FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Intelligent and Autonomous Usage of the Energy Generation Systems in Vehicles

Kevin Carvalho de Amorim



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: João Pedro Mendes Moreira

Supervisor: Leonel Rocha Araújo

Supervisor: Luís Alexandre Moreira Matias

3rd July, 2019

© Kevin Carvalho de Amorim, 2019

Intelligent and Autonomous Usage of the Energy Generation Systems in Vehicles

Kevin Carvalho de Amorim

Mestrado Integrado em Engenharia Informática e Computação

3rd July, 2019

Resumo

O combustível é um dos maiores custos que as empresas de transportes possuem. No caso de veículos que transportam mercadoria que precisam de temperatura controlada o consumo de combustível é ainda maior, pois, geralmente, usam um motor a gasóleo inteiramente dedicado ao sistema de refrigeração. O uso deste motor adicional apresenta também custos adicionais em manutenção e maiores emissões de CO_2. Para reduzir, e eventualmente eliminar completamente, a necessidade de usar combustível para alimentar o sistema de refrigeração, a AddVolt desenvolve um sistema de gestão de energia com travagem regenerativa. Esta travagem regenerativa permite produzir energia durante a utilização do veículo. A energia produção provém da conversão de energia cinética em energia elétrica durante travagens e desacelerações. A energia produzida é armazenada em baterias que alimentam o sistema de refrigeração permitindo que este funcione em modo elétrico.

O sistema de gestão de energia recolhe informação acerca da utilização dos veículos, sendo esta informação enviada para os servidores da AddVolt. Os dados disponíveis abrem a oportunidade de desenvolver uma camada de inteligência para a operação do sistema de travagem regenerativa. Esta inteligência tem como objetivo tomar decisões acerca de quando aplicar, e quando não aplicar, a travagem regenerativa, dependendo da necessidade energética de cada viagem. Esta decisão irá permitir reduzir a utilização do motor a diesel para alimentar o sistema de refrigeração.

Para conseguir tomar uma decisão sobre quando aplicar travagem regenerativa, é necessário uma previsão do tempo que o veículo irá demorar a percorrer uma dada viagem. Com a previsão do tempo de viagem é possível estimar a carga da bateria em cada momento da viagem. Este documento irá apresentar uma possível solução para chegar a esta previsão, utilizando os dados disponíveis. Primeiro, todos os dados disponíveis passarão por um processo de limpeza de dados e discretização espacial, utilizando um método baseado em grelha. Um segmento de teste será então encontrado, analisando a cobertura de dados, no mapa, para encontrar o segmento com maior abundância, e variedade, de dados. Tendo o segmento de teste os dados, dessa mesma viagem, são agrupados por viagem. Os dados agrupados por viagem passarão por um processo de "features engineering" para desenvolver novas features que irão melhorar o modelo final de previsão. Finalmente, alguns modelos de previsão serão testados, de forma a encontrar o que melhor se adapta ao problema. A previsão final será então utilizada para tomar decisões.

Abstract

One of the largest costs transportation companies have to deal with are fuel related. In the case of vehicles that transport refrigerated goods the fuel consumption is even greater, because they tipically have an engine entirely dedicated to feed the refrigeration unit (RU). The usage of this diesel engine presents also additional costs in maintenance and greater CO_2 emissions. To reduce, and eventually completely eliminate, the need for fuel to feed the RU, AddVolt develops an energy management system (EMS) with regenerative braking (RB) that allows to produce energy during vehicle operation. The produced energy comes from the conversion of kinetic energy to electrical energy during desaceleration and braking, by the RB. The produced energy is stored in batteries that feed the RU allowing it to run in electrical mode.

The EMS collects information about vehicle operation which is sent to the AddVolt's cloud server. This available data opens the opportunity to develop an intelligence layer for the operation of the RB. The intelligence layer aims to make decision about wether to apply, or not apply, RB depending on the energy demand during each given trip. This decision would then allow to reduce the dedicated diesel engine usage to feed the RU, ideally running in full electrical mode.

In order to decide when to apply RB a prediction of the time the vehicle will take to complete a given trips is needed. With this travel time prediction, the predicted battery state-of-charge (SOC) can be estimated for each moment of the trip. This document will present a solution to arrive at this prediction, using the available data. First, all data available is properly cleaned and then a spatial discretization is made using a grid-based method. A testing road segment is then found by analyzing the map coverage to find the road segment with the most abundant amount of data. Having the test road segment, data from this same road segment is discretized by trip. The trip discretized data goes through a process of features engineering to develop new features that will improve the final prediction model. Finally, some prediction models are tested in order to find the one that best fits the problem. The resulting travel time prediction is used to make a decision.

Acknowledgements

The process of research, work and writing of a dissertation is a great challenge, which is not possible to fulfill individually.

I would like to give my sincere thanks to AddVolt team for providing an excellent environment on which to work on this thesis and for all the expertise, specially to Leonel Araújo who closely followed the entire process of this work, always being fully available to help and review.

I must also mention João Mendes Moreira, PhD and Luís Alexandre Matias, PhD, who also supervised all of this work, and greatly helped in the entire process, and for that, my sincere gratitude.

Finally, I want to express my sincere gratitude to my parents and my sister who fully supported me throughout all these years.

Kevin Amorim

"Aim for success, not perfection.
Never give up your right to be wrong, because then you will lose the ability to learn new things
and move forward with your life."

David M. Burns

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Goals . . . 2
- 1.3 Document Structure . . . 2

2 Literature Review **5**
- 2.1 Data Mining - What is it and what it serves for? . . . 5
  - 2.1.1 Supervised Learning . . . 6
  - 2.1.2 Unsupervised Learning . . . 7
- 2.2 Time Series Analysis . . . 9
  - 2.2.1 Floating Car Data (FCD) . . . 10
  - 2.2.2 Short-Term Traffic Flow Estimation and Prediction . . . 10
  - 2.2.3 Travel Time . . . 11

3 Problem **13**
- 3.1 Proposed Solution . . . 14

4 Data Collection and Pre-Processing **19**
- 4.1 Pre-Processing . . . 19
  - 4.1.1 Missing Data . . . 19
  - 4.1.2 Reliability . . . 20
  - 4.1.3 Accuracy . . . 21

5 Selection of Test Road Segment **23**
- 5.1 Spatial Discretization . . . 23
- 5.2 Trip Discretization . . . 24
- 5.3 Sequential Pattern Mining . . . 25

6 Travel Time Prediction **29**
- 6.1 Features . . . 29
  - 6.1.1 Data Grouping . . . 29
  - 6.1.2 Features Building Using Domain Knowledge . . . 30
- 6.2 Prediction Model . . . 30
  - 6.2.1 Model Evaluation . . . 32
  - 6.2.2 Linear Regression . . . 33
  - 6.2.3 Stochastic Gradient Descent Regressor . . . 33
  - 6.2.4 Random Forest . . . 33
  - 6.2.5 Summary and Results . . . 33

7 Decision Support Heuristic **35**
7.1 Ranking Cells by Energy Potential 35
7.2 Final Decision 36
7.3 Summary and Results 36

8 Conclusions **39**
8.1 Future Work 39
8.1.1 Additional Road Characteristics 40
8.1.2 Spatial Discretization - A Different Approach 40
8.2 Online Adjustment 40

References **43**

List of Figures

2.1 Example of a linear regression. 7
2.2 Example of clustering, with 3 clusters. 9

3.1 Truck with an EMS, RB and RU schematic. 14
3.2 Flow of the kinetic energy from the wheel, to the shaft and finally to the generator motor. 15
3.3 Proposed Solution Architecture 16
3.4 Travel time scatter plots by hour of the day and day of the week. 17

4.1 Example of a temporary lost signal during a trip. Since, the signal was lost for less than 61 consecutive seconds the trip was not cut. 20
4.2 Rectangular bounding box of where the data is expected to be present. 22

5.1 Data Coverage 24
5.2 Number of unique vehicles travelling, for each cell. 25
5.3 Example of a trip finishing with a parking maneuver. 26
5.4 Biggest trip found inside the used bounding box. 27
5.5 Bounding box and selected segment 28

7.1 Decision flow chart. 36

8.1 What3words usage example. 41

List of Tables

6.1 Table of features used. 31
6.2 Weather features used. 32
6.3 Results obtained with each different algorithm. 34

7.1 Results obtained with the decision support heuristic. 37

8.1 Example of additional features, related to road characteristics. 40

Abreviaturas e Símbolos

AVL	Automatic Vehicle Location
ATT	Arrival Time-Based Link Travel Time
DM	Data Mining
DTT	Departure Time-Based Link Travel Time
EMS	Energy Management System
FCD	Floating Car Data
ITS	Intelligent Transport System
MAE	Mean Absolute Error
MAPE	Mean Average Percentage Error
MSE	Mean Squared Error
ML	Machine Learning
RB	Regenerative Braking
RMSE	Root Mean Squared Error
RU	Refrigeration Unit
SGD	Stochastic Gradient Descent
SOC	State-of-Charge
SVM	Support Vector Machine
TTP	Travel Time Prediction
TTV	Travel Time Variability
TSDM	Time Series Data Mining

Chapter 1

Introduction

In this chapter a brief introduction to the problem is made. The economical and environmental impact the solution may have is explained in the context. The business needs and room for improvement is approached in the motivation section. Finally an overview of the expected goals and document structure is made.

1.1 Context

One of the largest fixed costs that transportation companies need to deal with are fuel related. According to recent studies, fuel comprises 40% of the total operational costs of transportation companies with trailer-based heavy trucks [29]. Although the average diesel price fluctuates a lot, its trend has been upwards, which brings more concern to reduce diesel consumption.

The transportation sector is also one of the main responsibles for the greenhouse gasses emission, 34% in 2012, having a big impact in the quality of our environment [33]. In order to mitigate this problem, many countries are announcing the ban of fossil fuel vehicles. To achieve this they are starting to forbid high emissions vehicles in big cities and/or increasing taxes to the owners of such vehicles [13].

The transportation sector has a high demanding subsector, transportation of refrigerated goods. It was estimated to be over 1 million refrigerated road vehicles worldwide, in 2002 [3]. Power consumption of the refrigeration unit (RU) is very high and for most the energy needed comes from a diesel engine (be it the main traction engine or a dedicated diesel engine). Inevitably there is an increase in fuel consumption and maintenance, making these vehicles more costly to operate, more polluent and more noisy. From a survey made about different weight, distribution type and operating temperature vehicles, it was estimated that, on average, 37% of the total energy consumed came from the RU [32].

