UNIVERSITY OF TARTU

Institute of Computer Science

Computer Science Curriculum

Sander-Sebastian Värv

Travel Time Prediction Based on Raw GPS Data

Master's Thesis (30 ECTS)

Supervisors: Amnir Hadachi, PhD

Artjom Lind, MSc

Tartu 2019

Travel Time Prediction Based on Raw GPS Data

Abstract:

With the ever growing pace of our everyday lives, time planning has gained a lot of importance. One of the key factors for time planning is to estimate the duration of moving from one place to another. Therefore, travel time prediction has become essential part of any logistics based business. This thesis is conducted in collaboration with Bolt, which is one of the leading ride hailing companies. This thesis is describing route based travel time prediction algorithm based on raw GPS data. The goal is to analyze each of the preprocessing steps and to develop a coherent method to predict arrival time based on GPS input data supplied by Bolt. Furthermore, route based method described in this thesis is validated by comparing it to two well-known and established methods.

Keywords:

Location-based services, estimated time of arrival, travel time prediction, geospatial information, Open-Street-Map based solution

CERCS: P170

Sõiduaja ennustamine töötlemata GPS andmete põhjal

Lühikokkuvõte:

Aja planeerimine on muutunud aina olulisemaks üha kiireneva elutempoga ühiskonnas. Üheks oluliseks komponendiks aja planeerimisel on võimalikult täpselt hinnata, kui palju aega kulub ühest kohast teise liikumiseks. Käesolev magistritöö on valminud koostöös Boltiga, mis on üks suurimaid sõidujagamisteenust pakkuvaid ettevõtteid. Sõiduaja ennustamine tooreste GPS andmete põhjal nõuab suures koguses andmete eeltöötlemist, kasutades seejuures väliseid andmekogusid, et siduda tooreid GPS andmeid ümbritseva keskkonnaga. Käesolevas töös käsitletakse kõiki vajaminevaid eeltöötlemise samme, millest moodustub terviklik meetod sõiduaja ennustamiseks töötlemata GPS andmete põhjal. Meetodi efektiivsuse valideerimiseks on seda võrreldud kahe laialdaselt kasutuses oleva meetodiga.

Võtmesõnad:

Asukohapõhised teenused, sõiduaja ennustamine, aegridade analüüs, Open-Street-Maps

CERCS: P170

Table of Contents

Acknowledgement ... 5

Abbreviations and Acronyms ... 6

Introduction ... 7

1.1 Objectives and limitations ... 7

1.2 Contribution ... 8

1.3 Roadmap ... 8

2. State-of-the-art solutions ... 10

2.1 Introduction ... 10

2.2 Travel Time Estimation ... 12

2.3 Travel Time Prediction or Estimated Time of Arrival (ETA) ... 13

3. Methodology Adopted ... 17

3.1 Introduction ... 17

3.2 Data used ... 18

3.3 Pre-Processing ... 19

3.4 Map Matching ... 19

3.5 Route Reconstruction and OSRM ... 22

3.6 Traffic Status Estimation ... 23

3.7 Gathering Additional Data ... 28

3.8 Building The Model and Predicting the Travel Time ... 29

4. Results ... 31

4.1 Experiment Setup ... 31

4.2 Results and Feature Performance ... 33

4.3 Comparison with Existing Methods ... 35

4.4 Comparison to Open Source Routing Machine (OSRM) ... 38

4.5 Comparison to major global maps provider service ... 41

5. Discussion 45

5.1 Future perspectives 47

6. Summary 48

7. References 49

Acknowledgement

The author would like to thank his supervisor Amnir Hadachi and all other ITS lab members. Special thanks to Artjom Lind and Shan Wu who provided lots of technical help and guided the process.

In addition, the author would like to thank Bolt for supplying the data and supporting the whole process. Especially would like to thank André Karpištšenko, Ando Saabas and Asko Tamm from the Bolt team for providing excellent feedback, support and for solving technical difficulties.

