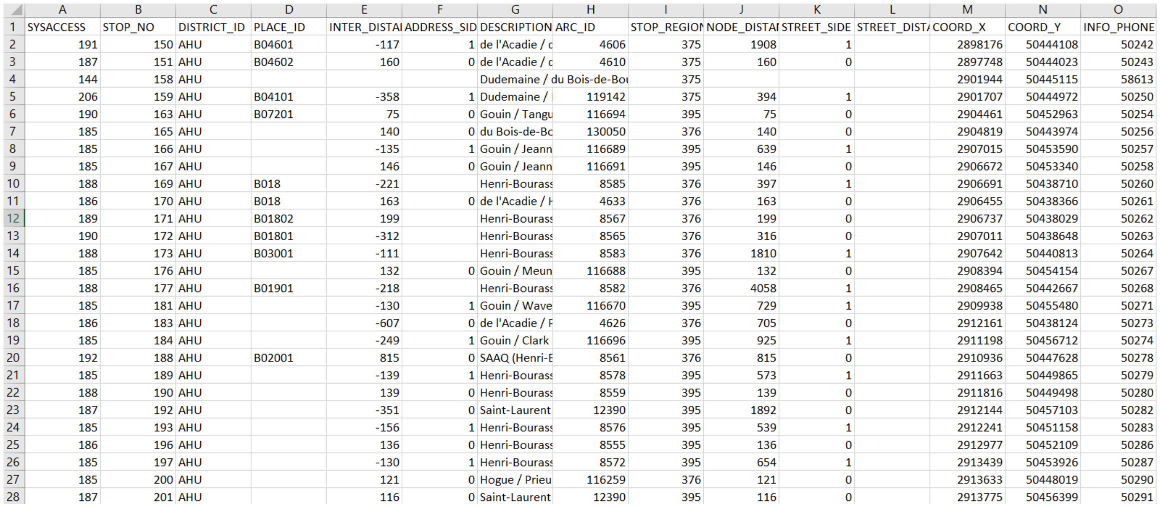
3.3 External information

Once the planning and APTS data were gathered, other sources of information were analyzed to add features to our model. New data had to be off-line information, as the objective of the project is to work with planning data and not in real-time. GIRO provided additional information involving stops information, driver schedules and layover time.

3.3.1 Stop info

The objective of having stop information was dual : on one hand to have geographic information for the terminus, and on the other hand to know the road taken by the different lines.

The file received was a CSV file composed of the stops id and their characteristics, as shown in the figure 3.5. The information used was in the columns districts and coordinate X and coordinate Y. A change in the format had to be done to normalize it. In order to link this data with the planning data, the place Id was used. The range of the value goes from 2,69 to 3.06 for X coordinates, and from 5,029 to 5,062 for the Y coordinates.

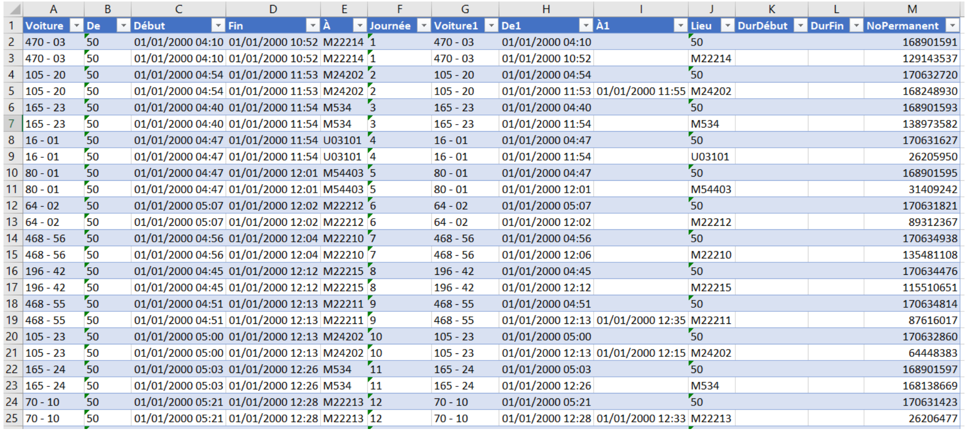


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SYSACCESS	STOP_NO	DISTRICT_ID	PLACE_ID	INTER_DISTAI	ADDRESS_SID	DESCRIPTION	ARC_ID	STOP_REGION	NODE_DISTAI	STREET_SIDE	STREET_DIST/	COORD_X	COORD_Y	INFO_PHONE
2	191	150	AHU	B04601	-117	1	de l'Acadie / c	4606	375	1908	1		2898176	50444108	50242
3	187	151	AHU	B04602	160	0	de l'Acadie / c	4610	375	160	0		2897748	50444023	50243
4	144	158	AHU				Dudemaine / du Bois-de-Bou		375				2901944	50445115	58613
5	206	159	AHU	B04101	-358	1	Dudemaine /	119142	375	394	1		2901707	50444972	50250
6	190	163	AHU	B07201	75	0	Gouin / Tangu	116694	395	75	0		2904461	50452963	50254
7	185	165	AHU		140	0	du Bois-de-Bc	130050	376	140	0		2904819	50443974	50256
8	185	166	AHU		-135	1	Gouin / Jeann	116689	395	639	1		2907015	50453590	50257
9	185	167	AHU		146	0	Gouin / Jeann	116691	395	146	0		2906672	50453340	50258
10	188	169	AHU	B018	-221		Henri-Bourass	8585	376	397	1		2906691	50438710	50260
11	186	170	AHU	B018	163	0	de l'Acadie / H	4633	376	163	0		2906455	50438366	50261
12	189	171	AHU	B01802	199		Henri-Bourass	8567	376	199	0		2906737	50438029	50262
13	190	172	AHU	B01801	-312		Henri-Bourass	8565	376	316	0		2907011	50438648	50263
14	188	173	AHU	B03001	-111		Henri-Bourass	8583	376	1810	1		2907642	50440813	50264
15	185	176	AHU		132	0	Gouin / Meun	116688	395	132	0		2908394	50454154	50267
16	188	177	AHU	B01901	-218		Henri-Bourass	8582	376	4058	1		2908465	50442667	50268
17	185	181	AHU		-130	1	Gouin / Wave	116670	395	729	1		2909938	50455480	50271
18	186	183	AHU		-607	0	de l'Acadie / F	4626	376	705	0		2912161	50438124	50273
19	185	184	AHU		-249	1	Gouin / Clark	116696	395	925	1		2911198	50456712	50274
20	192	188	AHU	B02001	815	0	SAAQ (Henri-E	8561	376	815	0		2910936	50447628	50278
21	185	189	AHU		-139	1	Henri-Bourass	8578	395	573	1		2911663	50449865	50279
22	188	190	AHU		139	0	Henri-Bourass	8559	395	139	0		2911816	50449498	50280
23	187	192	AHU		-351	0	Saint-Laurent	12390	395	1892	0		2912144	50457103	50282
24	185	193	AHU		-156	1	Henri-Bourass	8576	395	539	1		2912241	50451158	50283
25	186	196	AHU		136	0	Henri-Bourass	8555	395	136	0		2912977	50452109	50286
26	185	197	AHU		-130	1	Henri-Bourass	8572	395	654	1		2913439	50453926	50287
27	185	200	AHU		121	0	Hogue / Prieu	116259	376	121	0		2913633	50448019	50290
28	187	201	AHU		116	0	Saint-Laurent	12390	395	116	0		2913775	50456399	50291

Figure 3.5 Example of Stop file

3.3.2 Driver change

Driver schedules could be decided during the planning phase, and so could be integrated into the features. One driver can do several blocks in a working day, and several drivers could operate one block. In the figure 3.6, the id in the column "journée" means that a unique operator was operating all the trips having this id in characteristic. Having a change of driver during a block could imply delays if the new driver is late for different reasons.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Voiture	De	Début	Fin	À	Journée	Voiture1	De1	À1	Lieu	DurDébut	DurFin	NoPermanent
2	470 - 03	50	01/01/2000 04:10	01/01/2000 10:52	M22214	1	470 - 03	01/01/2000 04:10		50			168901591
3	470 - 03	50	01/01/2000 04:10	01/01/2000 10:52	M22214	1	470 - 03	01/01/2000 10:52		M22214			129143537
4	105 - 20	50	01/01/2000 04:54	01/01/2000 11:53	M24202	2	105 - 20	01/01/2000 04:54		50			170632720
5	105 - 20	50	01/01/2000 04:54	01/01/2000 11:53	M24202	2	105 - 20	01/01/2000 11:53	01/01/2000 11:55	M24202			168248930
6	165 - 23	50	01/01/2000 04:40	01/01/2000 11:54	M534	3	165 - 23	01/01/2000 04:40		50			168901593
7	165 - 23	50	01/01/2000 04:40	01/01/2000 11:54	M534	3	165 - 23	01/01/2000 11:54		M534			138973582
8	16 - 01	50	01/01/2000 04:47	01/01/2000 11:54	U03101	4	16 - 01	01/01/2000 04:47		50			170631627
9	16 - 01	50	01/01/2000 04:47	01/01/2000 11:54	U03101	4	16 - 01	01/01/2000 11:54		U03101			26205950
10	80 - 01	50	01/01/2000 04:47	01/01/2000 12:01	M54403	5	80 - 01	01/01/2000 04:47		50			168901595
11	80 - 01	50	01/01/2000 04:47	01/01/2000 12:01	M54403	5	80 - 01	01/01/2000 12:01		M54403			31409242
12	64 - 02	50	01/01/2000 05:07	01/01/2000 12:02	M22212	6	64 - 02	01/01/2000 05:07		50			170631821
13	64 - 02	50	01/01/2000 05:07	01/01/2000 12:02	M22212	6	64 - 02	01/01/2000 12:02		M22212			89312367
14	468 - 56	50	01/01/2000 04:56	01/01/2000 12:04	M22210	7	468 - 56	01/01/2000 04:56		50			170634938
15	468 - 56	50	01/01/2000 04:56	01/01/2000 12:04	M22210	7	468 - 56	01/01/2000 12:06		M22210			135481108
16	196 - 42	50	01/01/2000 04:45	01/01/2000 12:12	M22215	8	196 - 42	01/01/2000 04:45		50			170634476
17	196 - 42	50	01/01/2000 04:45	01/01/2000 12:12	M22215	8	196 - 42	01/01/2000 12:12		M22215			115510651
18	468 - 55	50	01/01/2000 04:51	01/01/2000 12:13	M22211	9	468 - 55	01/01/2000 04:51		50			170634814
19	468 - 55	50	01/01/2000 04:51	01/01/2000 12:13	M22211	9	468 - 55	01/01/2000 12:13	01/01/2000 12:35	M22211			87616017
20	105 - 23	50	01/01/2000 05:00	01/01/2000 12:13	M24202	10	105 - 23	01/01/2000 05:00		50			170632860
21	105 - 23	50	01/01/2000 05:00	01/01/2000 12:13	M24202	10	105 - 23	01/01/2000 12:13	01/01/2000 12:15	M24202			64448383
22	165 - 24	50	01/01/2000 05:03	01/01/2000 12:26	M534	11	165 - 24	01/01/2000 05:03		50			168901597
23	165 - 24	50	01/01/2000 05:03	01/01/2000 12:26	M534	11	165 - 24	01/01/2000 12:26		M534			168138669
24	70 - 10	50	01/01/2000 05:21	01/01/2000 12:28	M22213	12	70 - 10	01/01/2000 05:21		50			170631423
25	70 - 10	50	01/01/2000 05:21	01/01/2000 12:28	M22213	12	70 - 10	01/01/2000 12:28	01/01/2000 12:33	M22213			26206477

Figure 3.6 Example of Driver Schedule

3.3.3 Mean Layover Prev

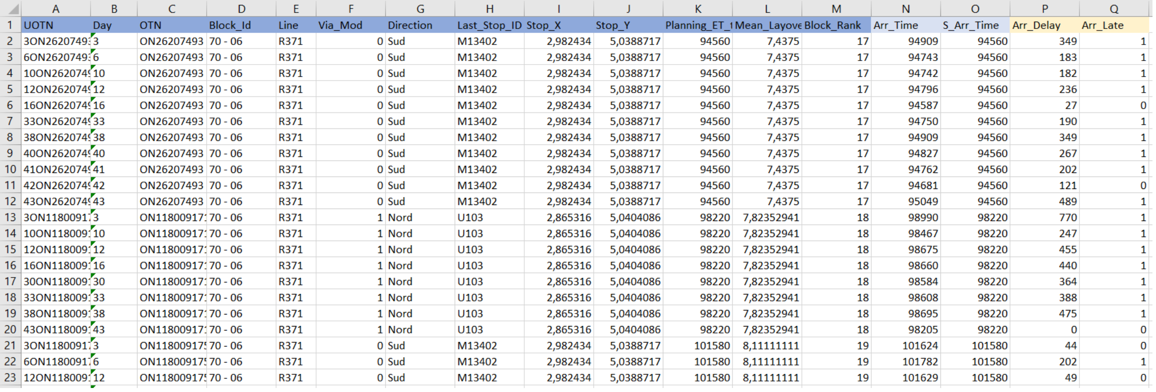
The layover time is a buffer time which can prevent the delay propagation. Hence it is interesting to have information about all the previous layovers. Thus the mean layover of the previous trip was calculated. A sum of all the layovers could have been done, but the information would have been correlated to the block rank. When the trip was the first of the block, instead of letting a blank, a decision was made to choose 20 minutes, which is a high value, similar to a one which would prevent delay propagation. This choice emphasizes the fact that there is no delay propagation for the first trip of a block.

3.3.4 Day of the week

The information about the day of the week was also extracted. Only the weekdays are present.

3.3.5 Creating the final database

Finally, all the information were gathered in a final database, and the followinf figure gives an overview.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	UOTN	Day	OTN	Block_Id	Line	Via_Mod	Direction	Last_Stop_ID	Stop_X	Stop_Y	Planning_ET_	Mean_Layove	Block_Rank	Arr_Time	S_Arr_Time	Arr_Delay	Arr_Late
2	3ON2620749	3	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94909	94560	349	1
3	6ON2620749	6	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94743	94560	183	1
4	10ON262074	10	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94742	94560	182	1
5	12ON262074	12	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94796	94560	236	1
6	16ON262074	16	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94587	94560	27	0
7	33ON262074	33	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94750	94560	190	1
8	38ON262074	38	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94909	94560	349	1
9	40ON262074	40	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94827	94560	267	1
10	41ON262074	41	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94762	94560	202	1
11	42ON262074	42	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	94681	94560	121	0
12	43ON262074	43	ON26207493	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	94560	7,4375	17	95049	94560	489	1
13	3ON1180091	3	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98990	98220	770	1
14	10ON118009	10	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98467	98220	247	1
15	12ON118009	12	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98675	98220	455	1
16	16ON118009	16	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98660	98220	440	1
17	30ON118009	30	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98584	98220	364	1
18	33ON118009	33	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98608	98220	388	1
19	38ON118009	38	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98695	98220	475	1
20	43ON118009	43	ON11800917	70 - 06	R371	1	Nord	U103	2,865316	5,0404086	98220	7,82352941	18	98205	98220	0	0
21	3ON1180091	3	ON11800917	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	101580	8,11111111	19	101624	101580	44	0
22	6ON1180091	6	ON11800917	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	101580	8,11111111	19	101782	101580	202	1
23	12ON118009	12	ON11800917	70 - 06	R371	0	Sud	M13402	2,982434	5,0388717	101580	8,11111111	19	101629	101580	49	0

Figure 3.7 Example of Final Database file

3.4 Preprocessing

Once the database is created, a preprocessing work was done to prevent unreliable information to be part of the database. Indeed, it is essential to have all the information for each record to afford machine learning models to find more accurate patterns between the data.