A solution proposed by AddVolt - this thesis host institution - is the implementation of an Energy Management System (EMS) aboard refrigerated trucks. The EMS encompasses a regenerative braking (RB) system, a battery pack, a communication system and all the required controllers to operate. In short, it can be seen as a high-power and high-capacity intelligent power bank to

supply heavy energy loads. The RB system is an optional add-on to the EMS which is installed on vehicles that require greater autonomy or when the client prefers to recharge the battery while running instead of plugging it into the grid. This system consists in converting kinetic energy into electrical on vehicle braking or deceleration, rather than wasting it, being dissipated as heat [5]. The recovered energy is then stored in the battery pack that can be used to allow the RU to run in electrical mode. Alternatively, the batteries energy can be charged by the grid. Finally, operations metrics of the EMS, the RU, the vehicles and their accessories are exchanged through a secured communication channel with AddVolt server. On the system side, the communication is ensured via a SIM (Subscriber Identity Module) card with GSM (Global System for Mobile Communications). The received information comes in the form of time-series (series of data points in chronological order), in real time.The exchanged data allows, among other tasks, to perform predictive maintenance and feed machine learning algorithms on traffic and routing.

1.2 Goals

The EMS is responsible for feeding the RU. However its energy demand is highly unpredictable due to its dependency on different conditions. The type of transported goods will require different operating temperatures. Those operating temperatures can be divided in two groups: fresh and frozen. Since for frozen goods the operating temperature is lower, the energy demand will be higher than for fresh goods. Also traffic congestion, traffic signals in intersections, road type and weather conditions may oblige to longer trips, increasing the energy demand.

The goal of this dissertation is to improve the intelligence layer of the system. One way of achieving this is predicting the battery autonomy and distribution demand at each moment of a given trip. For this prediction several variables will be used such as departure time, road path to be travelled, stopping points and RU temperature. Predictions about traffic at each instant will also be fundamental to generate accurate models. Also, daily, weekly and monthly seasonalities will be taken into consideration. The predictions made will then be used to optimize the intelligence of the system, that will use the received variables to better manage the effort applied in both the vehicle and the RB system. Using predictive models for the expected trip development, such as travel time and traffic, the system may choose to apply regenerative braking. This decision will also be based on the minimization of the wear of the system and according to the preferences of the client.

1.3 Document Structure

This document will contain the entire process of the study for a solution proposal for the described problem, improving the EMS RB system intelligence layer.

In chapter 2 a literature review is made, starting with basic concepts of data mining (DM) 2.1. Also, different types of learning algorithms are introduced, supervised 2.1.1 and 2.1.2. It is followed by a short review on time series 2.2 and floating car data 2.2.1. More importantly most

of the literature is focused on travel time prediction 2.2.3 and traffic flow prediction 2.2.2, being those the main focus of the work.

In chapter 3 a more in-depth description of the problem is made and its respective solution proposal will be present.

Chapter 4 shows an overview of how data collection and pre-processing is made, in this work. Followed by chapter 5 which presents the method of selecting the test road segment. In chapter 6 the steps to predict travel time, and evaluate the results, are defined. Using the travel time prediction, a decision support heuristic is developed, in chapter 7.

Finally, chapter 8 contains the conclusions of this thesis, including future work ideas.

Chapter 2

Literature Review

In this chapter an overview of the literature review is made. The concepts of data mining, what it is and what it serves for (Section 2.1), are introduced, as well as its steps and the concept of machine learning (ML). Followed by an introduction on two types of learning: supervised (Section 2.1.1) and unsupervised (Section 2.1.2) learning, and examples of tasks for each type.

A review on time series analysis (Section 2.2) is then approached, since it will be a fundamental problem for the scope of this work. It is then complemented with a review on some relevant application fields and floating car data (Section 2.2.1), a type of time series. The contemplated application fields are: short-term traffic flow (Section 2.2.2) and travel time (Section 2.2.3) predictions, both being relevant for this work to be complete.

2.1 Data Mining - What is it and what it serves for?

A large amount of data is available for analysis and with an ever growing amount of information, the opportunity to find useful patterns in data through Data Mining (DM) arises. According to [24] and [21] DM is the science of finding non-trivial, valuable and unique information from a large size of datasets and databases, with two primary goals: find new insights and generate predictions. We will focus on the latter.

According to [12] the extraction of knowledge with DM follows an iterative sequence of steps. Those steps include:

- Data Cleaning: Removing outliers and inconsistent data;
- Data Integration: Combining different data sources;
- Data Selection: Selecting relevant data from all data;
- Data Transformation: Transforming data into appropriate dataset for mining;
- Data Mining: Applying intelligent methods to extract patterns in data;
- Pattern Evaluation: Measuring patterns using measures relevant to the task;

- Knowledge Presentation: Presenting results in readable ways for final users;

Some of these steps will be further explored in this work.

An example presented by [22] shows how data mining with machine learning (ML) is utilized for fleet management. With the growing of transportation companies, fleet management techniques are growing in order to predict future maintenance and track driver behaviour. To achieve this, real-time data from tracking devices and on-board vehicle diagnosis devices, as well as historical data is used to provide condition based maintenance and other fleet management information specific to the driver. The FCD obtained by these companies, grow with each vehicle and each trip performed, being data cleaning a critical point to clean unusable data, errors and outliers. Also, techniques of data integration must be applied in order to combine data from each individual vehicle into a single database. Having all data collected, integrated and cleaned, a selection for relevant data must be applied, to isolate data useful to the problem to be solved. Finally, the data is transformed into appropriate datasets for mining (data transformation) and patterns in this data are extracted and evaluated to be presented in a readable format for the final user. To make decisions based on all collected data, and improve the decisions taken over time, techniques of ML are applied to the collected and mined data.

ML studies how computers can learn to detect patterns and make intelligent (predictions and recommendations) decisions based on data [12]. These algorithms may adapt to new data in order to improve its efficiency over time.

There are different problems in ML, related to data mining, from which only supervised learning 2.1.1 and unsupervised learning 2.1.2 are presented in this work.

2.1.1 Supervised Learning

Supervised learning consists in using training data and human feedback to learn the relationship between a given input and its output [1]. It is usually seen as a classification problem [12]. The input data is labelled and the output variable is the labelling result, defined a priori, by a human. The algorithm uses this dataset to learn the connection between the input and the output in order to get sufficiently accurate to be applied in new unknown datasets. Some examples of algorithms that follow the supervised learning paradigm are: algorithms of regression (linear, ridge, lasso), algorithms of decision trees (like, C4.5, ID3, CART), random forest, adaptive boosting (AdaBoost), support vector machine (SVM), etc.

Following the example given in 2.1, using historical data to predict vehicles maintenance based on drivers behaviours may be a type of supervised learning. Using known input (driving metrics) and the resulting output (tracking of past maintenances) a model can be built to predict future maintenances based on new, unknown, datasets.

2.1.1.1 Classification

Data classification consists in developing a classification model that will then be used to predict class labels for given data [12]. The classification model is built using a training dataset that its

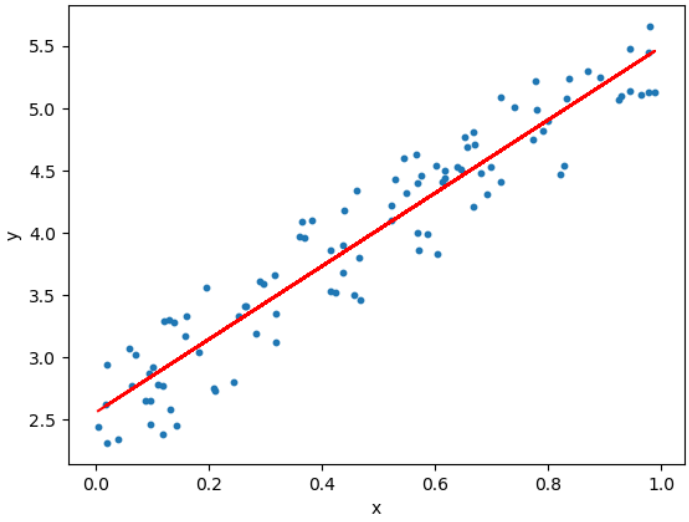


Figure 2.1: Example of a linear regression.

previously known class labels. The first goal is to learn the features that distinguish one class from the other [8]. The mapping between given data and corresponding class label is made using classification rules, decision trees or mathematical formulas. A testing data set is then used to test the classification model in order to evaluate its accuracy. If the accuracy is considered acceptable for the problem at hand the model can be used to classify new data [12].

There is a comparison made by [31] that compares different types of time series classifiers, focusing on: SVM, nearest neighbor and decision forest. However neither of them could be considered better than the other, it greatly depends on the given dataset.

2.1.1.2 Regression

When the expected output for a given input is not a label (class), but a numeric value, a regression approach can be used. According to [1], in regression, we try to learn a numeric function $f(x)$ that passes through points from a training set. This same function is then used to predict an output for a new, unknown, input. See figure2.1

2.1.2 Unsupervised Learning

An algorithm explores input data without any given expected output. According to [1] it is useful to find, for example, patterns in the data. It is mostly used when there is no specific classification for a given dataset and there is a need to use an algorithm to classify the data. It is usually seen as a clustering problem [12]. Clustering may be used, according to [12], to discover classes within a given dataset.

The potential of unsupervised learning in customer relationship management is demonstrated by [1]. Companies may have different sources of customers information, which after proper integration and cleaning, may be used to extract new, valuable, information about customers and market. After a process of selection of relevant data, customers can be grouped according to shared characteristics. No prior output is expected with the analysis of this data - unsupervised learning - but interesting outliers and new niches can be found with the final process of data mining. This new learned information may be used to improve marketing campaigns, provide a more customized customer service or present interesting information about the company - pattern evaluation and knowledge presentation steps of data mining.

2.1.2.1 Frequent Patterns Mining

As unsupervised learning technique, we use, in this work, the frequent pattern mining. As suggested by [12] this techniques finds frequent patterns in data, as frequent itemsets and frequent sub-sequences (sequential patterns). Following the previous example, in section 2.1.2, finding products that are commonly bought together would be a frequent itemset. The sequential pattern that customers follow buying some products, as buying first a phone, then a phone case and later a memory card, would be frequent sub-sequence. The frequent patterns mining may, then, reveal interesting correlations within data, as, for the scope of this work, finding the most frequent road segments.