Abbreviations and Acronyms

ETA – Estimated Time of Arrival

GBR – Gradient Boosted Regressor

GPS – Global Positioning Service

ITS – Intelligent Transportation Systems

LBS – Location-Based Services

LSTM – Long Short Term Memory neural network

MAE – Mean Absolute Error

MAPE – Mean Average Percentage Error

MGMPS – Major Global Maps Provider Service

MLD – Multi Level Dijksta

MLP – Multi Layer Perceptron

MSE – Mean Squared Error

OSM – Open Street Map

OSRM – Open Source Routing Engine

WDR – Wide-Deep-Recurrent model

Introduction

Time planning has become increasingly important part of our everyday lives to fit all the necessary operations into a short period of time. To make the planning more efficient and to make it reflect closer to the reality, we want to estimate the time consumption of each task. One of the tasks that we carry out many times every day is to travel from source to destination. The time consumption of this task can sometimes be very difficult to estimate. Therefore, travel time prediction or estimated time of arrival (ETA) is one of the most well known and most used location-based services.

The travel time prediction is particularly complex task because of all the different variables that influence the outcome. One of the most important variables is traffic density, which is related to the particular location, roadworks, time, weather and is influenced even by the events happening nearby. All of those variables contribute heavily to the route selection and therefore the travel time prediction.

Estimating travel time and time of arrival has become crucial part for any logistics-based business and therefore predictive systems and algorithms working with geospatial data have become increasingly more popular topic in the industry. Some of the ride sharing platforms have already integrated predictive systems to the extent where the system is able to predict the location of the next order and places ghost order in the area which is linked to the actual location as soon as the client confirms the order. This method helps to reduce arrival time and therefore increases customer satisfaction trough user experience. This has resulted the increase in interest of location-based services (LBS), route panning, multimodal journey planning and shared ride, which are the core services provided by all ride hailing companies such as Uber, Bolt, Lyft, etc.

1.1 Objectives and limitations

The goal of this paper is to introduce all the pre-processing stages, which have to be carried out in order to calculate ETA with considerable accuracy. In addition, this paper will give a short overview of the model, which we used to validate the outcome of the pre-processing steps to see how those stages contributed to the final result. The main emphasis will be on estimating traffic status based on the raw Global Positioning Service (GPS) data, which is one of the key factors of providing the accurate prediction for the arrival time.

Our proposed method will be efficient on the same transportation type as the original training data and it will provide travel time estimates only in the areas, which are covered in the training data. In addition, this method is efficient only in relatively similar environments as the original training data because it is static and does not adapt to changes in road network or vehicle behavior. The proposed method requires certain amount of sample resolution which should be at least 1 Hz. However, it can be adapted to sparser data, which is described more detailed in the section 3.5. Furthermore, this method is sensitive to the quality of external data covering speed limits and section lengths as well as input data quality containing average speed of the GPS device.

1.2 Contribution

This thesis has been conducted within collaboration project between the University of Tartu and Bolt (former Taxify) in order to enhance GPS data quality and develop predictive methods. Author of this thesis has taken part of this collaboration project as a member of Intelligent Transportation Systems lab (ITS). Many parts in this thesis are developed in the ITS lab and therefore made in collaboration with other team members, who were part of the Bolt collaboration project. Map matching algorithm together with Open Street Map (OSM) data extractor tool in this thesis is developed in the ITS lab and modified by the author in order to fit to this problem requirement. All other parts are developed solely by the author with some helpful assistance from the members of ITS lab.

1.3 Roadmap

In this thesis, the main focus is to address the problem of predicting the arrival time using information about traffic density. The second section gives an overview of state-of-the-art solutions and broader background of the topic. The third section describes characteristics of the used dataset and covers thoroughly all needed steps to process the raw GPS data into the form which can be used for accurate travel time estimation.

The third section comprises of eight sub-sections where the first sub-section focuses on the broader view of the workflow. The second sub-section describes characteristics of used dataset. Third sub-section focuses on pre-processing, introducing the overall problems of the raw GPS data and how to overcome some of the simple issues like calculating the distance between the samples. The fourth sub-section describes map matching stage and fifth sub-section focuses on route reconstruction and tools used for this purpose. Sixth sub-section is about traffic status estimation, seventh about gathering additional data and coupling with existing one. Last sub-section covers building the model and predicting the travel time.