3.4.1 Missing data

When the stop information was linked to the database, the coordinates of ten different stops present in the database were missing. These stops were extracted and analyzed by the routes they come from. It appears that because of the deviation of some routes, the end stop has a slightly different id and thus cannot be linked to the stop database. The id replacement was made, and the missing data were incorporated.

Also, 26 records from APTS data had UOTN which could not be linked with data from the planning. It represented four different OTN. Therefore these data were removed from the database. It was supposed that these trips were not planned initially and were added for special events or in support of public transport malfunction.

At this step, the database was composed of 139 426 records.

3.4.2 Database cleaning

To be sure the data from the APTS sources match precisely the planning data, the scheduled time for the last stop was also extracted with the macro VBA. It was found that 5176 extracted data had a scheduled arrival time which did not match with the one presents in the planning database. The explanation was possible malfunctions of the APTS devices, which could have stopped working at some points and thus not recording the last stops of a trip. As it represented less than 5% of the data, it was decided to remove these records from the database.

At this step, the database was composed of 134 250 records.

3.4.3 Remove the lines of less than 100 occurrences

In order to avoid to have a too unbalanced database, a decision was made to remove the records concerning lines with too few occurrences. Indeed, even if some methods had been developed in order to struggle with this issue, it is always better to have a balanced database. Figure 3.8 shows the number of occurrences for each line.

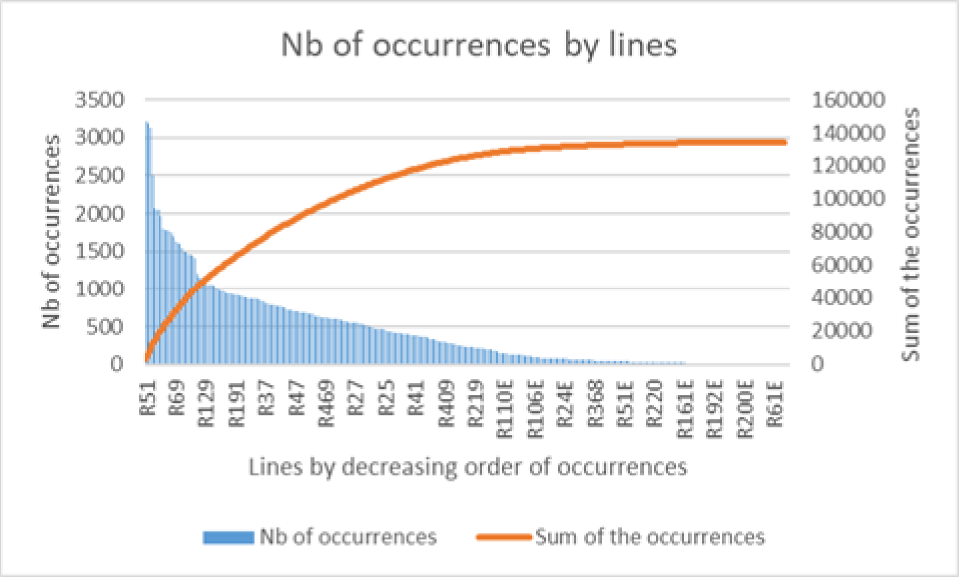


Figure 3.8 Number of records for each line

It was decided arbitrarily that trip describing lines with less than 100 occurrences would be removed from the database. Then, 3 769 records coming from 119 lines were removed. The remaining data reflected 184 lines. As it will be explained later in this thesis, the line feature is preprocessed with the one-hot method before being incorporated in machine learning models, and so decreasing the number of lines will improve the accuracy of the models and decrease the training time.

At this step, the database was composed of 130 481 records.

3.4.4 Conclusion of preprocessing

After these three steps, on the database of 139 452 records, 130 481 were used, which means that 9.4% of the data were removed. The table 3.1 synthesis the different steps.

Table 3.1 Synthesis of preprocessing

Step	Comments	Number of records in the database
0	Database initial	139 452
1	Missing Data	139 426
2	Unmatched Data	134 250
3	Lines with few occurrences	130 481

3.5 Database Analysis

Once the different features incorporated in the database, an analysis of the database was made.

3.5.1 Features Analysis

The different features of the database are synthesized in the table 3.2.

Table 3.2 Synthesis of the features

Planning Data	Other Data
Line (integer)	Last Stop X (continuous)
Block Rank (integer)	Last Stop Y (continuous)
Mean Layover Prev (continuous)	Driver Change (binary)
Scheduled Time (integer)	Week Day (integer)
Via (binary)	
Direction (integer)	

The next sections will describe more precisely each feature.

Line feature

Figure 3.9 shows the partition after the preprocessing explained in the previous section.

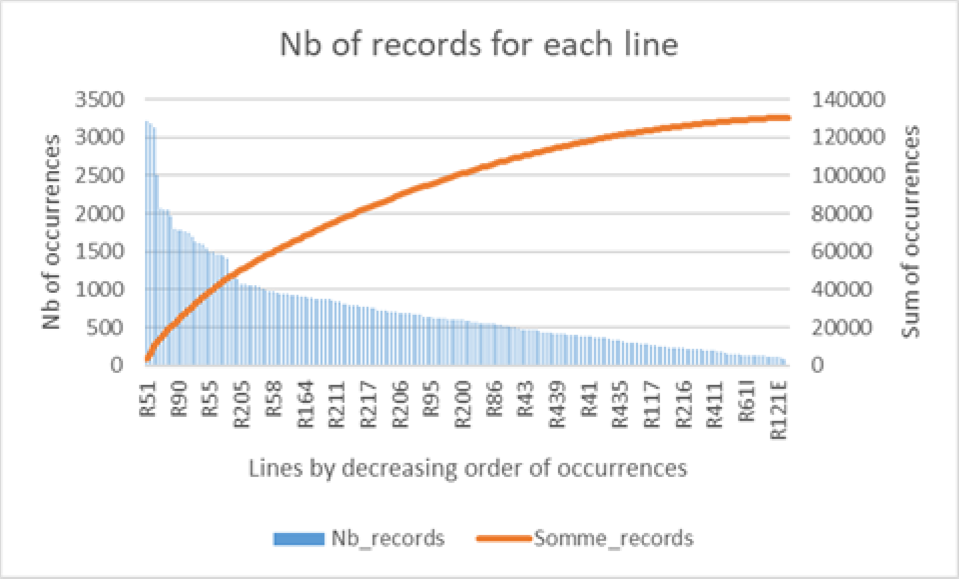


Figure 3.9 Number of records by line after preprocessing

The partition function is similar to a Pareto one.

Block Rank Feature

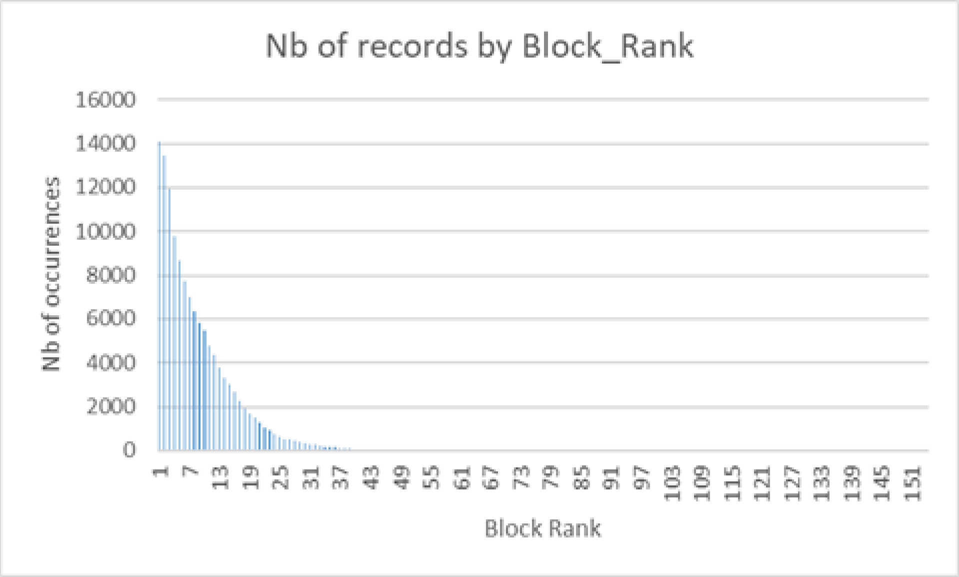


Figure 3.10 Number of records by block rank

The number of observation decreases when the block rank gets higher because if a trip is recorded, all the previous trips of the same block are also recorded.

Mean of the Previous Layovers feature

For each trip, the mean of the previous layovers of its block was calculated, except for the first trip of the block whose value was set to 20. The feature is continuous, and so to describe it, the different values were integrated into a class, calculated by making a superior rounding. For example, if a value was calculated at 3.2, the value was incorporated in class 4.

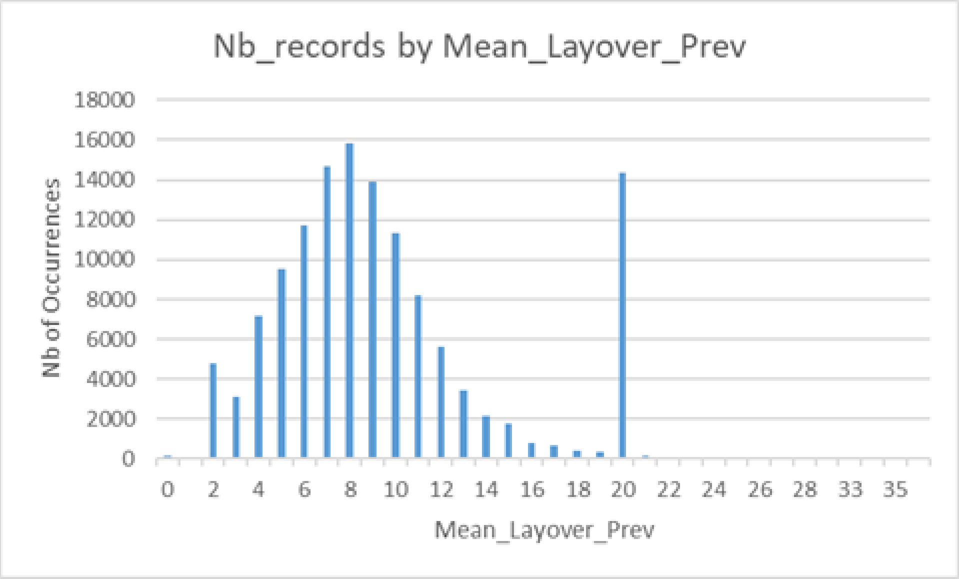


Figure 3.11 Number of records by Mean Layover Prev

The majority of the records have a mean layover prev included between 5 and 10 minutes. A large number of records with a 20 mean previous layover is explained by the fact that the mean layover previous feature had been set at 20 for the first trip of the blocks.

Scheduled time feature

As the feature is continuous, the values were gathered by time slots, which symbolize periods of 30 minutes. However, the values were integrated with a normalizing preprocessor in the machine learning models.

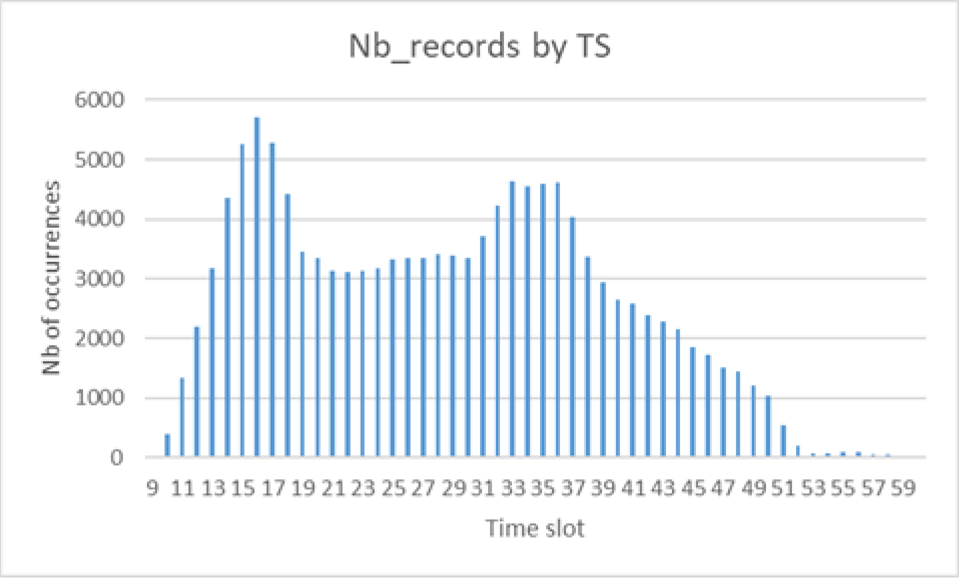


Figure 3.12 Number of records by time slot

They are two peak-time periods by day : one from 6 am to 9 am and the other from 3 pm to 6.30pm. They correspond to the time slot from 12 to 18 and from 30 to 37. These time slots are effectively the most represented.

Via feature

The feature is one if the line does not take the usual road, and 0 if it takes the usual one.

Table 3.3 Number of records Via

Via	Number of records
0	117 230
1	13 251

Around 10% of the trips recorded have taken a via road.

Direction feature

The possible directions are the different cardinal point.

Table 3.4 Number of records Direction

Direction	Number of records
East	44 128
North	19 488
West	46 513
South	20 350

The majority of the trips are made in the east/west axis. The number of trip north and south have roughly the same number of records, as for the east and west direction.

Stop feature

The easiest way of representing the stop features was not to use the stop coordinates but the stop id.

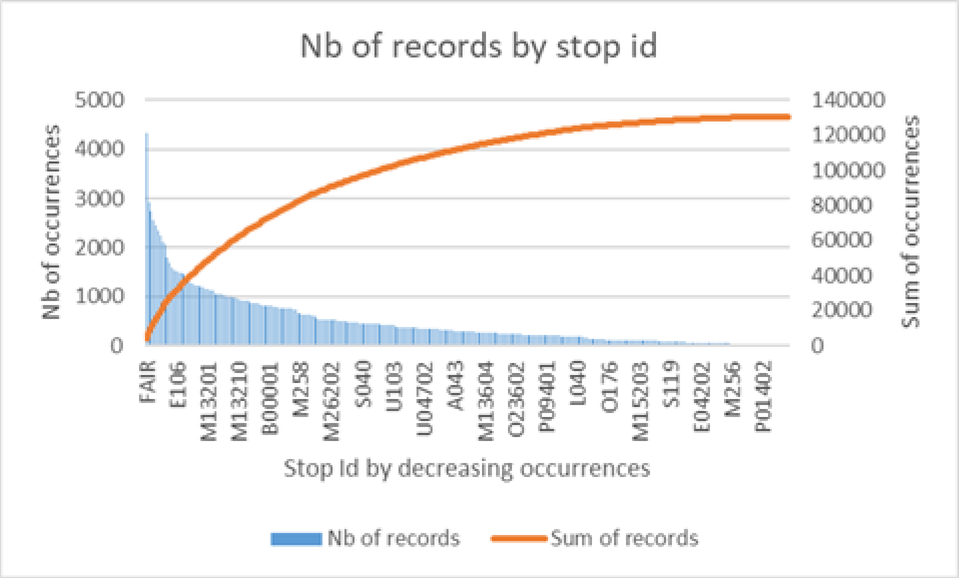


Figure 3.13 Number of records by Stop Id

The distribution is similar to the line distribution. There is less than twice the number of lines because several lines use the same terminus.