According to [15] the sequential pattern mining problem was first introduced as "Given a set of sequences, where each sequence consists of a set of items, and given a user-specified *min_support* threshold, sequential pattern mining is to find all of the frequent subsequences, i.e, the subsequences whose occurrence frequency in the set of sequences is no less than min _support." A great challenge with sequential pattern mining, also noted by [15], is the number of possible sequences to examine, which may exponentially grow with the size of the dataset. The solution used for this work is the usage of a sequential pattern mining method called PrefixSpan (Prefix-projected Sequential Pattern Mining). [15] defines PrefixSpan as having its main idea on basing its projection on frequent prefixes, instead of all possible occurrences of frequent subsequences, since any frequent subsequence can always be found by growing a frequent prefix.

2.1.2.2 Clustering

Finally, clustering is one of the main tasks of unsupervised learning, and this section would not be complete without a brief introduction to this task.

As mentioned in 2.1.2 clustering consists in finding natural groups within a dataset. This clustering has as objective minimizing differences inside cluster (intracluster) and maximizing differences between clusters (interclusters) [8]. In time series, clustering can be divided, according to [8], in two sub-tasks: whole series clustering and subsequence clustering. In whole series, clustering is applied to each complete time series in a set. In order to regroup entire time series into

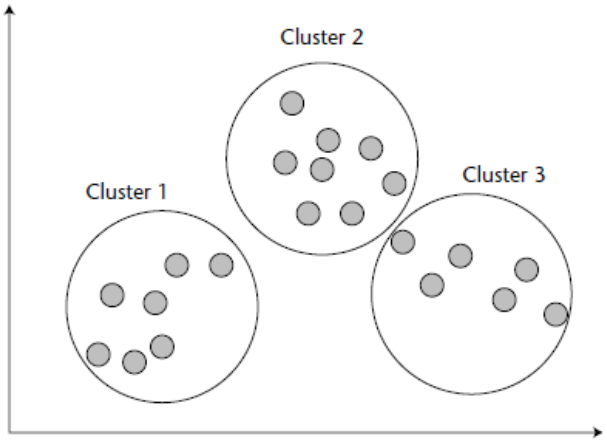


Figure 2.2: Example of clustering, with 3 clusters.

clusters, keeping all similar time series within the same cluster. In the case of subsequence clustering sub-sequences are extracted from longer time series according to non overlapping windows. A periodical structure of the time series should be detected to correctly define a width window. If no periodical structure is found some information may be lost while slicing the time series. This last sub-task can be applied to detect attacks within network traffic data or for DNA analysis [8].

2.2 Time Series Analysis

Time series are, in its core, data collected continuously over time in chronological order and, usually, collected in equally spaced intervals. They are normally large in size and considered as a whole [9]. And with the ever growing amount of available data it requires a constant development of algorithms so it is possible to fetch useful information quickly [8].

Pattern discovery is considered by many, one of the main mining tasks related to time series. Its main goal is to discover useful patterns in data that are no so trivial to find without deeper analysis. These patterns may include useful and important trends in the collected data that may further improve the interpretation made with the available data [9]. When analyzing a time series, the most important step, according to [30], is detrending. In the case of urban mobility, which will be important for the development of this work, there are recurring cyclic trends. Those trends usually follow daily and weekly patterns. It is then necessary to remove those trends from the original time series in order to forecast based on the residuals [30].

Some examples, provided by [8] and [35], about real-life problems where time series are applied may be: economic forecasting, earthquake prediction, intrusion detection, communication engineering or gene expression analysis. A study by [7] shows that there has been a constant development of models to work towards accurate stock market forecast, based on time series, using linear regressions and neural networks, for instance. Time series analysis can also be applied in intrusion detection on systems, as presented by [6], which uses neural networks to recognize

unfamiliar actions patterns that deviates from the user normal behavior. Finally, more directly related to this work, there are many applications of time series analysis in transportation. For example, travel time estimation ([38], [16], [19], [14], amongst others), traffic flow prediction ([23] and [37]) and taxi passenger demand forecast [26].

2.2.1 Floating Car Data (FCD)

In this work we use a specific type of time series: floating car data. This data is collected continuosly in, ideally, fixed time frames, in order to provide information about the traffic network.

Due to the need of improving traffic management many traffic data collection methods emerged, ranging from intrusive to non-intrusive. Some examples of those methods are: pneumatic road tubes, magnetic loops, manual counting, infrared and video image, as presented by [18] and [2] . More recently probe vehicles are being used to collect traffic information [2].

In addition to the previous methods floating car data (FCD) is a source of high quality data that can be an alternative or a complement to the traditional sources [18]. FCD is defined as data collected from global positioning system (GPS) or via mobile phone obtained from road vehicles [18] [4]. Since FCD is collected directly from the vehicles instead of fixed sensors, it is possible to obtain data for, virtually, the entire trip of a given vehicle. Having enough coverage it would be possible to have relevant information about entire road networks. This information has large application in intelligent transportation systems (ITS), such as: travel time and traffic flow prediction.

However, data collected from GPS or mobile phone is highly susceptible to noise, which can give undesirable results to any analysis made [4]. So, it is important to pre-process the data and evaluate the quality of the available FCD before proceeding to a more in-depth analysis. A set of metrics presented by [4] helps accomplish the evaluation: value (granularity and coverage) and veracity (missing data, reliability and accuracy). The granularity indicates the frequency of the received signals. On the other hand, coverage gives us an overview of how well is the data distributed. On a macro temporal (covered days in timespan - ideally all days of the week), micro temporal (coverage within a single day) and spatial (map coverage and its density) dimensions [4].

Not only is important to know the available data and its evaluation its also crucial to pre-process it in terms of veracity. GPS signal may be temporarily lost causing gaps in the dataset (missing data), incorrect data may be sent by the GPS leading to unrealistic positions and/or speed (reliability) and also the precision (accuracy) has to be taken into account [4].

2.2.2 Short-Term Traffic Flow Estimation and Prediction

Having enough FCD it is possible to estimate and predict the traffic flow, for a given day and hour. Altough there are possible special events (like traffic incidents, road works, special events, etc.) that may change the normal trend for traffic congestion, it is possible to find hourly and daily patterns in traffic flow. An overview of this topic is made below.

Traffic congestion is a critical problem that greatly affects economy and daily life. Because of its complexity, constant changing nature and its social and economic impact there is a lot of work towards traffic management [17]. A solution proposed by [28], to define the congestion level of a given segment, is using vehicles velocity (collected from GPS, odometer or video cameras). The instantaneous velocity of the vehicle is smoothed using weighted exponential average. Then, the smoothed velocity is classified in "green", "yellow" or "red" according to given thresholds. Finally, transient congestions are eliminated from the final congestion level calculation, also according to a given threshold. The proposed algorithm classified correctly, according to human perception, approximately half of the time. However, this value may differ for different locations and different parameters definition. Nonetheless it is an interesting approach when only using velocity as input data.

A more in-depth methodology to classify traffic congestion is explained by [41]. In this case, different features are analyzed: speed (average and distribution), spatial-speed, temporal-speed and spatial-temporal-speed, being the last (spatial-temporal-speed feature) the one that takes into account most particularities of road segments and GPS data, and yields the best results.

Two main groups of short-term traffic flow prediction methods, are considered by [23]: based on multi-source data and single-source data. For the scope of this work we will be focusing on single-source data. Multi-source data may yield interesting data, but it still present some problems, such as normalizing the different temporal scales and dimensions. Short-term traffic flow prediction models may be linear or nonlinear, to which the main difference is in the time/complexity it takes to train the model. Nonlinear models are the most complex, time consuming and that requires the largest dataset [23].

Different methodologies have been studied for short-term traffic flow prediction, such as: multivariate time-series model, Kalman filtering, nonparametric regression model and neural networks. However, the most appropriate is using hybrid models, combining different models for different patterns [39]. A solution proposed by [39] combines two neural network predictors: back propagation and radial basis function into a Bayesian combined neural network. This solution shows considerable improvement in comparison with single neural networks.

Essentially, a week can be considered a cycle, in terms of traffic flow [23]. In regular situations it is expected to experience similar peak hours and, thus, similar flow for the same hour of the same day of the week. Assuming this situation it is possible to make predictions using two dimensions: immediate past data (the n past timestamps collected) and the available historical data that shares the same characteristics of the present moment (same day of the week/same hour).

2.2.3 Travel Time

Traffic congestion is getting worse every year, especially in urban areas, due to the growth in population in those areas. As a consequence travel time and finding ways to predict and even reduce it has become a big concern. In fact, the ability to predict travel time of a route is a fundamental feature of most modern navigation applications [40].

Most intelligent transportation technologies need reliable prediction of travel time [42]. The best and simpler definition of travel time comes from [34] and it is: "...time required to traverse a route between any two points of interest...". It will be the definition we will be using for the basis of this work. However, as pointed by [36], travel time is the result of stochastic and non-linear interactions between drivers, making the prediction process complex.

Travel time is affected by a lot of variables, such as: freeflow travel speed, road geometry, traffic flow, traffic signals, weather conditions and even holidays and special incidents [20], [11] and [10]. An application of a Kalman filtering methodology is presented by [14] to predict travel time based on real-time and historic data. The main advantage pointed by the same authors, on the Kalman filtering, its the continuous update of the state variable with new observations.

An interesting separation between arrival time-based link travel time (ATT) and departure time-based link travel time (DTT) is presented by [25]. ATT would be the travel time associated with the arrival time at the destination, simply put, the difference between the arrival time and the departure time. In the other hand, DTT is the expected travel time, given a departure time but no arrival time is available. So, to have a reliable DTT prediction it is important to have an accurate ATT estimate. One of the main solutions is a route-based solution. The travel time is given by the sum of the travel time through each road segment and the time spent in intersections. Following this solution, the problem of travel time estimation is divided in two sub-problems: estimation of the time needed to travel a given road segment and the time spent in each intersection [38].

There are two main problems of travel time prediction (TTP), according to the prediction horizon, short-term and long-term prediction. Both problems differ greatly, and there is a lot more work done for the short-term TTP. However both share a common approach to its solving, Regression. From the available types of approach to solve TTP problems, time series models will be used for short-term prediction.

Using real-time data for TTP proves to be more effective [14]. However, it also results in higher standard deviations from punctual events that may occur or biased results by missing data. Such problems will be addressed in the next sub-chapter.