The fourth section of this thesis gives an overview of the results and experiment setup used for the evaluation. The first sub-section is about experiment setup, second about our proposed method performance and feature importance. Third to fifth sub-sections are about comparison of our proposed method with two other well-known external methods.

Fifth section covers discussion and future perspectives. Lastly, the thesis ends with summary covered in the sixth section.

2. State-of-the-art solutions

Travel time estimation has gained popularity among scientist to develop new solutions helping the industry to provide better products and features. This section will discuss how the input information is obtained and what are the strengths and weaknesses for different types of travel time estimation algorithms. The first sub-section will cover collection of GPS data and specialties for processing such data. Second part discusses travel time estimation in general and third part introduces two alternative approaches which solve the same problem but in very different manner.

2.1 Introduction

The most used data for travel time estimation is collected using satellite navigation system such as Global Positioning System (GPS). This navigation system is owned by the United States government and operated by the United States Air Force. It provides geolocation and time information to a GPS receiver anywhere on or near the Earth. The GPS project was launched by the U.S. Department of Defense in 1973 for the use by the U.S. military. It became fully operational in 1995, while using only 24 satellites [1].

Nowadays, GPS technologies has gained a wide usage and become part of people's everyday life. It is an extremely versatile system and can be found in almost any sector with its applications of positioning, navigation, tracking, mapping and timing. Analyzing GPS data enables optimizing wide variety of different applications. In addition to the GPS system numerous countries have built or are developing their own global satellite navigation systems. For example Russia has developed GLONASS, China has launched BeiDou-2 and European Space Agency is developing Galileo [2], [3].

However, the quality of GPS data is not always consistent, as there are many variables which influence the GPS signal quality and accuracy. For example, GPS signal accuracy is considerably lower close to the high buildings or mountains (Figure 1). This is called canyon effect, which induces considerable amount of noise into the data and can occur even in Tallinn, although there are very few high-rise buildings and even fewer mountains nearby [4]. Canyon effect is very important factor in large cities where the high-rise buildings and other geospatial landmarks might interfere and reflect the signal.

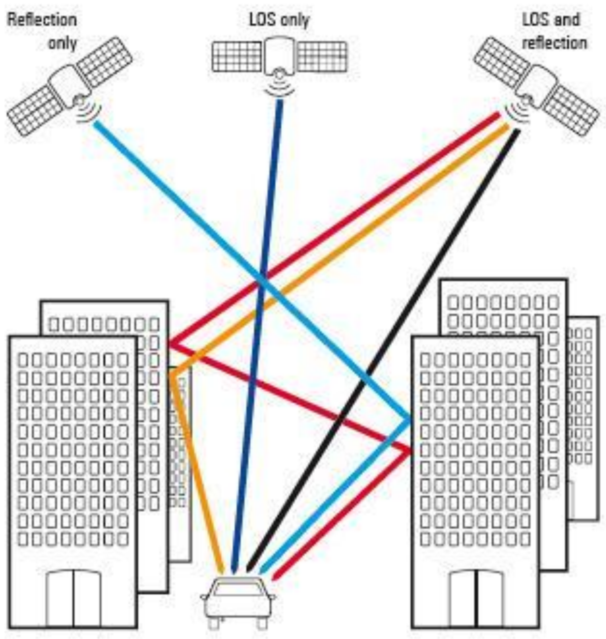

Figure 1. Urban canyon effect, which results in loss of GPS data quality, as the signal is reflecting from the building causing noise and interference [5].