Driver change feature

The feature is one if the driver changes at the beginning of the trip inside a block. A driver can’t be changed in the middle of a trip.

Table 3.5 Number of records Driver change

Driver Change	Number of records
0	128 833
1	1 648

The feature take the value of 1 when the driver change within the block, which means that the feature does not show that a new driver starts a block.

Day of the week

Table 3.6 Number of records by days

Day	Number of records
Monday	20 808
Tuesday	27 474
Wednesday	27 468
Thursday	26 850
Friday	27 881

Each day is present nine times in the database, except for Monday which is present only seven times. Indeed the two public holidays removed from the database took place on Mondays. All the days have around 27 000 occurrences, and seven-ninth correspond to 21 000, which is coherent with the data.

Conclusion

For some features, the data can be heavily unbalanced, which explains the importance of oversampling during the building of the Machine learning algorithm. This part of the thesis discussed the general characteristics of the features but not about their selection and their importance. These choices were made later by analyzing the results of the different Machine learning algorithms.

3.5.2 Features Correlation

The correlation method analyzes the relation between the pair of variables. Features should not be correlated to each other because it biases machine learning models. Figure 3.14 show the correlation matrix between the features. A negative correlation is shown in green and an absence of correlation in white and high correlation in red.

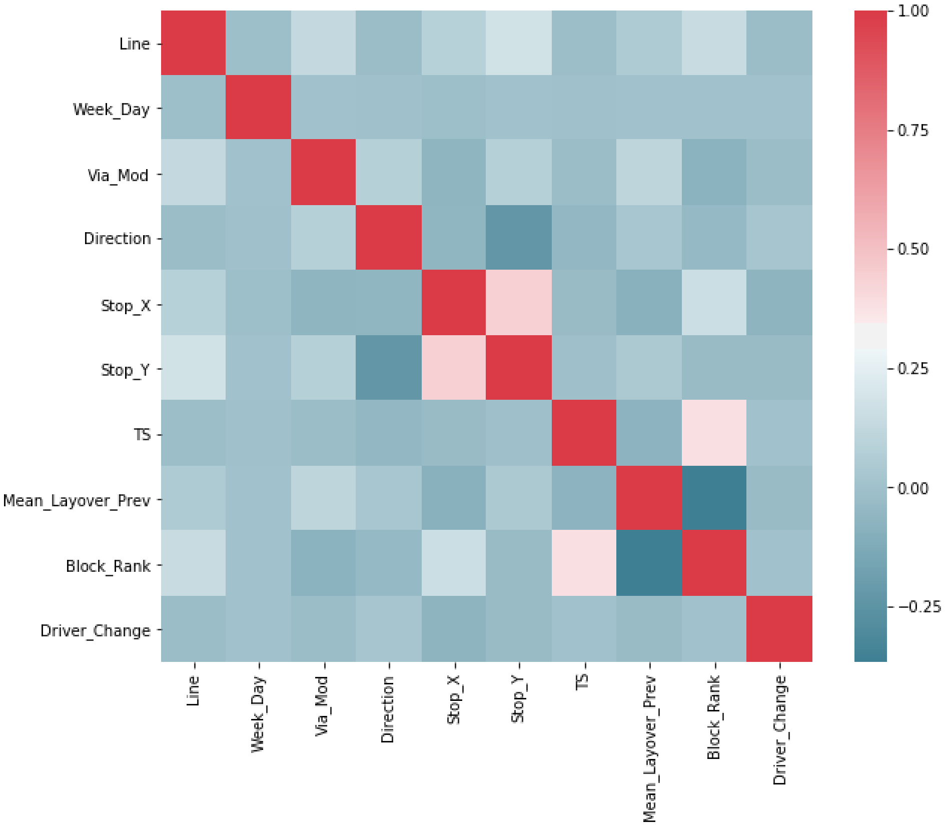


Figure 3.14 Correlation between the features

Stop X and Stop Y are correlated because they represent the same feature : the stop id. However, using the coordinates displays more information about the stops rather than using the id. It affords the models to find geographic patterns in the models.

However, the Stop coordinates and the direction are not correlated, and mainly the Y coordinate because most trips are made in the East-West direction. Moreover, the block rank and the mean layover prev are not correlated, and it can be explained because of the high variability of the mean value for the first trips of a block.

Most of the variables are not or little correlated to each other. That is why all the data were kept for the models to improve its performance. Normalization and auto-encoders will be used to regularise the data.

3.5.3 Objectives Analysis

This section will focus on the analysis of the objective : the bus end trip delays. This information was calculated by subtracting the actual measured time by APTS systems with the scheduled arrival time.

Objectives Partition

It was decided that a trip was considered late if it ends more than three minutes of delay or 180 seconds. Asserting that a bus is late when the delay is larger than 3 minutes is a standard in public transportation, and was confirmed by GIRO. The partition is as shown in 3.7.

Table 3.7 Number of records Late

End Status	Number of records	Percentage
On-time	94 339	72.3
Late	36 142	27.7

However, a more detailed report can be produced, adding more delay category : buses which ended before the scheduled time, delays less than three minutes, delays between three and ten minutes and the other. The partition would the following one :

Table 3.8 Number of records Late by category

$\text{Delay} < 0$	$0 \leq \text{Delay} < 180$	$180 \leq \text{Delay} < 600$	$600 \leq \text{Delay}$
61 610	32 608	27 432	8 710
47%	25%	21%	7%

When the previous trip ends late, the number of the bus arriving late rises to 44%. The propagation of the delay explains it. Moreover, the mean delay for the buses which end their trip late is 498 seconds in general, and it rises to 563 seconds when the previous trip has ended late. For these analyses the first trip of a block in order to have the information on the previous trip. The conclusion is that there are two main reasons for a bus ending late : either it had problems on its trip or it results from the previous trips which ended late.

3.6 Autumn 2016 database

As announced during the introduction of this section, the Autumn 2016 database will not be explicitly described. However, it will be compared to the Autumn 2017 database.

3.6.1 General description

For the Autumn 2016 database, 124 732 records were extracted from the APTS systems for the weekdays from the 29th August 2016 to the 28th October 2016, except Monday 5th September and Monday 10th October. The data were processed similarly to Autumn 2017, and only the lines present in Autumn 2017 were kept. After these steps, 114 092 records left for the machine learning models : 9.1% of the data were removed.

During Autumn 2016, 27.8% of the trips ended late, very similar to the percentage of Autumn 2017.

3.6.2 Comparison to the Autumn 2017 database

The three lines with the most occurrences are the same in both databases : "R51", "R105" and "R24".

The driver change feature was not available for this period. Moreover, three lines were present in Autumn 2017 and not in Autumn 2016.

3.7 Data Analysis conclusion

This section explained how the data was collected, gathered and processed. It was also explained how the problems of missing data or data format were treated.

An analysis of the features and the objectives showed that the features are mainly not correlated to each other, but correlated to the objective. The selection and integration of the data in the machine learning models will be treated in the next section.

CHAPTER 4 MODELLING BUS END TRIP DELAY

This section introduces the building of end-trip delay prediction models and their optimization. The first model predicts if a trip is going to end late, which means it would have more than three minutes of delay. The second one estimates the probability for each trip to end within different slots of delays. For both models, only offline data were used. Finally, the Autumn 2016 database is aggregated and the potential to estimate end trip delay on several periods is evaluated.

During the whole project, Python 3.7 was used for this work. The packages Numpy and Panda were used for respectively numeric applications and data treatment. Machine learning models were built with the SKlearn package. All the models had been created on the server RossoVerde of the chair with :

- 2X Intel Xeon E5-2650V4 @ 2.2GHZ, 12 cores
- 256GB RAM
- GPU Nvidia Titan Xp Pascal 12Gb Gddr5 G

4.1 General Overview

This first section will describe the framework of the modeling, describing the general structure of the algorithms used : the preprocessing of the data, the machine learning algorithms, and the output results. The metrics used during the project are defined in this section as well.

4.1.1 Methodology

The different algorithms follow the same different steps, which are described in this section.

Reading Data

The first step of the algorithm is to transfer the data from the original files to the scripts. The original database had been put under a JSON format to speed up this step.

Preprocessing

The data are therefore preprocessed. First, a label encoder transforms the data which are not numerical : the lines, the weekday and the direction are affected to numbers. Then through the use of a pipeline, the numerical data are standardized. The categorical features which are not binary are transformed via a one-hot method. The one-hot methods transform a categorical feature in several features equivalent to the different state of the primordial feature. Then for each record, it affects 1 to the corresponding feature and 0 to the other. It is often used when the different stated have not an order relation between them.

For example, for the feature weekday, the different states are Monday, Tuesday, Wednesday, Thursday and Friday. After the label encoder step, the different states are 1, 2, 3, 4 and 5. For example, a record occurring a Wednesday would have the one-hot method transforms it in a vector [0, 0, 1, 0, 0].

Splitting the data

The data are then split into different sets. Ten percent of the whole database is put apart for the test results, and the ninety percent are for the training set. Then ten percent of the training set is extracted for the validation set, and the rest are kept for the training. The module *train_test_split* was used. The concern of this step is to provide general results for the models because sets tested have not participated in the training phases.

Oversampling the data

The train set is unbalanced, as discussed in the data analysis section. In order to improve the accuracy of models with an unbalanced database, it is essential to oversample [64] the train set in order to recall more the underrepresented categories. Different techniques of oversampling were tried and are discussed later.

Tuning the hyper-parameters and training the model

The algorithms can be tuned with different parameters. In this step, models with different hyper-parameters are created and trained on the train set. Then they are tested on the validation test and compared to all the other. The metrics within which they are compared are discussed in the next section. The model which got the best score on the validation set is kept.

Results

The model with the best hyper-parameters is tested on the test set.

Finally and depending on the analysis, output files are produced : confusion matrices or excel files for example.

4.1.2 Metrics

The choice of metrics to evaluate a machine learning algorithm is decisive. Different metrics are used to compare the different aspects of the performance of the algorithms. Thus, it influences a lot the final results.

Before presenting different metrics, the definition of the concepts needs to be done. P is the set of the positive elements and N the set of the negative ones. A True Positive (TP) is a positive classified as a positive, a True Negative (TN) is a negative correctly classified, a False Positive (FP) is a negative classified as positive, and a False Negative (FN) is a positive classified as negative [65].

Accuracy

The accuracy A is simply the fraction of all the instances rightfully labeled. It is the easiest way of assessing a machine learning model. A perfect model would have an accuracy of 1.

$$A = \frac{TP + TN}{N + P}$$

However, for an unbalanced database, the accuracy can be very high by just predicting everything on the dominant class. Thus accuracy is excellent even if the model did not learn anything. However, the objective of some problems, especially in the medical field, is to detect anomalies.

Precision

The precision P is the proportion of true positives over all the predicted positives.

$$P = \frac{TP}{PredictedPositive}$$

Recall

The recall R metric, also called positive predictive value, is the ratio of the positive value classified as is over the number of positive value to be classified.

$$R = \frac{TP}{P}$$

The recall metric can be relevant in the case of the unbalanced database. Both accuracy and recall should be taken into account ; hence metrics combining them both were studied.

ROC

Receiver operating characteristic (ROC) is a metric used for binary classification algorithms and represents the faculty of a model to choose between negative and positive classes. It plots the recall for positive and for negative class depending on different thresholds [1]. The ROC value is the area under the ROC curve and can be up to 1.

The first experiments were done with this metric, but a model with a best ROC value did not imply that it will be the best at a fixed threshold. Also, in our experiments, the threshold was set at 0.5.

F1 Score

The F1 score is the harmonic mean of precision and recall, and thus combines precision and recall.

$$F1 = 2\frac{P * R}{P + R} = 2\frac{2 * TP}{2 * TP + FP + FN}$$

This measure is the one used to compare the different algorithms. For the binary classifier algorithms, the f1 score was the one calculated with the accuracy and the recall of the delay class. For multi-label classifier, the F1 score was the weighted average of the f1 score.

Confusion Matrix

A confusion matrix is not directly a performance metric, but a convenient way of visualizing the results of a binary classification algorithm. It shows the number of TP, TN, FP and FN in a matrix with a variation of colors depending on the importance of the number.

Normalized confusion matrices can also be displayed, in which of instead of having the number of occurrences for each part of the matrix, it has the ratio of the occurrences of the number of total positive or total negative depending on the category.

[1] A threshold is a numeric limit at which a probability for being into a class is above this limit, it is decided to assign this class to this input. By default, a threshold is set at 0.5

4.1.3 Building the different algorithms

As presented in the previous sections, three different machine learning models will be tested through this work : random forest, gradient boosted tree and artificial neural network. The following sections will describe for each algorithm the different hyperparameters which could be tuned and the used range of the tuning. Moreover, these models were compared to logistic regression.

For all of these models, a random state was applied in order to get the same results for different runs of the script. The stopping criteria was set by default with a toll of 10e-4 for the last ten iterations.

Decision Tree

As the gradient boosted tree and the random forest algorithms are based on decision trees, these algorithms should be detailed. A decision tree is a tree-like model composed of nodes (at which different paths are offered depending on the value of some feature) and of leaves (which indicate for each class its probability). The trees build themselves in order to minimize the entropy of the leaves.

Random Forest

A random forest is an estimator which fits a decided number of trees with sub-portion of the database and make them vote for the final results in order to limit the over-fitting and improve precision. Random forest classifiers were used because of their speed of training and the great results they produced. The number of estimators was set to 500, and the only parameter which was tuned was the maximum depth of the forests [66].

Gradient Boosted Tree

Gradient Boosted tree is a new estimator which used decision trees in order to make the prediction. For each stage, several regression trees equivalent to the number of classes are fit on the errors made by the previous decision trees. However, this model has a high training time. For this model, the only parameters which were tuned were the learning rate, which was tried with 0.1 and 0.01, and the maximum depth of the tree within a range of 5 to 500 [67].

Artificial Neural Network

One axiom of Machine Learning is that large multi-layer perceptron can describe any mathematical function. Artificial neural networks are composed of layers of perceptron interconnected to each other, and whose weight adjust through the learning. Even if this algorithm is the most suitable to have the best result, it is also the most difficult to tune. The artificial neural network tested is the multi-layer perceptron. The ANN have multiple parameters to tune [68] :

- hidden layer size : a range from the half to the double of the number of features were tested
- the activation function was chosen between 'identity' which returns x, 'logistic' which returns the sigmoid function and 'relu', which returns the rectified linear unit function

The chosen solver was 'Adam', which a stochastic gradient descent optimizer which usually gets good results on a large dataset. The maximum number of iteration was also set to 500 instead of 200. The batch size was set to 'auto', the alpha regularization term set to a default of 10E-4 and the learning rate set to 10E-4, except on some specifics analysis.