2.2.3.1 Outliers

To achieve a correct, and useful, travel time estimation the outliers - extreme values caused by sampling bias - must be detected and removed from the dataset. However, a simple fixed-range outlier filtering method is not suitable [25] because of situations such as traffic congestions.

A simpler way of removing outlier is suggested by [14]. A record with a travel time highly deviated from the average link travel time is discarded. In the cited paper, the author discards records that deviates in 50% (higher or lower). This reduced the standard deviation in approximately 90%, for that specific case study.

Chapter 3

Problem

AddVolt's equips RU-enhanced lorry vehicles with an advanced EMS. As referred in the introduction, 1, the EMS encompasses a battery pack - which feeds the RU system - a RB system which allows to recharge the batteries on-the-road and a communication system that provides operation metrics, in time-series format, to AddVolt's servers. See Fig.3.1 for details.

The combination of the EMS and the RB is where the developed intelligence layer will thrive. The RB, in essence, consists in converting the wasted kinetic energy, while braking, to electric energy that can be stored in the batteries. The truck's axle has accopled a magnetic generator. At cruising speed, this generator has no load. However, during braking time, the generator gets activated by the natural motion atritum. See figure 3.2 for more details. The resulting energy is then converted to electric energy and stored in the batteries. This converted energy would be otherwise dissipated as heat. Additionally, since the RB system helps to brake the truck, there is less power applied to the brakes, thus reducing their wear. The intelligence layer developed in this work will help optimize the RB usage.

Different operation metrics are collected and transmitted by the EMS, which will be enumerated in section 3.1. As a result of the dimension of the number of available variables and the real-time data feed to the server, the collected data cannot be analyzed by hand, opening the possibility to apply data mining. Given all possible variations, resulting from both the dimension of the data and the stochastic nature of traffic, we cannot guarantee, in a trivial way, that the vehicle will have sufficient electric energy to last until the next charging point/opportunity or the end of the trip. And, although a regenerative braking system is installed and can recharge the battery while on the road two main problems must be mitigated:

1. If the driver does not need to brake during the trip, the energy demand may not be satisfied;

2. If too much regenerative braking is applied we may have unnecessary wear to the system, greater diesel consumption and wasted energy.

The problem will then consist in, using ML, determine when to apply RB, in order to optimize its usage. As a result, it is expected to, using the predictions, minimize the diesel usage to feed the

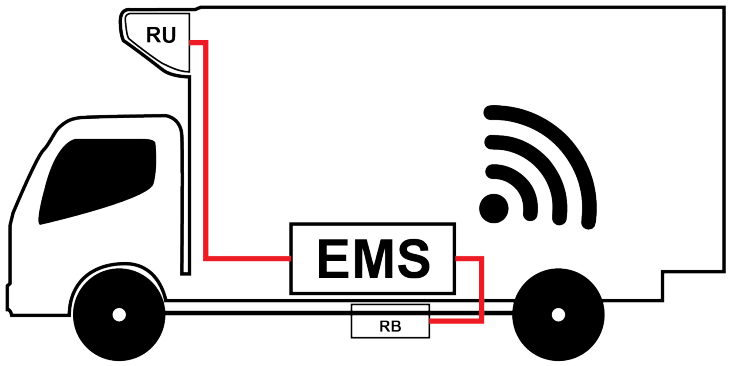


Figure 3.1: Truck with an EMS, RB and RU schematic.

RU (ideally, not using it at all) and minimize wear on the RB (by minimizing its usage) for each trip, by choosing the ideal braking points when needed.

3.1 Proposed Solution

The proposed solution consists in improving the EMS intelligence layer with basis on data collected over time. ML concepts are applied to data in order to reveal patterns and extract useful information from data. This data processing allows the EMS to make intelligent decisions about whether to automatically apply RB in strategic points of the trip.

The operation metrics collected and transmitted by the EMS include:

1. Timestamp
2. Geo-location: latitude, longitude and altitude coordinates
3. Vehicle operation: Speed measured by GPS, speed recorded by the vehicle odometer, pressure applied to both accelerator and brake pedal, engine's rotations per minute (RPM)
4. Battery state-of-charge (SOC)

Using these variables we aim to predict the travel time and, consequently, the battery autonomy at each instant of a given trip. As baseline, the prediction horizon will be short-term (less than 60 minutes). Most of this solution revolves around having enough battery autonomy to complete the whole trip, without using the diesel engine to feed the RU.

According to the problem definition, 3, the system will be divided in the states represented by Fig.3.3. A training dataset will be used to select a study segment, after all necessary data cleaning. Using the resulting most common segment and the same dataset a travel time prediction model will be built. Finally, the resulting model and a testing dataset will be fed to the heuristic in order to perform the final decision: when to apply RB.

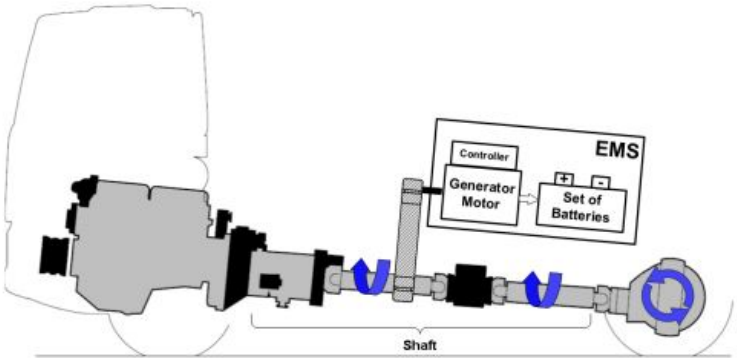


Figure 3.2: Flow of the kinetic energy from the wheel, to the shaft and finally to the generator motor.

To reach all needed variables a thorough data cleaning has to be made, since the signal is highly dependent on network coverage and may be temporarily lost during acquisition, returning invalid data, that may compromise the predictions made. Also, the dimension of the available data will require well optimized algorithms. Furthermore, seasonalities and trends will have to be studied, since, for instance, it is expected that the traffic on a Monday morning is a significantly different from the traffic on a Sunday morning, see figure 3.4 for travel time trends examples. Several variables need to be inferred since they are not directly available in the dataset. The collected data will then be combined with weather information, obtained from an external API. Using the available data, some new features are built, which will be defined in the features engineering section, 6.1, of this document. For this prediction it will also be taken into account factors that may affect travel time such as: length of the road, number of intersections with traffic lights, traffic flow/pattern (peak hours, e.g.), traffic incidents, road type (freeway, urban, industrial, etc.), weather conditions (mainly rain), driver behavior and special events that may affect the normal traffic. Predicting the travel time, and knowing, *a priori*, the battery usage (depending on the RU operating temperature) one can calculate the battery autonomy needed to finish the trip. If there is not enough battery, a decision to apply braking to the vehicle, in order to use the RB technology to recharge the battery, may be taken. For instance, if we expect to spend a long time in a highway with no traffic, on which the driver would rarely need to brake the vehicle, we may not have enough battery to feed the refrigeration unit. In this case, the braking system would automatically actuate, with low braking power, regenerating energy back to the battery. This last decision would be seamless to the driver. In the other hand, if the energy is dropping to low levels, but the driver is expected to need to brake no automated braking would be applied.

Finally, the decision made using the predicted travel time, to optimize the RB system, will be a binary decision on whether the RB system should be applied or not, in the current cell. Each cell of the trip is ranked by energy potential, based on: mean speed, slope, distance and mean number of brakes. This heuristic will prioritize high ranked cells to apply braking until all trip expected energy demand is fulfilled. The energy consumption of the refrigeration system is assumed to be

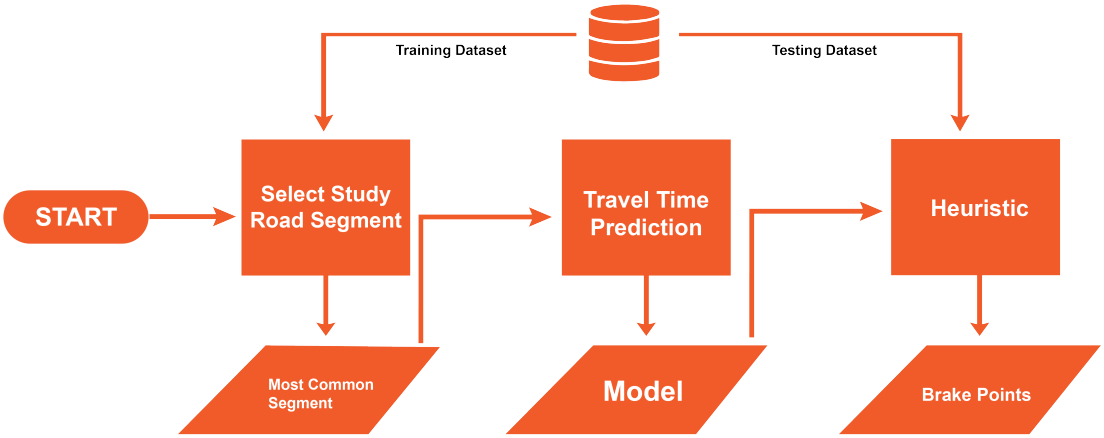
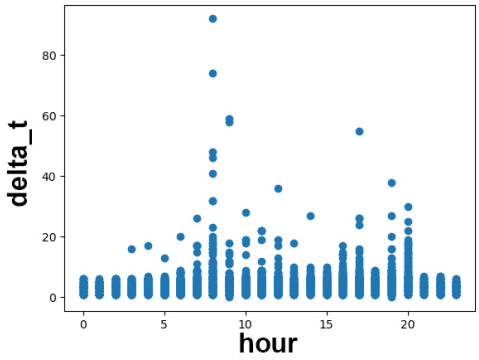


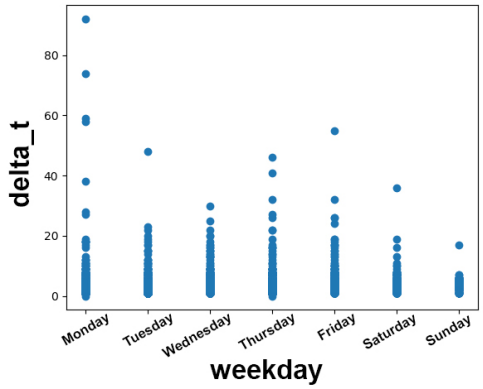
Figure 3.3: Proposed Solution Architecture

constant as well as the energy recovered by the RB system for each second it is applied.