All aforementioned reasons eventually pile up and result a very noisy and inconsistent data. For example, noisy GPS trace of a single vehicle has been captured in Tallinn on the crossing of Laikmaa and Gonsiori streets (Figure 2). The area is covered with nearly uniform density of samples where it is almost impossible to determine the actual position of the device without any contextual information. In this case, the amount of noise is caused by traffic congestion where the device has stayed in the same region for long time and surrounding buildings have added additional noise by reflecting the signal. This effect can be reduced by coupling signal from many satellites at the same time to filter out the noise. This is usually done on the hardware level of the GPS receiver. This situation cannot be avoided completely because buildings and other objects reduce the connectivity and, in many cases, only minimal amount of satellites are visible for the receiver. Such complex problems on the other hand is a very good basis for researchers to solve the problem and develop even better solutions for the problem. The next sub-sections will define the problem scope and cover various different approaches how to tackle those problems and how to calculate ETA with considerable accuracy.

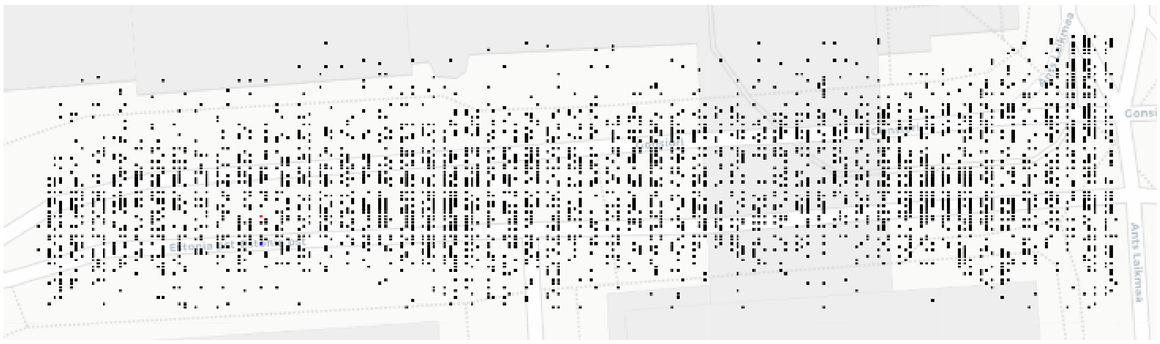

Figure 2. GPS samples are plotted as black dots on the underlying segment covering A. Laikmaa and Gonsiori streets in Tallinn.

2.2 Travel Time Estimation

Travel time estimation is considered as the travel time between the source and destination locations (Figure 3). Accurate travel time estimation has become important in different applications. For example, it is commonly used in public transportation systems for timetable construction and nowadays often integrated into a passenger information system as part of the core functionality of intelligent transportation systems [6].

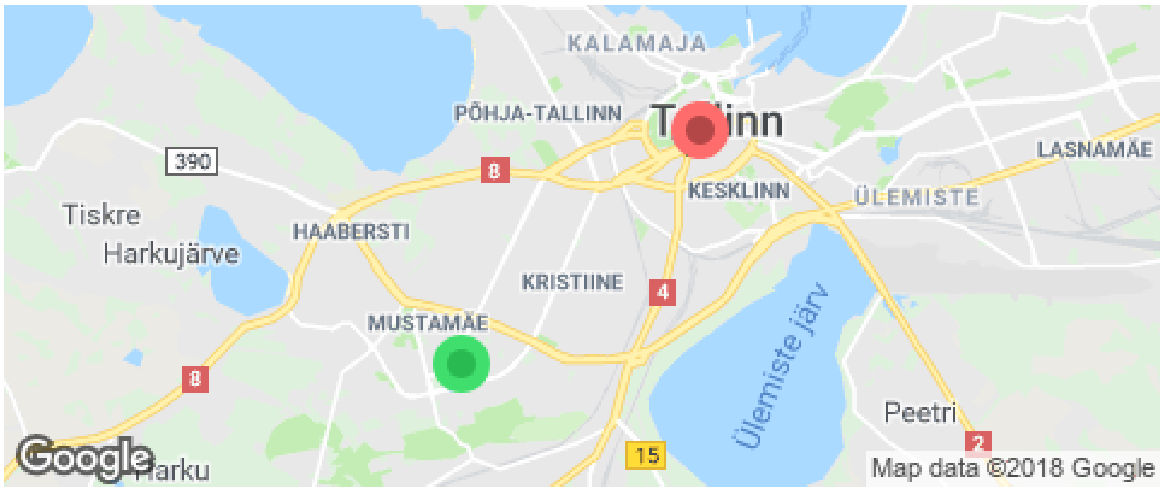

Figure 3. Travel time prediction between the source (green pin) and destination (red pin) is referred as estimated time of arrival (ETA).