Logistic regression

Logistic models are broadly used in a statistical model and are one of the common ways to practice classification. It modifies the parameters of a logistic function to model a binary dependent variable. For the analysis on sklearn, all the parameters were chosen by default except that penalization (both l1 and l2 penalizations were tried) and the strength of the penalization [69].

Logistic models were used because they are widely used to represent statistical models, and they are very fast to train.

4.2 Binary classification

In this section, the Autumn 2017 database was used for the analysis. The positive class qualified the class which represents "late".

For every experience, the following information will be given for each algorithm :

- Accuracy score for all the test set
- Recall score for the positive class
- F1 score for the positive class
- Training time

The models were run without optimization, and the results are presented in the table 4.1.

Table 4.1 Scores of the different algorithms before optimization

Model	RF	ANN	GBT	Log
Accuracy	0.72	0.69	0.75	0.61
Recall	0.56	0.61	0.50	0.64
F1 Score	0.53	0.52	0.52	0.48
Training time (s)	143	1964	5846	3

Random forest, gradient boosted trees, and artificial neural networks get similar results which outperform the logistic regression. However, the training times for GBT and ANN are significant compared to RF.

The artificial neural network achieves a better recall score with a F1 score similar to the two other models.

A random method based on the proportionality of the database would have produced an accuracy of 0.72 and a recall of 0.28. The models outperform these results. A conclusion is that the models have learned.

4.2.1 Feature Optimization

The topic of this section is to determine the relevancy of the features and to check if the models would perform better without some features. The analysis was run with the Random Forest (because of the training time of the model), and the conclusions were tested with all the models.

Features removal

In this section, each feature was removed one at a time, and the difference in the results with the standard version was analyzed. If removing some features improves the model, a combination of removal of the features will be tested.

Table 4.2 Results with different categorical features removed

Feature removed	Driver Change	Direction	Via	Line	Week day
Accuracy	0.73	0.73	0.73	0.72	0.74
Recall	0.54	0.54	0.54	0.59	0.59
F1 Score	0.53	0.53	0.53	0.54	0.56

Table 4.3 Results with different numerical features removed

Feature removed	Block Rank	Mean Layover Prev	Schedule time	Stop
Accuracy	0.72	0.72	0.71	0.72
Recall	0.54	0.54	0.50	0.57
F1 Score	0.52	0.52	0.50	0.53

From the results of 4.3 and 4.2, three categories of features can be identified :

- positive features, without which the performances of the algorithms decrease : block rank, mean layover prev, schedule time
- neutral features, without which the performances of the algorithms do not change : stops, driver change, direction and via
- negative features, without which the performances of the algorithms increase : the weekdays

For the follow up of the project, only the positive and neutral features were kept.

Features treatment

For various features, various ways of integrating them in the models were tested :

- for the binary features, 'via' and 'driver change' were turned in the numerical features
- for the time scheduled, the feature was either split into the different parts of the day (peak hour) or grouped by half hours (called time slots). The time slots were tested as a numerical feature and categorical feature.
- the block-rank feature was processed into its relative rank in the block. For example, a trip which was the second one of a block composed of four trips was attributed to the value of 0.5.
- all the features were treated as a numerical one, except for the line, direction and weekday (because of the lack of meaning)

Table 4.4 Results with 'via' and 'driver change' features processed as numerical features

Feature removed	Standard	Numerical
Accuracy	0.72	0.73
Recall	0.56	0.54
F1 Score	0.53	0.53

The table 4.4 shows that the accuracy improves but the recall decreases, the f1 score remaining the same. As a result, it was decided not to change the treatment of these features. The explication of these results could be that this feature does not impact in general the results, and thus the way of integrating them does not impact either.

Table 4.5 Time feature

Time features	Scheduled time	Time slot numeric	Time slot category
Accuracy	0.72	0.73	0.73
Recall	0.56	0.53	0.45
F1 Score	0.53	0.52	0.48

The table 4.5 shows that splitting the scheduled time into half-hour category does not improve the result. Moreover, if it is considered as a categorical feature, information is lost, and thus it depreciates the model.

Table 4.6 Block Rank features

Feature removed	Block Rank	Block Rank Norm
Accuracy	0.72	0.73
Recall	0.56	0.53
F1 Score	0.53	0.52

Normalizing the block rank feature does not change the results of the model, as displayed in 4.6.

Table 4.7 Block Rank features

Feature removed	All features	Numerical features	Numerical and Binary features
Accuracy	0.72	0.73	0.72
Recall	0.56	0.64	0.64
F1 Score	0.53	0.56	0.56

The table 4.7 shows that removing the categorical features improves the model accuracy and recall, and moreover, it decreases its training time. However, this analysis is valid for the random forest model and should be tested on the other models.

Conclusion of feature optimization

Once the analyses were done on random forest, the combination of features which outperformed the first model had to be tested with the other models to generalize the conclusions. Table 4.8 shows the results when only the numerical features were tested.

Table 4.8 Scores with all the features in numeric categories

Model	RF	ANN	GBT	Log
Accuracy	0.73	0.69	0.73	0.51
Recall	0.63	0.65	0.60	0.55
F1 Score	0.56	0.55	0.55	0.38
Training time (s)	50	19922	3919	0.5

Results were better for the machine learning models but not for the logistic model. Moreover, the training time has reduced a lot except for the ANN. The ANN model has an extended training time because it went through 500 iterations, which was the set limit.

Then the second experiment was done and all the features except the weekdays were kept. The results are presented in table 4.9.

Table 4.9 Scores with all the features except Week-days

Model	RF	ANN	GBT	Log
Accuracy	0.74	0.70	0.75	0.61
Recall	0.59	0.67	0.56	0.64
F1 Score	0.56	0.55	0.55	0.48
Training time (s)	190	4590	3893	21

The training time is longer because the number of features as input is substantial. However, the results have improved compared to the first model, especially for the logistic regression.

The two combinations of features outperformed the first machine learning model, but only when the weekday feature was removed that the results improved for the logistic regression. Thus it was decided to keep the second combination (without the weekday) to allow the logistic regression to perform better and to keep a maximum of information. Once again, it can be noted that the artificial neural network has a more important recall score than the other models but the same f1 score.

4.2.2 Oversampling

Oversampling the data is very important for building machine learning on an unbalanced dataset. The original method used was the SMOTE oversampling method : Synthetic Minority Over-sampling Technique. This technique creates new data from existing ones and generally outperforms the other over-sampling methods [70].

Two other over-sampling methods were tested : the naive one and the ADASYN, short for Adaptive Synthetic. The naive method duplicates random data from the original dataset, whereas the ADASYN generates new data as the SMOTE method, but focuses more on the data which are difficult to learn [71].

The analysis for comparing the different oversampling methods was made with the four machine learning models. For each over-sampling model, the ratio is set to 1 : for the learning phase, both classes are represented equally.

The four following tables present the results for the models tested, using the various over-sampling methods described.

Table 4.10 Scores with RF depending on the Over-sampling Method

Oversampling Methods	NONE	SMOTE	Random	ADASYN
Accuracy	0.76	0.74	0.70	0.73
Recall	0.46	0.59	0.69	0.61
F1 Score	0.52	0.56	0.56	0.56

Table 4.11 Scores with ANN depending on the Over-sampling Methods

Oversampling Methods	NONE	SMOTE	Random	ADASYN
Accuracy	0.76	0.70	0.70	0.69
Recall	0.45	0.67	0.69	0.69
F1 Score	0.51	0.55	0.56	0.55

Table 4.12 Scores with GBT depending the Over-sampling Methods

Oversampling Methods	NONE	SMOTE	Random	ADASYN
Accuracy	0.76	0.75	0.70	0.74
Recall	0.43	0.56	0.70	0.58
F1 Score	0.51	0.55	0.56	0.56

Table 4.13 Scores with logistic regression depending on the Over-sampling Methods

Oversampling Methods	NONE	SMOTE	Random	ADASYN
Accuracy	0.73	0.61	0.61	0.60
Recall	0.09	0.64	0.64	0.67
F1 Score	0.16	0.48	0.48	0.48

In all the models, the results are improved by the presence of over-sampling. The results decrease a lot when there is no oversampling. All the oversampling methods perform similarly. Then the Smote method was kept for the rest of the analysis.

4.2.3 Dimensionality reduction

The categorical features are processed and transformed via the 'one-hot' method. It increases a lot the number of features, and thus increase the training time and the difficulty to tune the hyper-parameters. The dimensionality reduction is a method which decreases the number of features by gathering them into a reduced number of featured which is supposed to give more meaning to the dataset [72].

The dimensionality reduction method used was the most classical one : the principal component analysis, which decomposes the data via their singular value in order to project into lower dimensional space [73].

For the three models, three number of components were tried. Out of the 198 original features, it was reduced to 20, 50 and 100. The following tables display the results for the various algorithms.

Table 4.14 Scores with RF depending on the Dimensionality Reduction factor

Number of inputs	198	20	50	100
Accuracy	0.74	0.71	0.72	0.73
Recall	0.59	0.63	0.65	0.64
F1 Score	0.56	0.55	0.57	0.57

Table 4.15 Scores with ANN depending on the Dimensionality Reduction factor

Number of inputs	198	20	50	100
Accuracy	0.70	0.70	0.71	0.71
Recall	0.67	0.66	0.65	0.65
F1 Score	0.55	0.56	0.55	0.55

Table 4.16 Scores with GBT depending on the Dimensionality Reduction factor

Number of inputs	198	20	50	100
Accuracy	0.75	0.73	0.73	0.73
Recall	0.56	0.60	0.61	0.61
F1 Score	0.55	0.56	0.56	0.56

Table 4.17 Scores with Logisitic regression depending on the Dimensionality Reduction factor

Number of inputs	198	20	50	100
Accuracy	0.62	0.56	0.53	0.59
Recall	0.64	0.56	0.62	0.60
F1 Score	0.48	0.41	0.42	0.45

The dimensionality reduction method does not improve the results for the different algorithms relevantly. Thus this method was not used for the other analysis.

4.2.4 Removing the first trips

The "mean layover previous" was set to 20 for the first stops of the trips. Moreover, as discussed in section 3, there are two possibilities for a bus to be late at the end of a trip. The first one is the inherent probability of being late. The second one is because it is the cause of delay propagation. So for the first stops, only the first reason is possible.

Then, the first trip of each rank was removed from the database and the results are presented in table 4.18 :

Table 4.18 Scores of the different models with the first trips removed

Model	RF	ANN	GBT	Log
Accuracy	0.75	0.72	0.74	0.62
Recall	0.61	0.66	0.63	0.65
F1 score	0.57	0.56	0.57	0.48
Training time (s)	165	12170	5128	10

Without the first trips, the models improve a little and the training time decreases a little. It could be explained by the mean layover previous being more accurate. However, the improvement is not significant enough compared to the number of data removed from the database. Then all the lines will be kept for further analysis.

4.2.5 Models on one line

The models are always trained on the whole database. Even if the line affected is taken into account, it is interesting to see if the results would be better with the models fit on one database. In order to get enough data for the training phase, the line tested was the one with the most occurrences on the database, which were line 51 and its 3220 occurrences in the database.

As only one line was taken into account, some features were deleted such as the stop coordinates and the line feature. Indeed, their information indicate the difference between the lines, and thus they lack relevance when the tests are made on only one line. The results are displayed on table 4.19.

Table 4.19 Scores of the different models on the line 51

Model	RF	ANN	GBT	Log
Accuracy	0.66	0.66	0.66	0.56
Recall	0.57	0.57	0.55	0.57
F1 score	0.55	0.55	0.54	0.49
Training time (s)	0.5	19	4	0.01

Training the model on only one line has two main issues : the lack of data and the impossibility for the model to find a larger pattern to the task, such as the links between the time of the day and the geography of the city. Hence the models with the whole database outperform model with only one line.

4.2.6 Overfitting

In machine learning, overfitting describes the over learning of the models, which represent too closely the set of training and thus may be unsuccessful to estimate other observations [74].

In order to check the overfitting of the models, the accuracy and recall score on the training set can be produced. They are presented in table 4.20.

Table 4.20 Scores of the training and the test set for the different models

Model	RF	ANN	GBT	Log
Accuracy on test set	0.74	0.70	0.75	0.62
Recall on test set	0.59	0.67	0.56	0.64
Accuracy on training set	0.85	0.81	0.85	0.63
Recall on training set	0.87	0.86	0.86	0.64

Hence the models perform better on the training set, but this analysis cannot prove the overfitting.

The overfitting can be properly analysed by drawing learning curves on the ANN. For different learning rate, the learning curves for accuracy and recall are presented on the following figure :

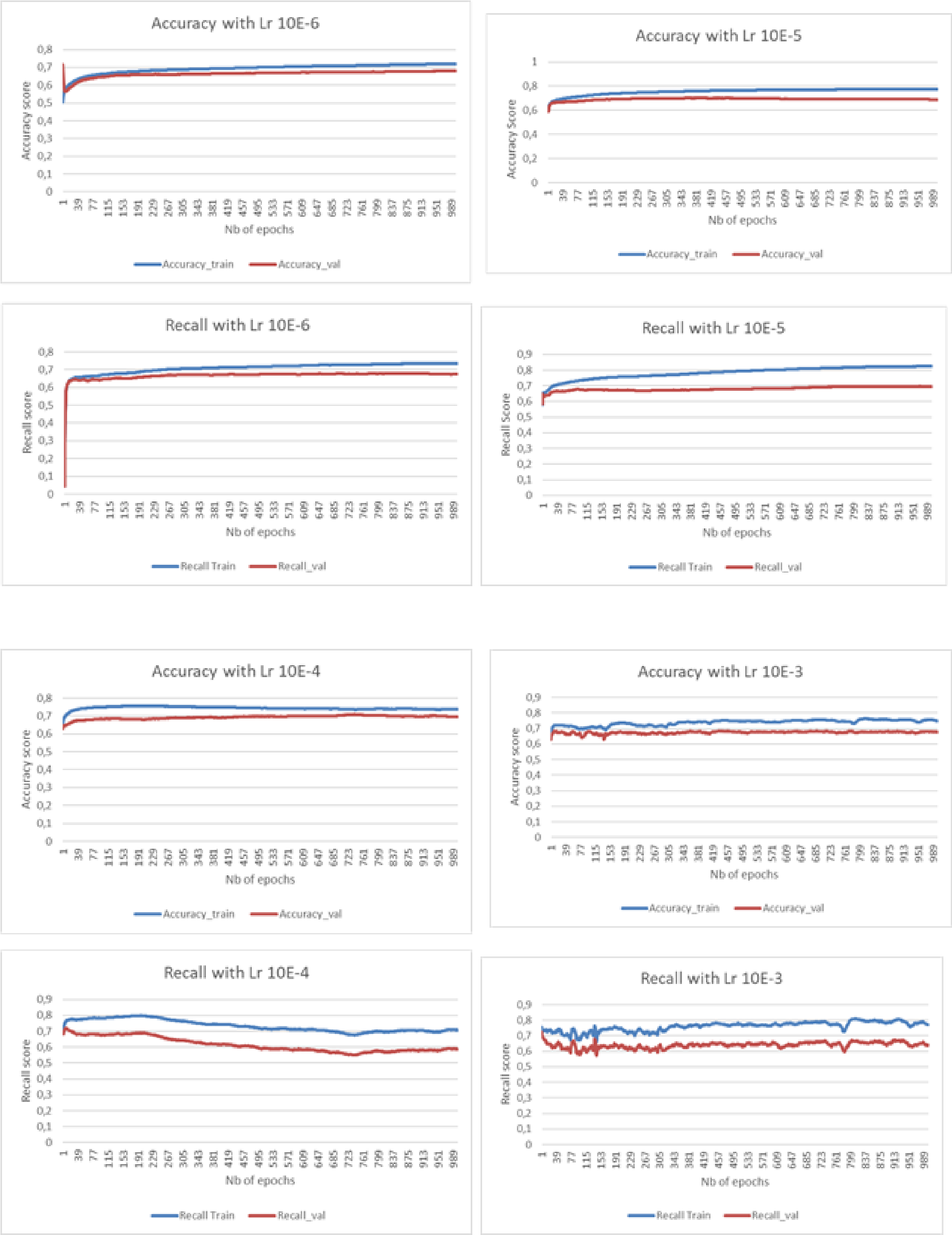


Figure 4.1 Accuracy and Recall Score depending the learning rate (continued)