In order to evaluate the heuristic, in an business perspective, the number of seconds the RB is applied, during a trip, will be taken into account, on which less seconds is better.



(a) Scatter plot of the time it takes to travel a chosen cell, by hour of the day.



(b) Scatter plot of the time it takes to travel a chosen cell, by day of the week.

Figure 3.4: Travel time scatter plots by hour of the day and day of the week.

Chapter 4

Data Collection and Pre-Processing

The data collected for this work include operation metrics of the EMS, the RU, the vehicle and their accessories, as stated in the introduction of this document, and comes in the form of time series. This information is transmitted to the AddVolt cloud server via a secure communication channel that is ensured by a SIM card with GSM. However, the data acquired is highly susceptible to noise, inconsistency, inaccuracy and missing data. This may occur mainly due to phenomenons like: lost signal and/or device malfunction. If no pre-processing of the data is made before trying to mine knowledge, the results may be of low-quality or even not useful to improve the system intelligence. In addition to smoothing noisy data and filling missing values we must proceed with data reduction. The available dataset is very large, and, if not reduced, several mining tasks would be very resource-heavy. Since the dataset includes many attributes that are not relevant for this work, one step for data reduction will be to keep only the relevant attributes.

The attributes that will be used are: *registered_at*, *latitude*, *longitude*, *altitude*, *speed*, *speed_can*, *brake_pedal*, *accelerator_pedal*, *fuel_throttle*, *rpm* and *battery_soc*.

4.1 Pre-Processing

4.1.1 Missing Data

Ideally the GPS device sends data to the cloud server at an almost constant rate. But, devices malfunction or lost signal may cause significant time gaps between two sent signals. Also, since the communication is made via a SIM card with GSM when a vehicle is parked in facilities or goes through a tunnel, signal may be temporarily lost.

According to [4] a data point may be considered missing "if the time elapsed since the last transmission falls above two times the median sampling rate of the vehicle in question".

In the scope of this work, missing values may also occur due to some vehicles not collecting all variables. For instance, some vehicles only transmit speed recorded by the GPS, others do not have data related to accelerator and brake pedals and others do not have RPM information. In those cases, the following measures are taken:

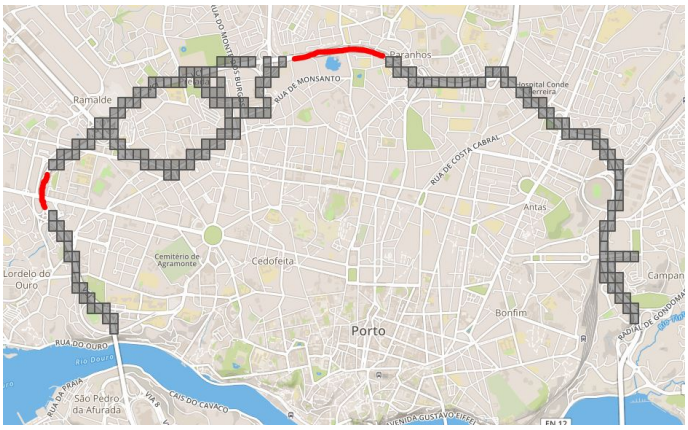


Figure 4.1: Example of a temporary lost signal during a trip. Since, the signal was lost for less than 61 consecutive seconds the trip was not cut.

- Speed: The speed value used for this work is always the recorded by the vehicle itself, if present, if not the GPS recorded speed is used. In cases of temporary loss of the speed value, the last known value is used (forward filling). If a vehicle does not have any kind of speed measure, the data is discarded.

- Pedals: If there is no information about the pressure applied in any of the pedals (accelerator and/or brake) a value of 0 is assumed. For those cases, the acceleration and braking time will be infered by the increase or decrease in speed.

- RPM: Missing values in RPM will not allow us to correctly discretize trips, since there is no information about wether the vehicle is running or stopped in situation where the speed is 0 kilometers per hour(km/h). This is due to situations where the speed is 0 km/h because the trucks is stopped in traffic (congestion, traffic signal, etc.). Missing RPM values are then filled with the last known value for 60 seconds of consecutive missing data, after that it is filled with 0. Data with RPM = 0 will be considered as the truck not being in a trip and thus discarded for the final dataset.

Finally, a timeout of 61 seconds is applied, in which if there is no data at all for 61 consecutive seconds cutting is applied indicating the end of the current trip. If not in a trip, this missing data will not be relevant.

4.1.2 Reliability

It is important when analyzing a dataset to guarantee its logical sense. Similarly to the missing data scenarios, a device malfunction and/or lost signal may send illogical data to the cloud server, causing noise. According to [12] noise can be seen as "a random error or variance in a measured variable". This noisy data must be treated in order to not produce outliers that would greatly affect our final model. Some examples of this type of data that can be present on the dataset are:

- GPS positions with a distance between each other that is much greater than the distance the vehicle could realistically travel in the elapsed time between signals. In this cases the data is simply discarded.
- Speed values greater than 150 km/h are also discarded, since it is not expected for the vehicles, present in our dataset, to travel at higher speeds.
- RPM values greater than 6000 RPM are discarded, since those values are over the maximum operable value of the used diesel engines.

4.1.3 Accuracy

The accuracy of a given GPS signal relates to the discrepancy between the position of the signal and the real position of the analyzed vehicle. Inaccurate GPS coordinates are a frequent problem when analyzing FCD which, if not handled carefully, may jeopardize any further analysis. A first and simple step to check the validity of the recorded GPS coordinates consists in verifying if the difference between the tested latitude and longitude and the previous received coordinates is greater, in degrees, than a certain value. The periodicity with which a new GPS signal is received is 1 second. Having this in mind, and knowing that we do not expect the vehicle speed to be higher than 150 km/h, see 4.1.2, we do not expect a difference greater than 0.01 degrees in coordinates.

Even though this step allows us to eliminate most of invalid GPS coordinates, an extra step should be taken. The GPS signal may be lost for an undetermined amount of time. This may cause consecutive invalid coordinates to be received. If we rely only on the previous described step some invalid GPS coordinates may be missed. In order to avoid this problem, any coordinate point that is outside a defined bounding box - the defined area the vehicle is expected to travel, see Fig. 4.2 - is removed from the dataset. The combination of both the described techniques removed all present outliers from the testing dataset.



Figure 4.2: Rectangular bounding box of where the data is expected to be present.

Chapter 5

Selection of Test Road Segment

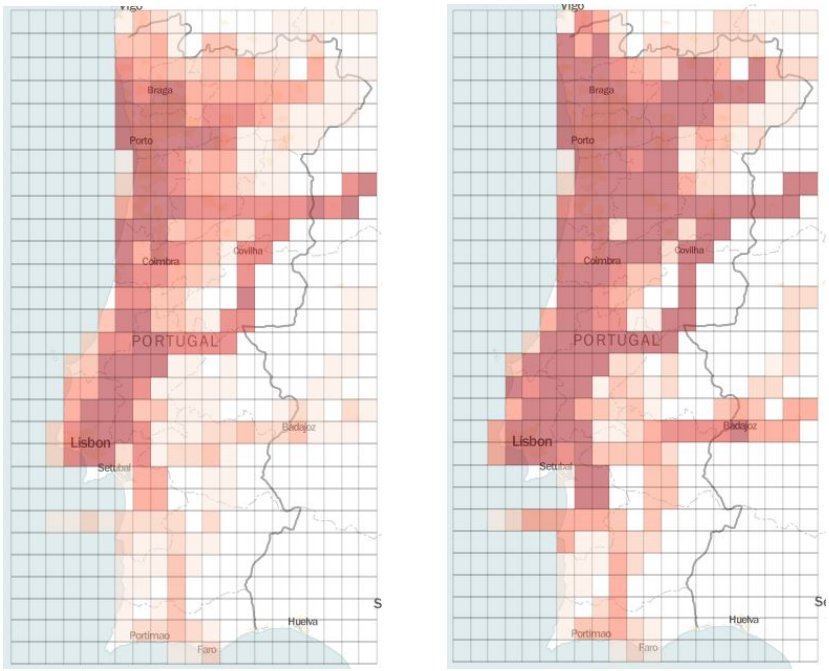
Each road may have different particularities that affect the time it takes to travel it. It would not be viable, in terms of processing and results, to use all available data, regardless of its geographic location, to develop a unique travel time prediction model. Since the number of vehicles equipped with these sensors is very reduced, at the moment, their spatial coverage is also very limited. In consequence, for the development of this thesis we will go with a proof of concept of limited scope. A critical step will then be to find a test road segment that has a relevant abundance of data, in order to develop a model and test it.

This process is done in three main steps, which are: spatial discretization, trip discretization and finding the most common segment.

5.1 Spatial Discretization

According to [27] spatial clustering can be defined as "the process of grouping objects with certain dimensions into groups such that objects within a group present similar characteristics". In order to be able to extract information from spatial coordinates - *latitude* and *longitude* - a spatial clustering is made. This clustering aims to group positions that are within a defined distance. With this is mind, a grid based method is used to divide the space into a finite equal number of cells with a given size. Each data point is then mapped to its corresponding cell. Given a bounding box of minimum and maximum latitude and longitude, the grid is generated by dividing the space horizontally for the longitude, and vertically for the latitude.

It is known that, at the time of the work, most vehicles are present in Portugal, so, most of the data is in Portugal. With this information, only Portugal was contemplated, initially. To choose the area, in Portugal, with most data for further investigation, a grid with cells of 10x10 kilometers was used to find the area with best hourly coverage (number of hours covered for each cell) and best week day coverage (number of days of the week covered), see figure 5.1. The vehicles coverage (number of unique vehicles that travelled each cell, at least once) was also taken into account, see fig. 5.2.Oporto was found to be the area with most relevant data, and thus the area used for the



(a) Number of hours covered for each cell.

(b) Number of days of the week covered for each cell.

Figure 5.1: Data Coverage

remaining of this work. Finally, the chosen area, was further divided into cells of 200x200 meters for the followings steps of the work. All data outside the selected area was discarded.

With these steps processed, the original dataset is greatly trimmed allowing us to proceed to the next step, trip discretization, which is explained in the next section.

5.2 Trip Discretization

After all pre-processing, spatial clustering and mapping is done, the dataset is segmented into trips. The segmentation into trips allow us to infer new features, trip related, and to analyze each trip independently. This step involves domain knowledge to define when the start and the end of a trip should be considered. The vehicles engine rotation per minute (RPM) is a viable feature to detect if a vehicle is running. A RPM value of 0 indicates that the vehicles is not running. In the other hand, it is known, by business knowledge, that the minimum RPM for an engine diesel to run is above 300. An increase of the RPM value, from 0, gets flagged as the start of a new trip. If the RPM drops below 300 while in a trip, a trip end is considered. Although this solution is good enough for most cases, additional conditions were added. If a vehicles does not move - change cells - or no data is received for more that 5 minutes, the trip is stopped, in order to avoid having too much temporal space without any data or consider load/unload and parking maneuvers which are not relevant. See fig. 5.3 for an example of a parking maneuver. Additionally, a trip

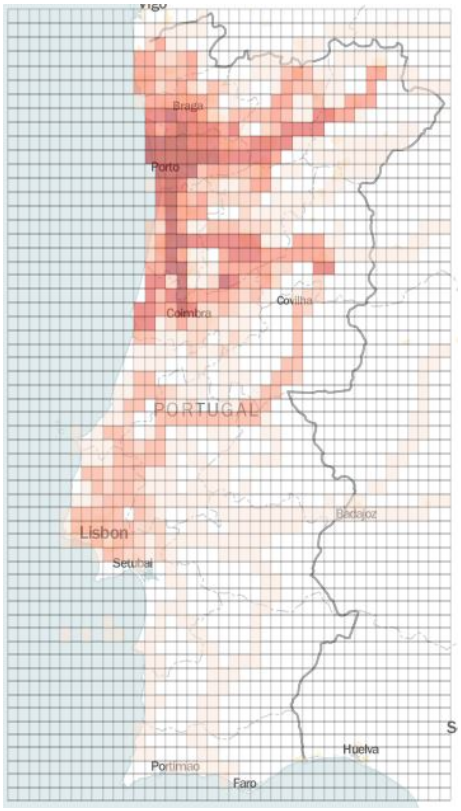


Figure 5.2: Number of unique vehicles travelling, for each cell.

with less than two cells covered is discarded, since they could also be only parking or load/unload maneuvers. As an example, Fig. 5.4 show one of the biggest trip found inside the bounding box.

Since each datapoint already has a cell associated, we can reformat a trip to be an ordered list of cells. The list can be mined for sequential patterns, in order to find the most common sequence of trips.

5.3 Sequential Pattern Mining

Having the dataset grouped by trip the final step for the selection of the test road segment is to find the most common segment. For this purpose a sequential pattern mining algorithm, Prefix Span, is used. This algorithm finds and counts all cells combinations present, on the trips dataset, taking into account the direction. This way, it is possible to take into account both trips that travel all selected cells, as well as only some of the cells. For the purpose of this work a combination of 25 cells was used, which resulted in a total of 1013 trips, with full hour and day coverage.

Figure 5.5 show the most common segment found with prefix span, and the bounding box in which the segment was contained. This segment will be used for the remainder of this work. The selected segment is on the N14 road, direction Maia to Oporto, starting at the entry to N14 by the

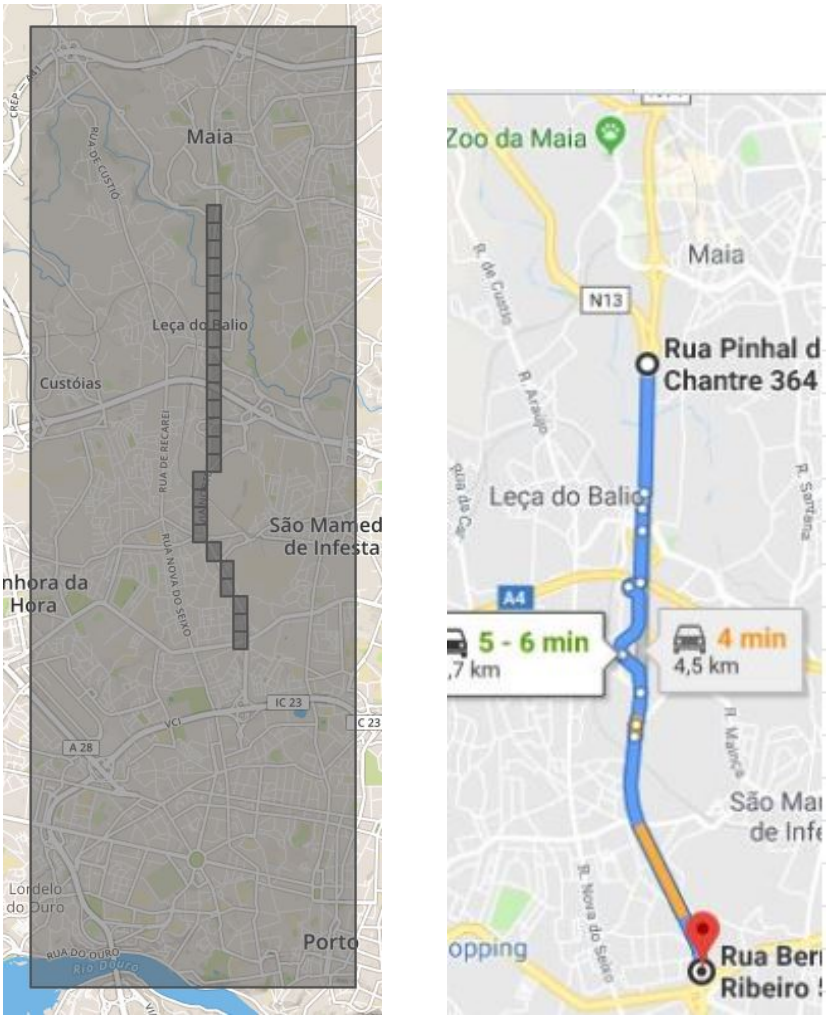


Figure 5.3: Example of a trip finishing with a parking maneuver.

N13 road and finishing at the N12 road intersection. The segment has a total of 4.5 kilometers. It is an highly frequented road, subject to the usual peak hours.



Figure 5.4: Biggest trip found inside the used bounding box.



(a) Bounding box and segment (b) Google Maps segment representation

Figure 5.5: Bounding box and selected segment

Chapter 6

Travel Time Prediction

Having the test road segment selected a travel time prediction model is developed based on that same road segment. This process includes a process of features engineering, model building, hyperparameter tuning and testing.

6.1 Features

The process of feature engineering, of this work, starts with the building of features by data grouping and includes the creation of new features, more business related, using domain knowledge. The features will then be used to predict the target variable *delta_t*, which corresponds to the time the vehicles takes to go through the input cell.

6.1.1 Data Grouping

The travel time prediction will be made on a cell level, making this step fundamental to develop a prediction model. The main goal of this step is to create features related to each cell that may bring useful resources to improve the prediction model. This data grouping is achieved by aggregating series that are inside the same cell of a given trip. With this grouping new features are created. Those features are focused in two main categories:

- Previous Hour: aggregated historic data for the previous hour, independent of the day, for data inside the defined bounding box.
- Previous Cell: aggregated historic data for the last travelled cell, in the trip defined in 5, independent of the day and hour.

For each of these categories the features built are: mean acceleration time, mean brake time and mean speed. For the previous cell case the mean entry speed is also calculated, which is the speed that the vehicle enters a cell.

6.1.2 Features Building Using Domain Knowledge

In order to build relevant features for the business some domain research was made to understand how analyzing and combining different features of the raw data could result in new important features. Since weekends and holidays proved to have some impact on travel time, due to less traffic on the road, those features were included in the dataset, in boolean format indicating wether or not the current day is a weekend day or holiday. Also, the topography of the road may have an impact on the travel time. On a descent the vehicle may be able to travel at a higher speed, whether on a high inclination ascent it may be hard to travel at high speed. So, slope, in degrees, and a boolean value indicating if a cell is a descent or not, were included as features.

Additionally, some existing features were transformed in order to better answer to requirements of this work. Hour, weekday and month features were converted to cosine and sine functions, in order to take into account the cyclic nature of those features (e.g. 12AM should be close to 1AM, which would not happen otherwise). The categorical feature, *cell _id*, was converted to a binary representation, using a One-Hot Encoding.

On a more "cell-specific" level, the distance to some main cities was used as features. In this specific case the center of Maia and Oporto were used, since they are the closest main cities of the test road segment. For different segments this features may have to be adjusted.

Finally, since weather greatly affects driving style and traffic, weather information, consisting of: temperature, visibility, humidity, wind speed, cloud cover and precipitation intensity and probability, was added to the features. See table 6.2 for a list of used weather features and their used units. This information was obtained from an API. The API used for the scope of this work was *DarkSky*, since the number of free requests was enough for the used dataset. The requested weather information was always based on the whole day of the data, and for the center of Oporto - one of the closest main cities from our testing road segment - in order to reduce the number of requests.

6.2 Prediction Model

Using the features built in the previous step some prediction models were built. Since the target variable has a continuous value, different regression algorithms were used. A linear regression model was used as a baseline model. Then, a SGD regressor and a random forest model were developed and tuned for improved performance.

The evaluation of each model was made using four main metrics: Mean Average Percentage Error (MAPE), Mean Average Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).

Hyperparameter tuning was conducted using grid search. The meta-optimization criteria used was the minimization of MSE. The generalization error of each training configuration was evaluated using k-fold cross validation.

Feature Name	**Type**	**Description**
cell_id	Categorical	ID of the cell
mean_acc_time_prev_hour	Float	Mean time the vehicle accelerated in the previous hour. (Historical)
mean_acc_time_prev_cell	Float	Mean time the vehicle accelerated in the previous cell of the trip. (Historical)
mean_brake_time_prev_hour	Float	Mean time the vehicle braked in the previous hour. (Historical)
mean_brake_time_prev_cell	Float	Mean time the vehicle brakedin the previous cell of the trip. (Historical)
entry_altitude	Float	Altitude at the beginning of the cell.
exit_altitude	Float	Altitude at the end of the cell.
mean_entry_speed_prev_hour	Float	Mean speed the vehicle enters the cell at the previous hour. (Historical)
mean_exit_speed_prev_hour	Float	Mean speed the vehicle leaves the cell at the previous hour. (Historical)
hour	Integer	Current hour.
weekday	Integer	Current weekday.
month	Integer	Current month.
is_weekend	Boolean	If it is Saturday or Sunday.
is_holiday	Boolean	If it is a Holiday.
is_descent	Boolean	If the current cell, at the current direction, is a descent.
slope	Float	Inclination, in percentage, of the cell.
weather_icon	Categorical	String describing the weather.
distance_to_maia	Float	Distance from the center of the current cell to the center of Maia.
distance_to_oporto	Float	Distance from the center of the current cell to the center of Oporto.
distance_to_end	Float	Distance from the center of the current cell to the center of the last cell of the trip.
distance_to_start	Float	Distance from the center of the current cell to the center of the first cell of the trip.