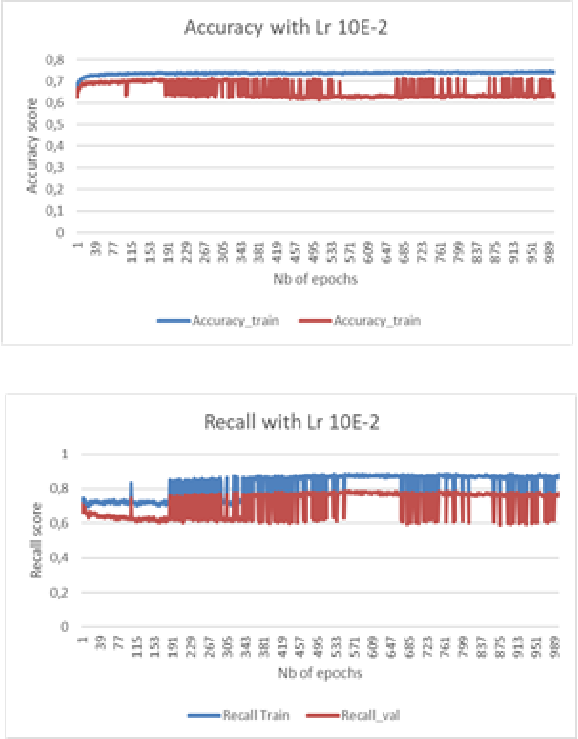


Figure 4.2 Accuracy and Recall Score depending the learning rate (continued and end)

The training results are better than the test results. However, no overfitting could be detected. It can be noted that too much learning during the learning phase can cause problems in the learning, and an insufficient one can lead to important training time. During the analysis, the learning rate of 10E-4 was used.

It can also be noted that with a maximum of 500 iterations, the maximum is almost achieved.

4.2.7 Comments on the results

The model with optimized features has finally not been modified with all the proposed optimization methods. Indeed, the improvement found in the different steps of this section was not relevant enough to be integrated into the model. The set of test data could have caused the difference with the initial models.

In order to visualize better the results, confusion matrices were produced for each model. The normalized confusion matrices are displayed :

As noted previously, the artificial neural network obtains a better recall score but a worse accuracy than the other machine learning models. The three machine learning models outperforms the logistic regression

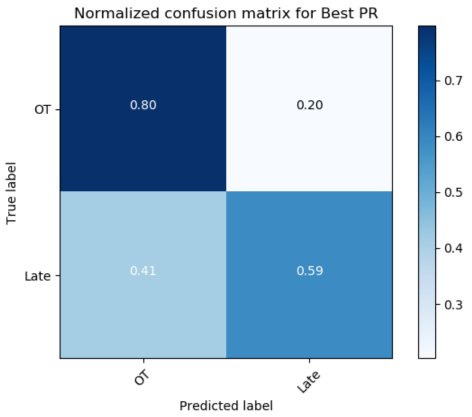


Figure 4.3 Normalized confusion matrix for Random Forest

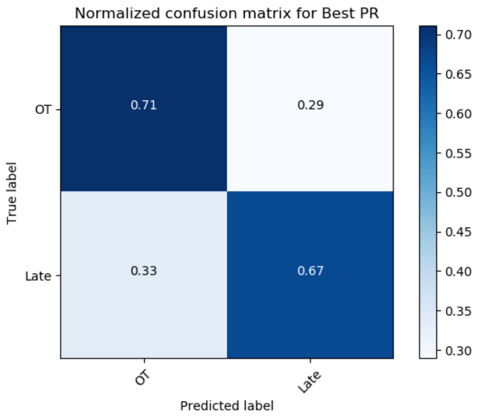


Figure 4.4 Normalized confusion matrix for Artificial neural network

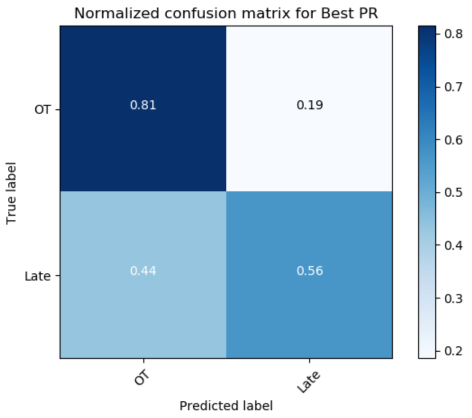


Figure 4.5 Normalized confusion matrix for Gradient boosted tree

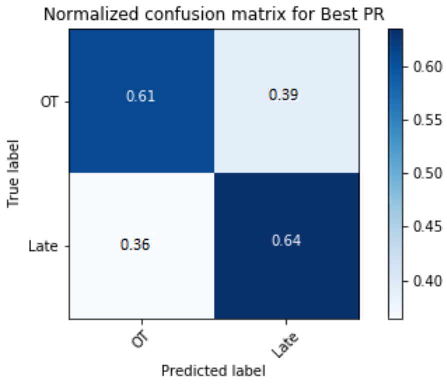


Figure 4.6 Normalized confusion matrix for Logistic regression

4.3 Multi-label classification

The multi-label classifiers aim at predicting in which time slot delay each trip would end. It estimates probabilities for being into each slot and returns the class which has the highest chance.

The time slots were first chosen as displayed in the section which described the database : the first class means the trip end before its scheduled time, the second one means that the trip has less than three minutes of delay, the third one less than ten minutes of delay and the last class is for the buses ending with more than ten minutes of delay. As shown in the figure 3.8, the data are unbalanced. That is the reason why the oversampling method was also needed for this classifier.

The same three machine learning algorithms were tested, with the features and the parameters optimal from the binary model. The multi-linear logistic regression was also tried in comparison.

The metric used was the weighted mean of the f1 score for each one of the classes. The result produced was a classification report which displays for each class the accuracy, recall and f1 score. The data come from the Autumn 2017 database.

4.3.1 First results

Random Forest

Table 4.21 Classification report for the Random forest algorithm

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.66	0.65	0.65	6053
Class 2	0.38	0.35	0.36	3354
Class 3	0.39	0.40	0.40	2784
Class 4	0.33	0.42	0.37	858
Avg weighted	0.51	0.51	0.51	13049

The training time for the best model was 154 seconds.

Gradient boosted tree

Table 4.22 Classification report for the Gradient boosted tree algorithm

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.65	0.67	0.66	6053
Class 2	0.38	0.34	0.36	3354
Class 3	0.40	0.40	0.40	2784
Class 4	0.36	0.42	0.39	858
Avg weighted	0.51	0.51	0.51	13049

The training time for the best model was 19803 seconds.

Artificial Neural Network

Table 4.23 Classification report for the artificial neural network algorithm

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.67	0.58	0.62	6053
Class 2	0.37	0.36	0.36	3354
Class 3	0.37	0.42	0.39	2784
Class 4	0.28	0.47	0.35	858
Avg weighted	0.50	0.48	0.49	13049

The training time for the best model was 43546 seconds.

Multi-linear regression

Table 4.24 Classification report for the multi-linear logistic regression

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.62	0.41	0.50	6053
Class 2	0.34	0.32	0.33	3354
Class 3	0.30	0.21	0.25	2784
Class 4	0.13	0.59	0.21	858
Avg weighted	0.44	0.36	0.38	13049

The training time for the best model was 9 seconds.

synthesis

Table 4.25 Multi-label classification

Model	RF	ANN	GBT	Log
Accuracy Mean	0.51	0.50	0.51	0.44
Recall Mean	0.51	0.48	0.51	0.36
F1 score Mean	0.51	0.49	0.51	0.38
Training time(s)	154	43546	19803	9

The random forest and the gradient boosted tree algorithms outperform the other algorithms. However, artificial neural networks have a better recall score for the classes representing high delay. The difference is however not significant.

A random model predicting the delays based on the proportion of the samples in each class would have obtained a weighted f1 score of 0.33. Then the models trained are better than a random model.

4.3.2 Unsupervised for classes

The different classes were chosen arbitrarily and to represent significant delay. One way of selecting the classes would be to use unsupervised learning on the delay arrival times to get different classes. Indeed, one of the objectives of unsupervised learning is to find patterns between the data.

The unsupervised algorithms used was a k-means algorithm, which creates clusters from one data series. For a given number of clusters, it decides the center of the clusters and then affects data to the cluster [75]. The algorithm was used through sklearn [76].

The metric decided was the silhouette score which calculates for each distance the intra-cluster distance and the closest-cluster distance. The number of clusters chosen will depend on the silhouette score and the possible meaning of the different borders. The range value of the silhouette score is between -1 and 1, 1 meaning that the clusters are perfect. The number of clusters tested was from two to six.

The table 4.26 displays the silhouette score and the inferior border of each cluster.

Table 4.26 Unsupervised clustering using k-means on Autumn 2017

Number of clusters	silouhette score	borders
n = 2	0.63	[-1929, 268]
n = 3	0.56	[-1929, 55, 556]
n = 4	0.53	[-1929, -49, 267, 788]
n = 5	0.52	[-1929, -120, 118, 433, 958]
n = 6	0.51	[-1929, -194, 12, 240, 557, 1076]

The silhouette score is superior to 0.5, which means that the cluster is relevant. The silhouette score decreases when the number of cluster increases.

For binary classification, unsupervised learning indicates that the borders between the two classes should have been at 268 seconds. However, this choice would not have been relevant because it would have separated trips which ended with four minutes and a half of delay, which has no practical meaning.

Models with three, four and five classes from the unsupervised learning are tested in the following section.

4.3.3 Multi-label classification with optimized classes

For each algorithm, the classification report for multi-label classification with three, four and five classes coming from the unsupervised learning is displayed. For each algorithm, several hyper-parameters were tried, and the results presented come from the test of the best models on the test set.

Random Forest

Table 4.27 Classification report for the Random forest algorithm and 3 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.72	0.70	0.70	7302
Class 2	0.53	0.53	0.53	4764
Class 3	0.35	0.45	0.40	983
Weighted Avg	0.63	0.62	0.62	13049

The training time for the best model was 225 seconds.

Table 4.28 Classification report for the Random forest algorithm and 4 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.61	0.62	0.61	4917
Class 2	0.57	0.53	0.55	5468
Class 3	0.37	0.39	0.38	2163
Class 4	0.30	0.40	0.34	501
Weighted Avg	0.54	0.53	0.54	13049

The training time for the best model was 187 seconds.

Table 4.29 Classification report for the Random forest algorithm and 5 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.52	0.54	0.53	3263
Class 2	0.54	0.51	0.53	5239
Class 3	0.39	0.38	0.39	3058
Class 4	0.32	0.36	0.34	1184
Class 5	0.26	0.36	0.30	305
Weighted Avg	0.48	0.47	0.47	13049

The training time for the best model was 211 seconds.

Gradient boosted tree

Table 4.30 Classification report for the Gradient boosted tree algorithm and 3 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.72	0.72	0.72	7302
Class 2	0.54	0.53	0.53	4764
Class 3	0.38	0.43	0.40	983
Weighted Avg	0.63	0.63	0.63	13049

The training time for the best model was 14809 seconds.

Table 4.31 Classification report for the Gradient boosted tree algorithm and 4 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.61	0.62	0.61	4917
Class 2	0.57	0.55	0.56	5468
Class 3	0.38	0.38	0.38	2163
Class 4	0.30	0.40	0.34	501
Weighted Avg	0.54	0.54	0.54	13049

The training time for the best model was 19603 seconds.

Table 4.32 Classification report for the Gradient boosted tree algorithm and 5 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.53	0.53	0.53	3263
Class 2	0.54	0.53	0.54	5239
Class 3	0.40	0.38	0.39	3058
Class 4	0.33	0.35	0.34	1184
Class 5	0.26	0.37	0.31	305
Weighted Avg	0.48	0.47	0.48	13049

The training time for the best model was 32053 seconds.

Artificial Neural Network

Table 4.33 Classification report for the Artificial Neural Network algorithm and 3 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.73	0.68	0.70	7302
Class 2	0.53	0.50	0.52	4764
Class 3	0.30	0.51	0.38	983
Weighted Avg	0.62	0.60	0.61	13049

The training time for the best model was 53155 seconds.

Table 4.34 Classification report for the Artificial Neural Network algorithm and 4 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.61	0.60	0.60	4917
Class 2	0.57	0.44	0.50	5468
Class 3	0.32	0.44	0.37	2163
Class 4	0.23	0.45	0.31	501
Weighted Avg	0.53	0.50	0.51	13049

The training time for the best model was 41248 seconds.

Table 4.35 Classification report for the Artificial Neural Network algorithm and 5 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.51	0.51	0.51	3263
Class 2	0.54	0.46	0.50	5239
Class 3	0.38	0.38	0.38	3058
Class 4	0.27	0.37	0.31	1184
Class 5	0.19	0.44	0.27	305
Weighted Avg	0.46	0.44	0.45	13049

The training time for the best model was 52352 seconds.

Logistic regression

Table 4.36 Classification report for the Logistic regression algorithm and 3 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.69	0.51	0.58	7302
Class 2	0.45	0.29	0.35	4764
Class 3	0.14	0.64	0.23	983
Weighted Avg	0.56	0.44	0.47	13049

The training time for the best model was 115 seconds.

Table 4.37 Classification report for the Logistic regression algorithm and 4 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.53	0.45	0.48	4917
Class 2	0.54	0.28	0.37	5468
Class 3	0.24	0.31	0.27	2163
Class 4	0.09	0.58	0.15	501
Weighted Avg	0.47	0.36	0.39	13049

The training time for the best model was 71 seconds.

Table 4.38 Classification report for the Logistic regression algorithm and 5 optimized classes

Model	Accuracy	Recall	f1 score	Nb of samples
Class 1	0.37	0.43	0.40	3263
Class 2	0.52	0.25	0.34	5239
Class 3	0.29	0.17	0.21	3058
Class 4	0.15	0.29	0.19	1184
Class 5	0.07	0.57	0.12	305
Weighted Avg	0.38	0.29	0.31	13049

The training time for the best model was 24 seconds.

Synthesis

Table 4.45 synthesis the results for a four labels classification for the different algorithms. The four labels were chosen to compare the results with the previous analysis with classes chosen arbitrarily.