Table 6.1: Table of features used.

Feature	Description
cloud_cover	Percentage of sky occluded by clouds (0 to 1).
humidity	Relative humidity (0 to 1).
icon	Categorical summary of weather info (cloudy, wind, snow, rain, etc).
precip_intensity	Milimeters per hour.
precip_probability	Probability of precipitation occuring (0 to 1).
precip_type	Type of precipitation: "rain", "snow", "sleet".
temperature	The air temperature in Celsius degrees.
visibility	Average visibility in kilometers.
wind_speed	Wind speed in kilometers per hour.

Table 6.2: Weather features used.

A simple prediction model was developed, as first baseline. This model consists in predicting the travel time by returning the historic average travel time for a given weekday and hour. The baseline is used to compare the improvements each more robust model can deliver.

6.2.1 Model Evaluation

Different regressions will be used, in order to find the algorithm that better fits the problem. The results obtained with each algorithm have to be described in an understandable way to find the improvements from one another and also choose the best fit to the problem. The error obtained from the prediction model results to the actual real value will be calculated using the following formula:

$\varepsilon_i = T_pi - T_Ai$ where ε_i is the prediction error, T_pi is the predicted travel time and T_Ai is the actual travel time.

For a clear understanding of the accuracy and precision of the model two metrics will be used: MAE for accuracy and MAPE for precision [11]. Additionally, MSE, to assess the quality of the predictor and RMSE, a measure more sensible to outliers, will be used.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\varepsilon_i|$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\varepsilon_i}{T_Ai}\right|$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n}^{1-n}(Y_i - \hat{Y}_i)^2}$$

6.2.2 Linear Regression

Different configurations were tested for the linear regression. As baseline, a linear regression without data scaling, no polinomial features, using all features and no regularization was tested. Finally, a L1 and a L2 regularization were tested. The best results were obtained scaling the features using MinMax algorithm, transforming the features to polinomial features of 2nd degree and with a L1 regularization, Lasso. From now on, this configuration will be refered as "best configuration".

6.2.3 Stochastic Gradient Descent Regressor

A stochastic gradient descent (SGD) regressor can be seen as a stochastic approximation of a gradient descent optimization. This model estimates the gradient of the loss each sample at a time, with a decreasing learning rate, in order to minimize the loss. One of the main advantages of this model is being adaptive to new data, which makes it a solution for future work with online learning.

A random search, with 60 iterations, was applied for the hyperparameter tuning. The configuration used, which returned the best results, was using an adaptive learning rate, an epsilon-insensitive loss function (with epsilon = 0.1) and a maximum of 1000 iterations.

6.2.4 Random Forest

Random forest is an ensemble learning method that fits a giver number of decision trees on differente sub-samples of the dataset, using average to improve its predictive accuracy and prevent over-fitting. The implementation used was made in Python, present in the package Sci-Kit Learn. Using a randomized search the hyperparameters of the random forest algorithm were tuned. The best results were obtained with 715 trees, a maximum depth of 110 and considering the maximum features for the best split to be: sqrt(n_features).

Although some improvements on accuracy could be obtained by increasing the number of trees, the maximum value tested was 800 trees, due to having a large set of features and some hardware limitation which would result in too much calculation time and memory problems.

6.2.5 Summary and Results

Finally, in this section we present the results obtained with the best configurations of each of the algorithms tested. The best final results were obtained with the SGD and Random Forest algorithms. SGD is the one that presented the best accuracy and MAE but more sensible to outliers than Random Forest. Since for the scope of this work the goal was to maximize accuracy, but also take into account possible outliers, due to the stochastic nature of traffic flow, the choosen final algorithm was Random Forest. See table 6.3 for an overview of the results of each model.

Algorithm	**MAPE**	**MAE**	**MSE**	**RMSE**
Baseline	18.14%	1.908	26.875	5.184
Linear Regression (Baseline)	26.01%	1.789	30.117	5.488
Linear Regression (Best Configuration)	16.60%	1.966	33.166	5.759
SGD	10.35%	1.481	31.808	5.64
Random Forest	14.40%	1.691	27.577	5.251

Table 6.3: Results obtained with each different algorithm.

Chapter 7

Decision Support Heuristic

In this chapter, the results from chapter 6 are used to support a decision heuristic. This heuristic aims to improve the efficiency of the RB system by minimizing its usage while also minimizing the need to use diesel, while also ensuring the operation of the RU. Additionally, all RB operation necessary to guarantee a full electric operation is, if possible, aimed to be applied in optimal braking points.The choosing of the optimal braking points will be explained in this chapter.

Finally, the final decision and its application on the truck are explained in 7.2, as well as the obtained results 7.3

7.1 Ranking Cells by Energy Potential

Since for all the previous work a grid system is used to predict travel time, the same grid is used for the decision heuristic. So, a decision to apply RB or not, is at a cell level. Meaning that if a decision is made to apply RB on a given cell, it will be applied for the entire cell.

For a given trip, spatially discretized by cells, each cell is ranked by its energetic potential. Given that calculating the actual energetic potential (how much energy is recovered) of each cell versus its actual energetic cost (how much more fuel is consumed by applying RB) is unrealistic and outside the scope of this work, a simplification is made. It is known, a priori, that the vehicle must brake if travelling a descent to maintain its speed, needing more brake power the greater the slope. In the other hand, when going uphill, the greater the slope, the more acceleration power is needed, thus increasing fuel consumption.

Based on the previous assumptions, cells that go uphill are heavily penalized, with more penalization the greater the slope, and used only in last resort. Cells that go downhill are highly ranked, having more importance the greater their slope.

The decision support heuristic gives, then, more importance to highly ranked cells, and uses poorly ranked cells only if really needed to complete the trip with a full electric operation.

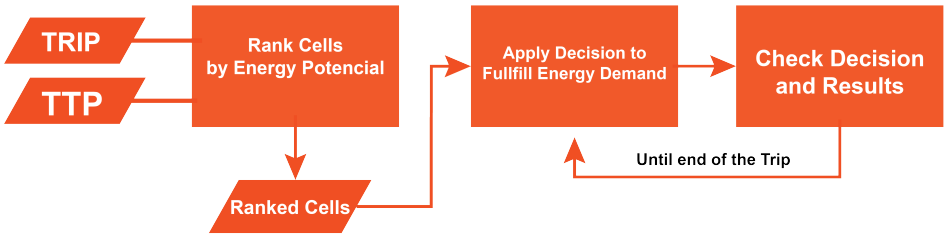


Figure 7.1: Decision flow chart.

7.2 Final Decision

The decision processed and sent to each vehicle is made taking into account a trip and its division by cell. Having the predicted travel time for the trip, and thus for each cell of the trip, the heuristic can calculate approximately the expected SOC for each point of the trip. The RU electric consumption is assumed to be constant, for the scope of this work, but can easily be defined for each RU system.

If it is predicted that there is not enough battery to complete the trip in full electric mode, the decision support heuristic takes the decision to apply RB in the first next highest ranked cell. A recalculation of the SOC for each remaining point of the trip is then made, and if there is still not enough electric energy to fulfill the trip, the process is repeated.

There are various security measures in place, which may disable RB in cases of overheat or emergency brakes, which this intelligence layer does not override. This cannot guarantee that a decision made is actually applied, being fundamental the constant recalculation of the SOC of the batteries. See Fig.7.1 for the decision flow.

7.3 Summary and Results

In order to evaluate the developed heuristic, a cost metric was created. The cost is defined by a cost of usage per seconds. This is calculated by considering the RB system cost, the expected lifetime (in seconds) and the total cost of maintenances per seconds.

The decision support heuristic is then compared to a random heuristic. The random heuristic consists in applying braking in random cells of the trip. The evaluation was made using 10 different trips, comparing the cost of the developed decision support heuristic results with the average cost of 10 runs of a random decision. See table 7.1 for results.

The developed work showed promising results, although some additional experiences should be made with more exceptional traffic situations and different road segments. In 5 tested trips, an average improvement of 29.3% was obtained, when compared to a random heuristic.

Heuristic	**Random**	**Improvement (%)**	**Hour**	**Weekday**
0.046	0.060	23.33	14	Wednesday
0.032	0.049	34.69	5	Wednesday
0.035	0.059	40.68	10	Tuesday
0.035	0.051	31.37	5	Monday
0.046	0.055	16.36	7	Thursday

Table 7.1: Results obtained with the decision support heuristic.

Chapter 8

Conclusions

The developed work uses data that was alread available to add a layer of intelligence to the EMS to improve the RB usage. Also, the developed travel time prediction model can be used for other ends like report to transportation companies how it will take to deliver the goods and how will be the energy levels of each vehicle at each instant of each trip.

The decision taken using the developed solution can be adapted to different energy profiles. Instead of only ensure enough energy to complete a trip it can be optimized to store the maximum energy on each trip. In this case, the remaining energy can be fed to other vehicle's RUs or give back energy to the grid. Not only can this solution help reduce energy consumption of transportation of refrigerated goods as it can transform this kind of vehicles into energy production units. This would bring even more value to the AddVolt's solution as well as the way transportation companies operate.

The developed framework has its focus on intelligently apply RB, automatically, on strategic points of each trip, in order to guarantee enough electrical energy to feed the RU in the whole trip, without having to use diesel. All of this work is based on historical data, with an introduction to online adjustments.

The developed work was made with maintenance and scalability, in such a way, that every part of it can be used and changed individually without compromising the others. Every step of this work, mentioned through the document, is self-contained with well-defined inputs and outputs making them easily adaptable to differents projects and easy to adapt to future improvements.

8.1 Future Work

The work proved its concept and showed improvements in comparison to a random heuristic or no decision-making at all. But, it also revealed a lot of potential present in the available data which combined with future improvements could yield great enhancements to the intelligence layer of the EMS. In this section some future work potential is suggested.

Feature	Description
speed_limit	Speed limit of the cell.
has_traffic_light	If a cell has any traffic light.
n_traffic_lights	Number of traffic lights in the cell.
type_of_road	The type of road of the cell.
has_radar	If the cell has any known speed radar
has_poi	If the cell has any Point of Interest
n_poi	Number of Points of Interest in the cell.