Table 4.39 Multi-label classification with 4 optimized classes

Model	RF	ANN	GBT	Log
Accuracy Mean	0.54	0.53	0.54	0.47
Recall Mean	0.53	0.50	0.54	0.36
F1 score Mean	0.54	0.51	0.54	0.36
Training time (s)	188	41248	19603	71

Once again, random forest and gradient boosted trees performs better than the other models, even if the artificial neural network has similar results. It could be explained by the difficulty to tune the parameters of the artificial neural network.

The artificial neural network and for the gradient boosted models are training for a longer time than the other models.

Moreover, the results have improved with the use of unsupervised classes. There is a potential to predict time slot delays for end trip using planning data.

4.3.4 Characterisation of the errors

The errors can be described as under-evaluating or over-evaluating the mistake. To compare that, for each optimized multi-label classifier models the difference between the predictions and the real results were analyzed on the test set. They are going to be classified into several categories. If a prediction is over-estimating a delay, it will go in the group "+" and the number of differences in the forecast.

For example, if a delay should be labeled in the second class (between -49 and 267 seconds) but the trips end before the scheduled time (first class), it would go to the category "-1". As there are 4 categories, six different categories for the errors exist : "-3", "-2", "-1", "+1", "+2" and "+3". The information about the correct predictions is also displayed.

The analysis is made with the four classes from the unsupervised learning in the autumn 2017 database.

Random Forest

Table 4.40 Errors classification for Random Forest

Class	neg3	neg2	neg1	Correct	pos1	pos2	pos3
Number	92	587	2441	6981	2433	453	62
Percentage	1	4	19	53	19	3	0.5

Gradient Boosted Tree

Table 4.41 Errors classification for GBT

Class	neg3	neg2	neg1	Correct	pos1	pos2	pos3
Number	98	625	2455	7002	2342	468	59
Percentage	1	5	19	54	18	4	0.5

Artificial Neural Network

Table 4.42 Errors classification for ANN

Class	neg3	neg2	neg1	Correct	pos1	pos2	pos3
Number	171	910	2738	6525	2246	403	56
Percentage	1	7	21	50	17	3	0.4

Logistic regression

Table 4.43 Errors classification for logistic regression

Class	neg3	neg2	neg1	Correct	pos1	pos2	pos3
Number	992	2066	2833	4713	1901	491	53
Percentage	8	16	22	36	15	4	0.4

Conclusion

The results are synthesized under a graph presented in figure 4.7

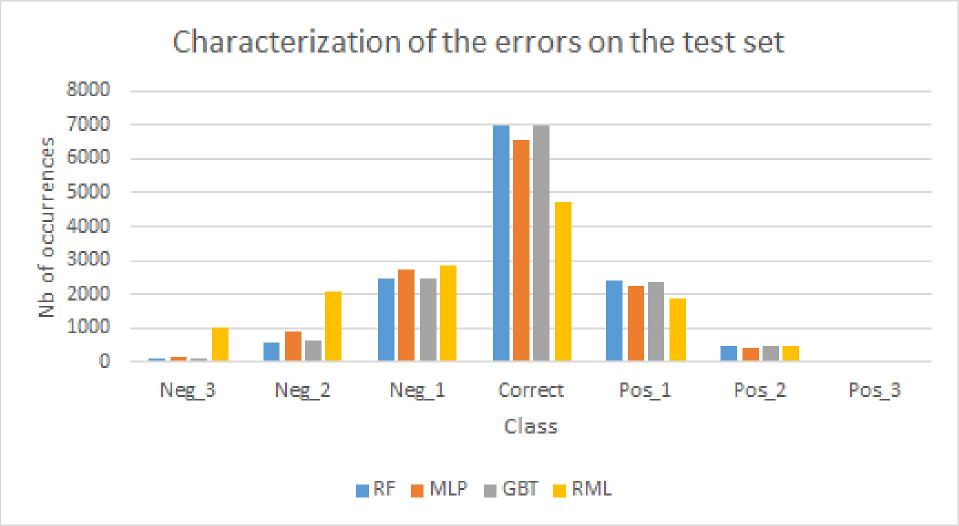


Figure 4.7 Occurences for each classes

The logistic regression presents more mistakes than the other models. As predicted, random forest and gradient boosted have similar repartition of their mistakes. Artificial neural networks present more negative mistakes because it tends more to predict high delay categories than the other machine learning models.

4.4 Binary model on several periods

The idea of the project is to build machine learning models to predict future planning. In the previous section, it has been demonstrated that it is possible to learn from schedule data to predict end-trip bus delay in one period.

However, the objective of GIRO is to predict reliability for future schedules. Then, tests should be made to assess the possibility to learn on a period and to test it on another.

Firstly, models are tried with database composed of the aggregation of the autumn 2016 and the autumn 2017 databases. These analyses assess the possibility of mixing different databases. Then the models are trained on the autumn 2016 database and tested on the autumn 2017 database.

4.4.1 Problems linked to linking different databases

The data from Autumn 2016 were treated as the one from Autumn 2017. However, two lines were not present in autumn 2016. Thus, the data containing these lines were removed.

Moreover, other problems appeared : how to manage the difference in input data, especially for the preprocessing of the one-hot and the regularisation.

It was decided that all the data would be preprocessed together then split for the different step.

4.4.2 Bilinear model with both database

First, binary classifier models were trained with the new database. The preprocessing method, the oversampling method and the features chosen were the same used for the binary classification on one period. The four usual algorithms were tried.

Table 4.44 Binary classification with Autumn2016 and Autumn 2017 databases

Model	RF	ANN	GBT	Log
Accuracy	0.74	0.72	0.75	0.61
Recall	0.58	0.63	0.57	0.62
F1 score	0.56	0.55	0.56	0.47
Training time(s)	316	19903	9298	23

The results are similar to the one found with the binary classifier models : the f1 score is the same, and artificial neural networks achieve better to find delays but have a worse overall accuracy.

The training time is longer than in the model with one database because more data were present in the training set. The training time for an artificial neural network does not change because the maximum number of iterations stopped the training.

4.4.3 Fitting on 2016 to predict on 2017

Then it was tried to fit a model on Autumn 2016 and test it on Autumn 2017. The four usual algorithms were tried.

Table 4.45 Binary classification fit on Autumn2016 test on Autumn 2017

Model	RF	ANN	GBT	Log
Accuracy	0.64	0.63	0.64	0.59
Recall	0.49	0.43	0.46	0.53
F1 score	0.43	0.40	0.43	0.41

The results are not as good as the one obtained when trained and tested on both databases. The models are still better than random models. The logistic regression has the same performance as the machine learning models.

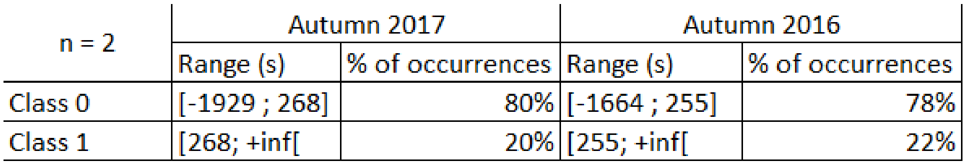
However, the results of the previous section have shown that the models can learn on several databases. Analyses should be done to find a way of improving the transfer of learning over several periods.

4.4.4 Unsupervised analysis

Unsupervised learning has been tested on the autumn 2016 period and compared to the autumn 2017 unsupervised learning to assess the possibility of using this method to find clusters for multi-label classification. As a recall, the clusters are formed by the end trip delays and are in seconds. Indeed, using unsupervised learning for choosing the classes improve the results for multi-label classification, as it was shown in the previous section.

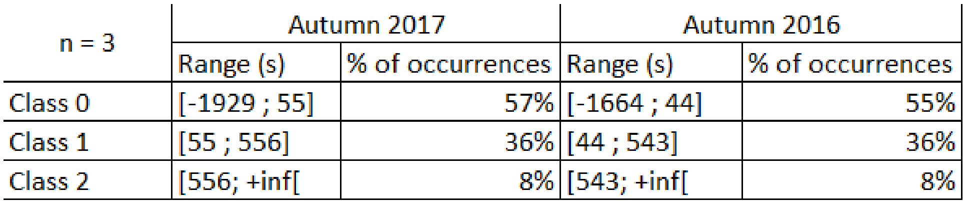
The following figures display for several numbers of clusters the results of unsupervised learning for autumn 2016 and autumn 2017. The range of the clusters and the percentage of occurrences present in each cluster are presented for each period.

Table 4.46 Comparison of Unsupervised learning with 2 clusters



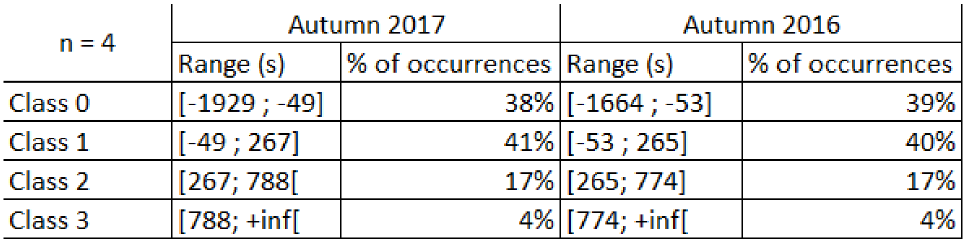
n = 2	Autumn 2017		Autumn 2016	
	Range (s)	% of occurrences	Range (s)	% of occurrences
Class 0	[-1929 ; 268]	80%	[-1664 ; 255]	78%
Class 1	[268; +inf[	20%	[255; +inf[	22%

Table 4.47 Comparison of Unsupervised learning with 3 clusters



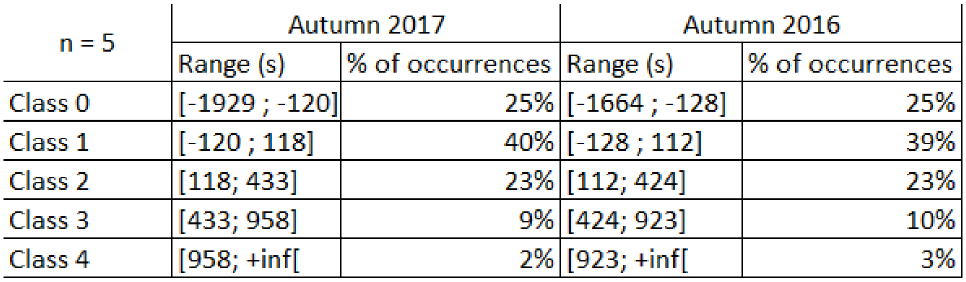
n = 3	Autumn 2017		Autumn 2016	
	Range (s)	% of occurrences	Range (s)	% of occurrences
Class 0	[-1929 ; 55]	57%	[-1664 ; 44]	55%
Class 1	[55 ; 556]	36%	[44 ; 543]	36%
Class 2	[556; +inf[	8%	[543; +inf[	8%

Table 4.48 Comparison of Unsupervised learning with 4 clusters



n = 4	Autumn 2017		Autumn 2016	
	Range (s)	% of occurrences	Range (s)	% of occurrences
Class 0	[-1929 ; -49]	38%	[-1664 ; -53]	39%
Class 1	[-49 ; 267]	41%	[-53 ; 265]	40%
Class 2	[267; 788[	17%	[265; 774]	17%
Class 3	[788; +inf[	4%	[774; +inf[	4%

Table 4.49 Comparison of Unsupervised learning with 5 clusters



n = 5	Autumn 2017		Autumn 2016	
	Range (s)	% of occurrences	Range (s)	% of occurrences
Class 0	[-1929 ; -120]	25%	[-1664 ; -128]	25%
Class 1	[-120 ; 118]	40%	[-128 ; 112]	39%
Class 2	[118; 433]	23%	[112; 424]	23%
Class 3	[433; 958]	9%	[424; 923]	10%
Class 4	[958; +inf[	2%	[923; +inf[	3%

For each analysis, the border and the percentage of occurrences are similar for each cluster in both periods. Unsupervised learning to determine classes could be integrated in the future for improving the learning on multiple periods.

4.5 Conclusion

In this section, it has been proven that the three machine learning models can model end trip delays using offline data, for both binary and multi-label classification. The machine learning models perform better than logistic regression. Random forest and gradient boosted trees outperform artificial neural network in term of the f1 score, but get a higher recall score the critical delays.

Building unsupervised machine learning models can improve the results for multi-label classification. Moreover, the comparison of the results of unsupervised learning in the autumn 2016 and 2017 databases have shown that the classes created are similar. Further analysis to identify the pattern for delays should be done.

Models can learn on several periods ; however, the results from learning on one database and testing on others are not relevant yet. Analyses should be done to understand and then solve this issue.

CHAPTER 5 EVALUATION OF THE SCHEDULE RELIABILITY

The objective of the research is to model the bus trip arrival delay to assess schedule reliability. The interest for GIRO was to estimate the delay probabilities for the different trips of a schedule.

Moreover, it is interesting to compare the results with statistics methods. Indeed, a normal distribution is easy to fit and is commonly used to represent reality.

Working with the probability of being in each time slot delays could assess the quality of the model and could be used by companies to improve their service reliability.

In this section, the analysis will be made with the optimized multi-labels models from Autumn 2017.

5.1 Method for assessing schedule reliability

A method for assessing schedule reliability which could be generalized over the time had to be found. Indeed, the trips are not the same over the different periods.

Estimating for the different time slot of the day the probability for each trip to end late would allow assessing the reliability of the service. Moreover, comparing the estimated distribution to the real one would permit to compare the prediction models to each other.

5.1.1 Methodology

Different trips are planned over various periods. In order to compare them, it is interesting to introduce the notion of Time Slot (TS). The day would be split in TS, which could be of various sizes depending on the part of the day. For the various analyses, TS could vary between 1 minute and 30 minutes.

For the rest of the work, a combination of TS, line, and direction will be called TSLD.

Once the day has been split into TS, they are attributed to the different trips. Then the trips are gathered by line, direction, and TS. Then for each of the instances (TS/line/direction) created, a probability distribution of being in each delay classes is created by averaging all the different distributions within it.

For example, if the TS value is 15 minutes, and four trips are leaving the line "L" in direction "D" between 8.15 a.m. and 8.30a.m. (the time slot associated is then 33, because 8.15a.m.

is the 33rd quarter of 15 minutes of the day). The probability of ending in class 0 for the aggregated 33LD is the mean of the probability of being in class 0 of the four trips.

Then, the estimated probability is compared to real behavior. The real behavior is estimated by counting for each TSLD the occurrences of being in each delay class, and the frequencies are calculated. Therefore, the distance between reality and the estimation are calculated through different metrics presented later.

The normal distribution used as a comparison was created by calculating the mean and the standard deviation of the delays end trip for each TSLD.

5.1.2 Comparing to probabilistic laws

The model is compared to probabilistic laws because they are the ones which could reproduce the reality and are easy to set up. The most common and used one is the normal distribution. It is characterized by its mean μ and by its standard deviation σ.

$$\phi(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

In practice, for all databases, trips were gathered by TSLD, and the delay means and the standard deviations were calculated. Then a normal distribution probability density was computed with the two parameters, and the probability of being in each time slot was calculated. It gave a distribution of probability which could be compared to the probabilities from the machine learning models.