Table 8.1: Example of additional features, related to road characteristics.

8.1.1 Additional Road Characteristics

Many external features, like road characteristics, may influence the braking profile for a given trip. So, adding those informations such as: speed limits, traffic lights, type of road, presence of speed radars, etc for each trip, to the dataset, could greatly improve the energetic potential of each cell of the trip, thus being able to better rank the cells. See table 8.1 for more details.

8.1.2 Spatial Discretization - A Different Approach

The approach for the spatial discretization of a trip, by a grid system, was not a problem for the development of this work, but may prove to not be easily scalable for bigger maps and forces the creation of a new grid for each trip.

This topic opens, then, a great room for improvement. An encoding based directly on the coordinates (latitude and longitude) may greatly improve precision, searching and grid matching time. This approach is presented in a project called "what3words" where the entire world is divided in equally sized squares (cells) denominated with an unique name composed by three words. See Fig.8.1 for more details.

8.2 Online Adjustment

Traffic flow has a stochastic nature, which has to be taken into account when making a decision. A new travel time prediction is given every time the vehicle enters a new cell. Nevertheless, if this prediction is consistently wrong due to some special occurence on the road (like a road accident, road closure, etc.), our heuristic should have the ability to adapt its decision taking that into account. The time predictions made for a given trip should be adjusted, as well as the expected state-of-charge (SOC) of the batteries, according to new data related to the current trip, that is collected by the AddVolt's servers. With the adjusted time prediction, a new predicted SOC for each remaining cell of the trip could calculated, potentially updating the decisions made by the decision support heuristic.

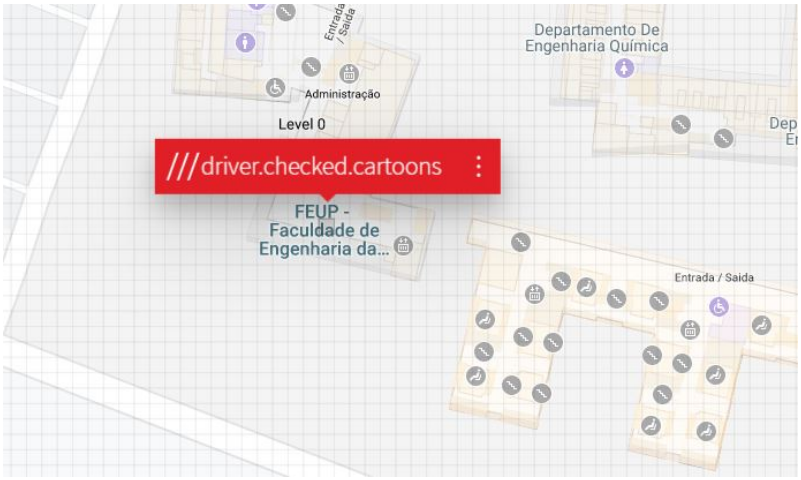


Figure 8.1: What3words usage example.

References

[1] Ethem Alpaydin. *Introduction to Machine Learning Second Edition*. MIT Press Ltd, 2nd editio edition, 2010.

[2] Christopher R. Bennett, Hernán De Solminihac, and Alondra Chamorro. Data Collection Technologies for Road Management. *Transport Notes Series; No. TRN 30. World Bank, Washington, DC. © World Bank*, 2006.

[3] F. Billiard. Refrigerated transport: what's new? *International Journal of Refrigeration*, 25(5):501–503, 2002.

[4] V Cerqueira, L Moreira-Matias, J Khiari, and H van Lint. On Evaluating Floating Car Data Quality for Knowledge Discovery. *IEEE Transactions on Intelligent Transportation Systems*, 19(11):3749–3760, 2018.

[5] Peter Clarke, Tariq Muneer, and Kevin Cullinane. Cutting vehicle emissions with regenerative braking. *Transportation Research Part D: Transport and Environment*, 15(3):160–167, 2010.

[6] Herve DEBAR, Monique BECKER, and Didier SIBONI. A Neural Network Component for an Intrusion Detection System. In *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, SP '92, pages 240—-, Washington, DC, USA, 1992. IEEE Computer Society.

[7] David Enke and Suraphan Thawornwong. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29(4):927–940, 2005.

[8] Philippe Esling and Carlos Agon. Time-series Data Mining. *ACM Comput. Surv.*, 45(1):12:1—-12:34, 2012.

[9] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[10] Chen Guojun, Teng Jing, Zhang Shuyang, and Yang Xiaoguang. Tendency-Based Approach for Link Travel Time Estimation. *Journal of Transportation Engineering*, 139(4):350–357, apr 2013.

[11] Chen Guojun, Yang Xiaoguang, An Jian, and Zhang Dong. Bus-Arrival-Time Prediction Models: Link-Based and Section-Based. *Journal of Transportation Engineering*, 138(1):60–66, jan 2012.

[12] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining Concepts and Techniques*. Elsevier, 3rd editio edition, 2012.

[13] Fiona Harvey. Four of world's biggest cities to ban diesel cars from their centres, 2016.

[14] Chien Steven I-Jy and Kuchipudi Chandra Mouly. Dynamic Travel Time Prediction with Real-Time and Historic Data. *Journal of Transportation Engineering*, 129(6):608–616, nov 2003.

[15] Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. PrefixSpan,: mining sequential patterns efficiently by prefix-projected pattern growth. pages 215–224, 2005.

[16] J.W.C. Van Lint. *Reliable Travel Time Prediction for Freeways*. Technische Universiteit Delft - Netherlands, 2004.

[17] Xiangjie Kong, Zhenzhen Xu, Guojiang Shen, Jinzhong Wang, Qiuyuan Yang, and Benshi Zhang. Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems*, 61:97–107, 2016.

[18] Guillaume Leduc. *Road Traffic Data: Collection Methods and Applications*. jan 2008.

[19] Chi-Sen Li and Mu-Chen Chen. A data mining based approach for travel time prediction in freeway with non-recurrent congestion. *Neurocomputing*, 133:74–83, 2014.

[20] Hong-En LIN and Rocco Zito. *A review of travel-time prediction in transport and logistics*, volume 5. jan 2005.

[21] Gordon S. Linoff and Michael J. A. Berry. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. Wiley Publishing, Inc., 3rd ed. edition, 2011.

[22] Gys Meiring, Hermanus Myburgh, Gys Albertus Marthinus Meiring, and Hermanus Carel Myburgh. A Review of Intelligent Driving Style Analysis Systems and Related Artificial Intelligence Algorithms. *Sensors*, 15(12):30653–30682, dec 2015.

[23] Ma Minghui, Shidong Liang, Hui Guo, and Jufen Yang. *Short-term traffic flow prediction using a self-adaptive two-dimensional forecasting method*, volume 9. aug 2017.

[24] Siti Aishah Mohd Selamat, Simant Prakoonwit, Reza Sahandi, Wajid Khan, and Manoharan Ramachandran. Big data analytics—A review of data-mining models for small and medium enterprises in the transportation sector. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(3):1–14, 2018.

[25] H M Moonam, X Qin, and J Zhang. Utilizing data mining techniques to predict expected freeway travel time from experienced travel time. *Mathematics and Computers in Simulation*, 155:154–167, 2019.

[26] L Moreira-Matias, J Gama, M Ferreira, J Mendes-Moreira, and L Damas. Predicting Taxi–Passenger Demand Using Streaming Data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, 2013.

[27] Mr.Subu Surendran and C V Neethu. Review of Spatial Clustering Methods. *International Journal of Information Technology Infrastructure*, 2(3), 2013.

[28] Wasan Pattara-atikom, P Pongpaibool, and Suttipong Thajchayapong. *Estimating Road Traffic Congestion using Vehicle Velocity*. jul 2006.

[29] Marek Potkány and Lucia Krajčírová. Cost Reporting of the Transport Company and its Use in Decision-making. *Procedia Engineering*, 192:731–736, 2017.

[30] Filipe Rodrigues, Ioulia Markou, and Francisco C Pereira. Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. *Information Fusion*, 49:120–129, 2019.

[31] Juan J. Rodriguez and Ludmila I. Kuncheva. Time Series Classification : Decision Forests and SVM on Interval and DTW Features. In *Proceedings of the Workshop on Time Series Classification, 13th International Conference on Knowledge Discovery and Data Mining*, 2007.

[32] S. A. Tassou, G. De-Lille, and Y. T. Ge. Food transport refrigeration - Approaches to reduce energy consumption and environmental impacts of road transport. *Applied Thermal Engineering*, 29(8-9):1467–1477, 2009.

[33] Siang Fui Tie and Chee Wei Tan. A review of energy sources and energy management system in electric vehicles. *Renewable and Sustainable Energy Reviews*, 20:82–102, 2013.

[34] Shawn M Turner, William L Eisele, Robert J Benz, and J Douglas. *Travel time data collection handbook. Technical Report*. Number March. 1998.

[35] Muhammad Imdad Ullah. Time Series Analysis and Forecasting, 2013.

[36] J W C van Lint, S P Hoogendoorn, and H J van Zuylen. Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5):347–369, 2005.

[37] Senzhang Wang, Xiaoming Zhang, Jianping Cao, Lifang He, Leon Stenneth, Philip S. Yu, Zhoujun Li, and Zhiqiu Huang. Computing Urban Traffic Congestions by Incorporating Sparse GPS Probe Data and Social Media Data. *ACM Transactions on Information Systems*, 35(4):1–30, 2017.

[38] Z Wang, K Fu, and J Ye. Learning to estimate the travel time. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 858–866. Association for Computing Machinery, 2018.

[39] Zheng Weizhong, Lee Der-Horng, and Shi Qixin. Short-Term Freeway Traffic Flow Prediction: Bayesian Combined Neural Network Approach. *Journal of Transportation Engineering*, 132(2):114–121, feb 2006.

[40] Z Wu, C Li, Y Wu, F Xiao, L Zhu, and J Shen. Travel time estimation using spatio-temporal index based on cassandra. In Zlatanova S Dragicevic S. Sithole G., editor, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 4, pages 235–242. Copernicus GmbH, 2018.

[41] Jungkeun Yoon, Brian Noble, and Mingyan Liu. Surface Street Traffic Estimation. In *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, MobiSys '07, pages 220–232, New York, NY, USA, 2007. ACM.

[42] Xiaoyan Zhang and John A Rice. Short-term travel time prediction. *Transportation Research Part C: Emerging Technologies*, 11(3):187–210, 2003.