With the example quoted before with the four trips of the line L in direction D for the TS 33, if the mean of delays for the four trips is 120 and the standard deviation of 130, a normal distribution law is associated with these values. Then the probability distribution of having a delay between -47 and 268 seconds is given by the formula :

$$P(-47 \leq x \leq 268) = \int_{-47}^{268} \phi(x|120, 130)dx$$

$$P(-47 \leq x \leq 268) = 0.77$$

5.1.3 Metrics for comparing distribution

Two distances were used to compare for each TSLD the probability distribution from the different methods and reality given by the frequencies.

The estimated probability for being in the class c for a TLSD is noted p_cTLSD. The frequency for the class c of a TLSD is noted $f_c TLSD$.

The first distance calculated is the Euclidean distance. It was used because this distance is often used to compare distance. For each combination, the difference between the classes estimated and the frequency was calculated, and then the Euclidean distance was calculated :

$$D_E(TLSD) = \sqrt{\sum_{c=1}^{4}(p_c(TLSD) - f_c(TLSD))^2}$$

The second distance is the Bhattacharyya distance. It was used because it measures the similarity between two discrete distribution probabilities. The Bhattacharyya distance for a TSLD is defined as :

$$D_B(TLSD) = -ln(BC(TLSD)$$

where :

$$BC(TLSD) = \sum_{c=1}^{4} \sqrt{p_c(TLSD) * f_c(TLSD)}$$

Once the distance calculated for each TSLD, the gap was multiplied by the number of occurrences of the TLSD in reality, in order to weight each TSLD. All the multiplications were summed to get final results. The distance was weighted because the most frequent TSLD needs to have more impact on the final results. Moreover, multiplying by the number of occurrences permits to compare the models with different TS between them.

5.2 Assessing the service reliability

The model was assessed according to the methodology described in the previous section. The results will be displayed under tables presenting the total weighted sum with Euclidean and Bhattacharyya distances.

5.2.1 Presentation of the results

The analyses were made by gathering the line and direction with a different time slot. The time slot of one minute, five minutes, fifteen and thirty minutes were tested. The results are displayed in the following tables

Table 5.1 Comparison for the distance for the different models with TS of 1 minute

Model	Normal Distribution	RF	ANN	GBT	Log
Euclidean Distance	21449	22676	29742	35 570	61757
Bhattacharyya distance	4607	6519	6568	14 844	32051

Table 5.2 Comparison for the distance for the different models with TS of 5 minutes

Model	Normal Distribution	RF	ANN	GBT	Log
Euclidean Distance	21382	22589	29530	35373	61420
Bhattacharyya distance	4564	6449	6428	14674	31720

Table 5.3 Comparison for the distance for the different models with TS of 15 minutes

Model	Normal Distribution	RF	ANN	GBT	Log
Euclidean distance	20498	21601	27697	33531	58279
Bhattacharyya Distance	3975	5660	5343	12942	29375

Table 5.4 Comparison for the distance for the different models with TS of 30 minutes

Model	Normal Distribution	RF	ANN	GBT	Log
Euclidean Distance	19488	20223	25690	31428	54841
Bhattacharyya distance	3388	4754	4370	11062	24743

The Bhattacharyya distance shows more difference between the results, and so seem more relevant for this kind of analysis.

With the TS increasing, the distance decreases. Indeed, with more data to estimate the probability distribution, the results are more accurate.

The artificial neural network and the random forest models perform better than the gradient boosted tree and the logistic regression. Normal distribution performs better than all the models, but the results are biased because the distribution created is directly related to the occurrences.

5.2.2 Analysis on peak hour periods

The analysis could be detailed on the most critical part of the day : the peak hours. For this peak hour, the analysis is done with time slots of 15 minutes. The aggregations with the time slots included in the peak hours were extracted and compared.

Table 5.5 Comparison for the distance for the different models with TS of 15 minutes in peak-hour periods

Model	Normal Distribution	RF	ANN	GBT	Log
Euclidean Distance	8562	8982	12231	13992	23455
Bhattacharyya distance	1592	2199	2344	5155	10699

The results are similar to the analysis of the overall database. Random forests and artificial neural networks outperform gradient boosted trees.

5.2.3 Explaination of the difference between the models

During the building and the analysis of the ML models, gradient boosted trees obtained the same score in metrics as the random forests, but the random is more precise for service reliability. To understand this, the standard deviation has been calculated for all the TSLD with a TS of 30 minutes. Then the mean of all the standard deviation has been calculated, and results are displayed in table 5.6

Table 5.6 Comparison for the Mean Standard Deviation with TS of 30 minutes

Model	Normal Distribution	RF	ANN	GBT	Log
Mean Standard Deviation	0.28	0.25	0.30	0.18	0.13

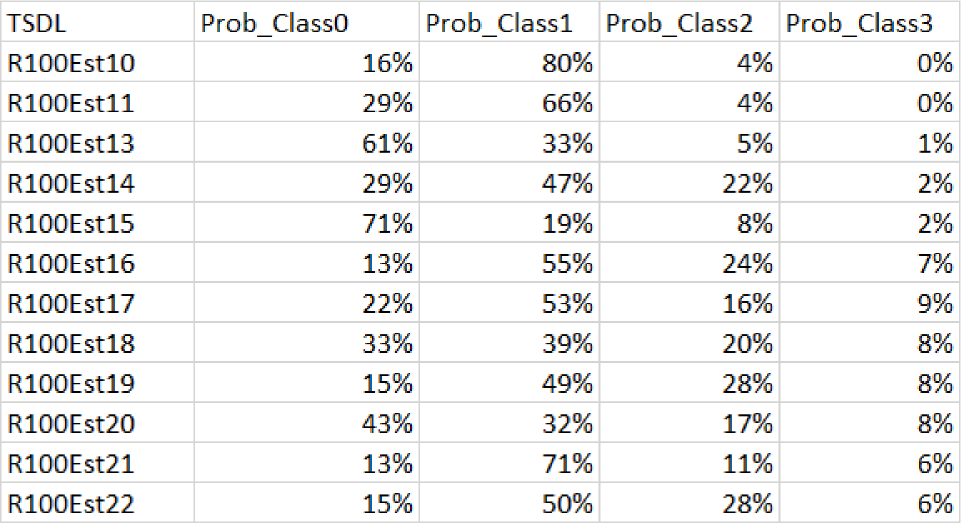
The mean, standard deviation is more important with the normal distribution, random forest and artificial neural networks rather than in the other models. It explains that gradient boosted trees produce probability distribution more flattered and then the distance with frequency is more important.

In all these models, the artificial neural network model has the most significant mean standard deviation, which shows the potential of this model to estimate accurately service reliability.

5.3 Conclusion

A document that GIRO could directly use was proposed. Figure 5.7 is an example of data GIRO could integrate into their optimization algorithms. For the different lines and directions at the different parts of the day, the probability of the trip ending in the different delay slot would be displayed.

Table 5.7 Example of a possible Output with a Random Forest algorithm



TSDL	Prob_Class0	Prob_Class1	Prob_Class2	Prob_Class3
R100Est10	16%	80%	4%	0%
R100Est11	29%	66%	4%	0%
R100Est13	61%	33%	5%	1%
R100Est14	29%	47%	22%	2%
R100Est15	71%	19%	8%	2%
R100Est16	13%	55%	24%	7%
R100Est17	22%	53%	16%	9%
R100Est18	33%	39%	20%	8%
R100Est19	15%	49%	28%	8%
R100Est20	43%	32%	17%	8%
R100Est21	13%	71%	11%	6%
R100Est22	15%	50%	28%	6%

Random forest and artificial neural networks are more suitable algorithms to assess service reliability. Even if the random forest algorithm is faster and easy to train, the artificial neural network model has the potential to achieve better results.

Normal distribution assesses better service reliability in one period. However the model is built from the mean and the standard deviation of the delay, so the results are biased.

It would be interesting to predict the probability distribution with the model trained in previous periods. The results would be more relevant.

However, the method proposed to assess the service reliability, which consists in aggregating lines with direction and time slots could be generalized. Indeed, it affords to display results which could be used directly by the companies.

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

The objective of this research project was to use machine learning in a public transportation context. The main difficulty of the project was the exploratory phase. The conclusion of this step was the subject of the thesis : using scheduling data to assess service reliability. During the whole project, the idea of implementing the project was fundamental because the work was done in cooperation with a company : GIRO. Hence using offline data and the training time were essential aspects of the project. The main database was composed of data from Montreal for autumn 2017.

6.1 Summary of Works

First, the data were extracted and treated from different databases and aggregated all together. This work was done by integrating a primary key which was the UOTN. Then the database was processed to fit in a machine learning model. As the data come from a schedule, they were not correlated to each other, and they all could be integrated into machine learning models.

Then different machine learning models were implemented : Random Forest, Gradient Boosted tree and Artificial Neural Network. The three models were tested at the same time as a logistic regression, which was used as a reference. The models were first about binary classification and aimed at predicting the status of late of the bus end-trip. The models were optimized through the oversampling method, the dimensionality reduction and the choice of the parameters. In the end, the f1 score for the models at the end of the optimization was about 0.57.

Therefore, the models went to a multi-label classification, which aims at deciding in slot time of delay for bus end trips. The classes of delay were selected at first arbitrarily but were then optimized through unsupervised learning. A multi-label classifier with four classes determined by unsupervised learning obtained an f1 score of 0.51.

Moreover, using unsupervised learning to determine classes showed that the results are similar on different periods, and thus this method could be generalized. Indeed, deciding of the classes of delay with this method improved the results subsequently.

Then the autumn 2016 data were integrated into the database. Tests showed that machine learning could model end trip delay but the learning on one period and testing on other periods should be improved.

The schedule reliability was assessed by displaying the probability prediction for trips aggregated by line, direction, and time slot. The models were compared by calculating the probability distribution between reality and the probabilities distribution. A normal distribution is better when it is on one period, but experiments should be done on the future to check if this result is still valid when it is to predict future periods.

The model could not be compared because no other works try to predict the end trip delay using only the planning data. Indeed the prediction models tend nowadays to use real-time data in order to estimate the end trip delay. However, the model is better than the theory because it outperforms normal distribution and the supposed behavior of the models, which predict that 95% of the trips would end on time.

A summary of the different analysis is presented in figure 6.1

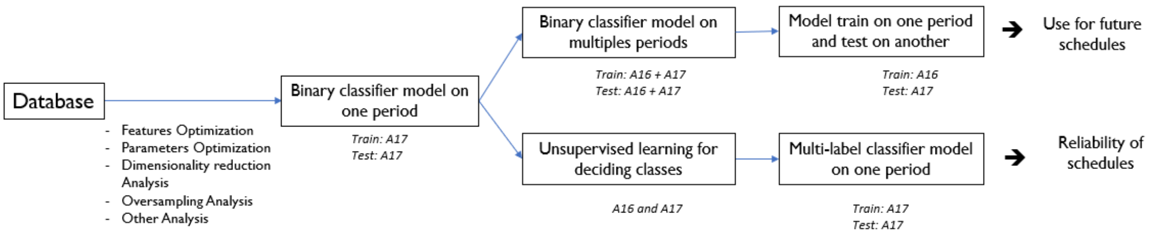

Database
- Features Optimization
- Parameters Optimization
- Dimensionality reduction Analysis
- Oversampling Analysis
- Other Analysis
Binary classifier model on one period
Train: A17
Test: A17
Binary classifier model on multiples periods
Train: A16 + A17
Test: A16 + A17
Model train on one period and test on another
Train: A16
Test: A17
Use for future schedules
Unsupervised learning for deciding classes
A16 and A17
Multi-label classifier model on one period
Train: A17
Test: A17
Reliability of schedules

Figure 6.1 Synthesis of the analysis

Random forests and gradient boosted trees achieve a higher f1 score for the models, and random forest needs less time for training. The artificial neural network obtained a smaller f1 score but achieved better to find delays. Artificial neural networks are supposed to achieve better, but the tuning of its hyper-parameters is difficult. Random forests could be used to predict models quickly and accurately, but if the time is not a constraint, for both the tuning of the hyper-parameters and for the training time, the artificial neural network should get better results.

Moreover, the method proposed in the service reliability assessment could be generalized.

Finally, this project has given GIRO an overview of the potential of the data they have for different machine learning projects.

6.2 Limitations

The model had some limitation, and the main one found in the problem was the fact of not using real-time data. It prevented the models from learning more. Indeed, weather condition or event if accidents could have helped to estimate the probability of delays.

There was the inherent problem of machine learning, which is the tuning of the hyper-parameters to improve the accuracy of the models. A compromised had to be found.

Moreover, the lack of data of the different years had prevented a total assessment of the model.

6.3 Future Research

Future research possibilities should solve the current limitations. Moreover, other theoretical and practical improvements could be suggested.

The first one could be the improvement of the models. It could be done by adding real-time data and by finding better hyper-parameters for the models. A new sklearn functionality had been developed and is AutoSklearn classifier.

Further analysis should be run to explain the difference of results when the model is trained on one period and tested on the other one. Finding a way of translating the input of one period to others is a real challenge. The method for assessing schedule reliability should be tested with a probability distribution and normal distribution trained on previous periods.

Moreover, the models could be tested in another city, to check that the models could be generalized. Other features could be found by doing this. With the output file of the models, a robustness index could be developed, giving to public transport planners an idea of the robustness of their schedules. In a more extended period, machine learning models could be directly included in scheduling algorithms and could be part of the objective function.

REFERENCES

[1] G.-J. Peek and M. Hagen, “Creating synergy in and around stations : Three strategies for adding value,” *Transportation Research Record*, vol. 1793, pp. 1–6, 01 2002.

[2] Wikipédia, “Montréal,” 22 mar 2019, https://fr.wikipedia.org/wiki/Montreal.

[3] Société de transport de Montréal. (2016) Programme des immobilisations, 2017 - 2026. https://www.stm.info/sites/default/files/pdf/fr/pi_17-26.pdf.

[4] “Apprentissage automatique,” 6 mar. 2019, https://fr.wikipedia.org/wiki/Apprentissage_automatique.

[5] G. Desaulniers and M. Hickman, “Public transit,” *Handbooks Oper. Res. Manag. Sci.*, vol. 14, pp. 69–128, 01 2007.

[6] Q. Deng and L. Cheng, “Research review of origin-destination trip demand estimation for subnetwork analysis,” *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 1485 – 1493, 2013, intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013).

[7] N. Paulley, R. Balcombe, R. Mackett, H. Titheridge, J. Preston, M. Wardman, J. Shires, and P. White, “The demand for public transport : The effects of fares, quality of service, income and car ownership,” *Transport Policy*, vol. 13, no. 4, pp. 295 – 306, 2006, innovation and Integration in Urban Transport Policy.

[8] R. van Nes, R. Hamerslag, and L. Immers, *The design of public transport networks.* National Research Council, Transportation Research Board, 1988, vol. 1202.

[9] S. B. Jha, J. Jha, and M. K. Tiwari, “A multi-objective meta-heuristic approach for transit network design and frequency setting problem in a bus transit system,” *Computers Industrial Engineering*, vol. 130, pp. 166 – 186, 2019, http ://www.sciencedirect.com/science/article/pii/S0360835219301111.

[10] A. T. Buba and L. S. Lee, “A differential evolution for simultaneous transit network design and frequency setting problem,” *Expert Systems with Applications*, vol. 106, pp. 277 – 289, 2018, http ://www.sciencedirect.com/science/article/pii/S0957417418302422.

[11] M. Owais and M. K. Osman, “Complete hierarchical multi-objective genetic algorithm for transit network design problem,” *Expert Systems with Applications*, vol. 114, pp. 143 – 154, 2018, http ://www.sciencedirect.com/science/article/pii/S0957417418304573.

[12] S. Bunte and N. Kliewer, “An overview on vehicle scheduling models,” *Public Transport*, vol. 1, no. 4, pp. 299–317, Nov 2009, https ://doi.org/10.1007/s12469-010-0018-5.

[13] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis, “Chapter 2 time constrained routing and scheduling,” in *Network Routing*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1995, vol. 8, pp. 35 – 139, http ://www.sciencedirect.com/science/article/pii/S0927050705801069.

[14] A. R. Odoni, J.-M. Rousseau, and N. H. Wilson, “Chapter 5 models in urban and air transportation,” in *Operations Research and The Public Sector*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1994, vol. 6, pp. 107 – 150, http ://www.sciencedirect.com/science/article/pii/S0927050705800866.

[15] N. Oort, “Service reliability and urban public transport design,” Ph.D. dissertation, Delft University of Technology, 01 2011.

[16] L. Redman, M. Friman, T. Gärling, and T. Hartig, “Quality attributes of public transport that attract car users : A research review,” *Transport Policy*, vol. 25, pp. 119–127, jan 2013, https ://www.sciencedirect.com/science/article/pii/S0967070X12001692.

[17] S. Elkosantini and S. Darmoul, “Intelligent public transportation systems : A review of architectures and enabling technologies,” in *2013 International Conference on Advanced Logistics and Transport*, May 2013, pp. 233–238.

[18] R. Bertini and A. El-Geneidy, “Generating transit performance measures with archived data,” *Transportation Research Record*, vol. 1841, pp. 109–119, 01 2003.

[19] T. R. Board, E. National Academies of Sciences, and Medicine, *Using Archived AVL-APC Data to Improve Transit Performance and Management*. Washington, DC : The National Academies Press, 2006, https ://www.nap.edu/catalog/13907/using-archived-avl-apc-data-to-improve-transit-performance-and-management.

[20] T. J. Kimpel, J. G. Strathman, D. Griffin, S. Callas, and R. L Gerhart, “Automatic passenger counter evaluation : Implications for national transit database reporting,” *Transportation Research Record*, vol. 1835, pp. 93–100, 01 2003.

[21] D. K. Boyle, “Passenger counting technologies and procedures,” pp. 9–14, 01 1998.

[22] S. Mukherjee, B. Saha, I. Jamal, R. Leclerc, and N. Ray, “Anovel framework for automatic passenger counting,” 09 2011, pp. 2969–2972.

[23] T. Yahiaoui, C. Meurie, L. Khoudour, F. Cabestaing, and F. Cabestaing, “A People Counting System Based on Dense and Close Stereovision A People Counting System Based on Dense and Close Stereovision. Abderrahim Elmoataz et al. 3rd International Confer-ence on Image and Signal Processing A people counting system based on dense and close stereovision,” *Lecture Notes in Computer Science*, vol. 5099, pp. 59–66, 2008, https ://hal.archives-ouvertes.fr/hal-00521106.

[24] J. Paul Bailly, J. Barry Barker, R. L. Barnes, G. L. Blair, A. Bonds, R. L. Brownstein, R. L. FREELAND Maryland MTA LOUIS J GAMBACCINI, C. Garber, S. GREENE Sharon Greene, A. Katharine Hunter-zaworski, J. H. Johnson, E. Lerner-lam, G. J. Linton, D. S. Monroe, P. S. Nettleship, J. P. REICHERT Reichert Management Services RICHARD J SIMONETTA MARTA PAUL P SKOUTELAS, P. Toliver, M. S. Townes, W. W. Millar Apta Kenneth R Wykle Fhwa John C Horsley, R. E. Skinner, and J. Trb, “TRCP oversight and project selection comittee ex officio members,” Tech. Rep., 2000, http ://www.nas.edu/trb/index.html.

[25] M. Mandelzys MASc Candidate, B. Hellinga, and P. Associate Professor, “Automatic identifying the causes of bus transit schedule adherence performance issues using avl/apc data,” Tech. Rep., 2009, http ://www.civil.uwaterloo.ca/bhellinga/publications/Publications/TRB 2010 Schedule Adherence Paper.pdf.

[26] F. Cevallos, X. Wang, Z. Chen, and A. Gan, “Using avl data to improve transit on-time performance,” *Journal of Public Transportation*, vol. 14, 09 2011.

[27] T. Van Oort and N. D. Yap, “CASPT 2018 Paper Automatic bottleneck detection using AVL data : a case study in Amsterdam, note = https ://pure.tudelft.nl/portal/files/48108501/CASPT_2018_paper_82.pdf, volume = 2018, year = 2018,” Tech. Rep.

[28] Y.-J. Lee, K. S. Chon, D. L. Hill, and N. Desai, “Effect of automatic vehicle location on schedule adherence for mass transit administration bus system,” *Transportation Research Record*, vol. 1760, no. 1, pp. 81–90, 2001.

[29] M.-P. Pelletier, M. Trépanier, and C. Morency, “Smart card data use in public transit : A literature review,” *Transportation Research Part C : Emerging Technologies*, vol. 19, no. 4, pp. 557 – 568, 2011, http ://www.sciencedirect.com/science/article/pii/S0968090X1000166X.

[30] T. Holleczek, S. Singapore, L. Yu, J. K. Lee, O. Senn, C. Ratti, and P. Jaillet, “Detecting weak public transport connections from cellphone and public transport data,” 2014, http ://dx.doi.org/10.1145/2640087.2644164.

[31] M. Berlingerio, F. Calabrese, G. D. Lorenzo, R. Nair, F. Pinelli, and M. L. Sbodio, “AllAboard : a System for Exploring Urban Mobility and Optimizing Public Transport Using Cellphone Data,” Tech. Rep., https ://www.ecmlpkdd2013.org/wp-content/uploads/2013/07/651.pdf.

[32] O. Cats, “The robustness value of public transport development plans,” *Journal of Transport Geography*, vol. 51, pp. 236–246, feb 2016, https ://linkinghub.elsevier.com/retrieve/pii/S0966692316000120.

[33] M. Friedrich, M. Müller-Hannemann, R. Rückert, A. Schiewe, and A. Schöbel, “Robustness Tests for Public Transport Planning,” no. 6, p. 16, 2017, http ://www.dagstuhl.de/16171.

[34] N. Oort, J. W. Boterman, and R. Nes, “The impact of scheduling on service reliability : Trip-time determination and holding points in long-headway services,” *Public Transport*, vol. 4, 07 2012.

[35] E. Diab and A. El-Geneidy, “Variation in bus transit service : Understanding the impacts of various improvement strategies on transit service reliability,” *Public Transport*, vol. 4, 03 2013.

[36] N. Oort and R. Nes, “Improving reliability in urban public transport in strategic and tactical design,” 03 2019.

[37] J. Parbo, O. Anker Nielsen, A. Landex, and C. Giacomo Prato, “Measuring Robustness, Reliability and Punctuality within passenger railway transportation-a literature review,” Tech. Rep., http ://www.trafikdage.dk/abstracts_2013/168_JensParbo.pdf.

[38] D. M. Scott, D. C. Novak, L. Aultman-Hall, and F. Guo, “Network robustness index : A new method for identifying critical links and evaluating the performance of transportation networks,” *Journal of Transport Geography*, vol. 14, no. 3, pp. 215 – 227, 2006, http ://www.sciencedirect.com/science/article/pii/S0966692305000694.

[39] M. Du, X. Jiang, and L. Cheng, “Alternative network robustness measure using system-wide transportation capacity for identifying critical links in road networks,” *Advances in Mechanical Engineering*, vol. 9, no. 4, p. 168781401769665, apr 2017, http ://journals.sagepub.com/doi/10.1177/1687814017696652.

[40] T. Büker and B. Seybold, “Stochastic modelling of delay propagation in large networks,” *Journal of Rail Transport Planning Management*, vol. 2, no. 1, pp. 34 – 50, 2012, http ://www.sciencedirect.com/science/article/pii/S2210970612000182.

[41] W. Feng and M. A. Figliozzi, “Identifying Spatial-Temporal Attributes of Bus Bunching through AVL/APC Data Using Archived AVL/APC Bus Data to Identify Spatial-Temporal Causes of Bus Bunching,” 2011, https ://www.researchgate.net/publication/261368125.

[42] B. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process : Theoretical basis and empirical results,” *Journal of Transportation Engineering*, vol. 129, pp. 664–672, 11 2003.

[43] M. Altinkaya and M. Zontul, “Urban bus arrival time prediction : A review of computational models,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 2, pp. 164–169, 01 2013.

[44] S. I.-J. Chien and C. Kuchipudi, “Dynamic travel time prediction with real-time and historic data,” *Journal of Transportation Engineering*, vol. 129, 11 2003.

[45] L. Weigang, W. Koendjbiharie, M. Jucá, Y. Yamashita, and A. Maciver, “Algorithms for estimating bus arrival times using gps data,” 01 2002.

[46] D. Sun, H. Luo, L. Fu, W. Liu, X. Liao, M. Zhao, D. Sun, H. Luo, X. Liao, and M. Zhao, “Predicting Bus Arrival Time on the Basis of Global Positioning System Data,” *Transportation Research Record : Journal of the Transportation Research Board*, pp. 62–72, 2034, http ://www.civil.uwaterloo.ca/itss/papers/2007-4 (Bus arrival time prediction using GPS data).pdf.

[47] P. Balasubramanian and K. R. Rao, “An Adaptive Long-Term Bus Arrival Time Prediction Model with Cyclic Variations,” Tech. Rep. 1, 2015, https ://pdfs.semanticscholar.org/6717/88dff66463c7fede54f8227f8d6a2c5e983c.pdf.

[48] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960, https ://www2.cs.duke.edu/courses/compsci527/cps274/fall11/papers/Kalman60.pdf.

[49] S. I.-J. Chien, Y. Ding, and C. Wei, “Dynamic bus arrival time prediction with artificial neural networks,” *Journal of Transportation Engineering-asce - J TRANSP ENG-ASCE*, vol. 128, 09 2002.

[50] J. Patnaik, S. Chien, and A. Bladikas, “Estimation of Bus Arrival Times Using APC Data,” Tech. Rep., https ://scholarcommons.usf.edu/cgi/viewcontent.cgi ?referer=https ://www.google.ca/&httpsredir=1&article=1337&context=jpt.

[51] F. Recknagel, “Applications of machine learning to ecological modelling,” *Ecological Modelling*, vol. 146, no. 1-3, pp. 303–310, dec 2001, https ://www.sciencedirect.com/science/article/pii/S0304380001003167.

[52] Z. Gurmu and W. Fan, “Artificial Neural Network Travel Time Prediction Model for Buses Using Only GPS Data,” *Journal of Public Transportation*, vol. 17, no. 2, pp. 45–65, jun 2014, http ://scholarcommons.usf.edu/jpt/vol17/iss2/3/.

[53] R. H. Jeong, “The prediction of bus arrival time using automatic vehicle location systems data,” Ph.D. dissertation, Texas AM university, 2004, https ://core.ac.uk/download/pdf/4268915.pdf.

[54] J. Amita, S. Jain, and P. Garg, “Prediction of Bus Travel Time Using ANN : A Case Study in Delhi,” *Transportation Research Procedia*, vol. 17, pp. 263–272, jan 2016, https ://www.sciencedirect.com/science/article/pii/S2352146516307062.

[55] Y. Zhang and A. Haghani, “A gradient boosting method to improve travel time prediction,” *Transportation Research Part C : Emerging Technologies*, vol. 58, pp. 308–324, sep 2015, https ://www.sciencedirect.com/science/article/pii/S0968090X15000741.

[56] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich, “Traveling time prediction in scheduled transportation with journey segments,” *Information Systems*, vol. 64, pp. 266–280, mar 2017, https ://www.sciencedirect.com/science/article/pii/S0306437915002112.

[57] Y. Bin, Y. Zhongzhen, and Y. Baozhen, “Bus arrival time prediction using support vector machines,” *Journal of Intelligent Transportation Systems - J INTELL TRANSPORT SYST*, vol. 10, pp. 151–158, 10 2006.

[58] B. Yu, Y.-L. Jiang, and Z.-Z. Yang, “Application of support vector machines in bus travel time prediction,” *Neural Network World*, vol. 34, pp. 158–160, 11 2008.

[59] M. Elhenawy, H. Chen, and H. Rakha, “Random forest travel time prediction algorithm using spatiotemporal speed measurements,” *21st World Congress on Intelligent Transport Systems, ITSWC 2014 : Reinventing Transportation in Our Connected World*, 01 2014.

[60] B. Yu, H. Wang, W. Shan, and B. Yao, “Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 4, pp. 333–350, apr 2018, http ://doi.wiley.com/10.1111/mice.12315.

[61] Y. Sun, Q. Yan, Y. Jiang, and X. Zhu, “Reliability prediction model of further bus service based on random forest,” *Journal of Algorithms & Computational Technology*, vol. 11, no. 4, pp. 327–335, dec 2017, http ://journals.sagepub.com/doi/10.1177/1748301817725306.

[62] M. Zorkany, M. Zaki, I. Ashour, and B. Hisham, “Online bus arrival time prediction using hybrid neural network and kalman filter techniques,” *International Journal of Modern Engineering Research (IJMER)*, vol. 3, 01 2013.

[63] A. F. Abidin, M. Kolberg, and A. Hussain, “Integrating sumo and kalman filter models towards a social network based approach of public transport arrival time prediction,” 2016.

[64] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-SMOTE : A New Over-Sampling Method in Imbalanced Data Sets Learning,” Tech. Rep., 2005, https ://sci2s.ugr.es/keel/keel-dataset/pdfs/2005-Han-LNCS.pdf.

[65] "Evaluation of binary classifiers," in *Wikipédia*, 08 apr. 2019, https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers.

[66] "Random forest classifier," in *Scikit-Learn*, 09 apr. 2019, https ://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[67] "Gradient boosting classifier," in *Scikit-Learn*, 09 apr. 2019, https ://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.

[68] "Multi-layer perceptron classifier," in *Scikit-Learn*, 09 apr. 2019, https ://scikit-learn.org/stable/modules/generated/sklearn.neuralnetwork.MLPClassifier.html.

[69] "Logistic regression," in *Scikit-Learn*, 09 apr. 2019, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[70] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE : Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, jun 2002, https ://jair.org/index.php/jair/article/view/10302.

[71] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li, "ADASYN : Adaptive synthetic sampling approach for imbalanced learning," pp. 1322–1328, jun 2008, http ://ieeexplore.ieee.org/document/4633969/.

[72] L. Van Der Maaten, E. Postma, and J. Van Den Herik, "Dimensionality Reduction : A Comparative Review," Tech. Rep., 2009, http ://www.uvt.nl/ticc.

[73] "Principal component analysis," in *Scikit-Learn*, 09 apr. 2019, https ://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

[74] D. M. Hawkins, "The Problem of Overfitting," 2003, https ://pubs.acs.org/doi/abs/10.1021/ci0342472 ?journalCode=jcics1.

[75] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451 – 461, 2003, biometrics.

[76] "K-means clustering," in *Scikit-Learn*, 09 apr. 2019, https ://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.