In recent years, the growth of sharing economy with new ride sharing platforms and services have increased the importance and usage of travel time prediction services. The accuracy of the platform predictive algorithms defines an important part of the user experience, as the knowledge, how long the user has to wait for the vehicle and what is the estimated arrival time helps users in planning their own time.

There are many ways to approach the travel time estimation problem. Some of the methods are built bottom-up from the raw GPS data, while other use broader metadata collected from the previous trips [7], [8]. This task is considered difficult because estimation is influenced by many variables such as type of transportation, environmental changes and correspondence of the data and the environment, where it was originally collected. In order to accurately predict ETA from the raw geographical location data many methods are coupling input data with external data sources to make it link with the environment [9]. In addition, various external sources which influence the travel time in the real life scenario can be used in order to further enhance the quality of prediction.

2.3 Travel Time Prediction or Estimated Time of Arrival (ETA)

One of the approaches, which can be considered as state of the art for ETA calculation and has influenced this research is published by researchers from DiDi AI Labs [7]. DiDi Chuxching Technology Co. is one of the largest transportation network companies reportedly valued after the last funding round in 2018 to be worth $56 billion [10]. Their proposed pipeline has been used as a basis for this research (Figure 4). The pipeline consists of four different data sources, including map information, GPS data, order context and augmented data.

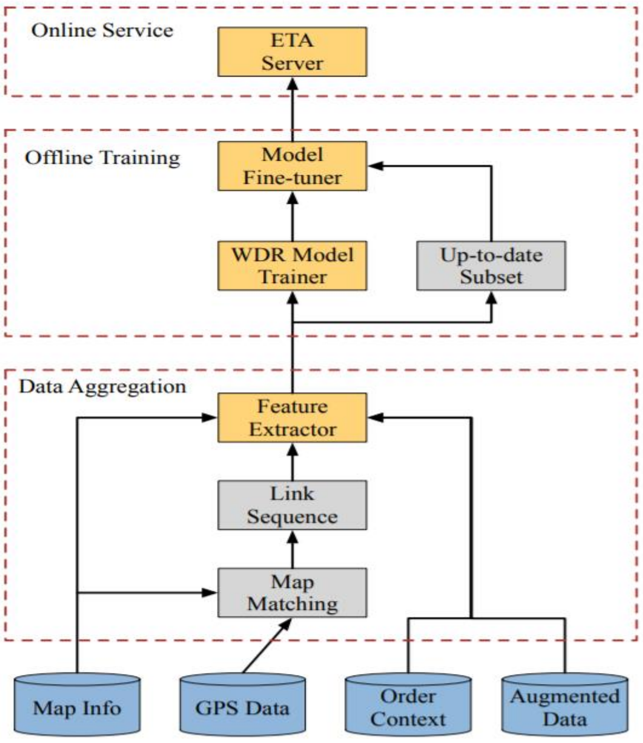

Figure 4. Travel time prediction pipeline proposed by DiDi AI Labs [7].

The GPS data is coupled at the first stage with the map information to do map matching. On the next stage, sequence is linked and trajectories are formed between map matched samples. Third stage combines features from all data sources, including order context and augmented data, which are joined before feeding them to feature extractor module. Once the features are extracted the larger module called data aggregation is completed, followed by the next large module of offline training.

Offline training section introduces the Wide-Deep-Recurrent (WDR) model, which comprises of four pieces [7]. The wide network part contains affine transformation and cross product and takes both dense and sparse features as the input. The deep part contains Multi-Layer Perceptron (MLP) and the recurrent part contains Long Short Term Memory neural network (LSTM) which handles sequential features. All the results from three sub models are orchestrated by the Regressor using mean average percentage error (MAPE) loss. This model is depicted on the Figure 5.

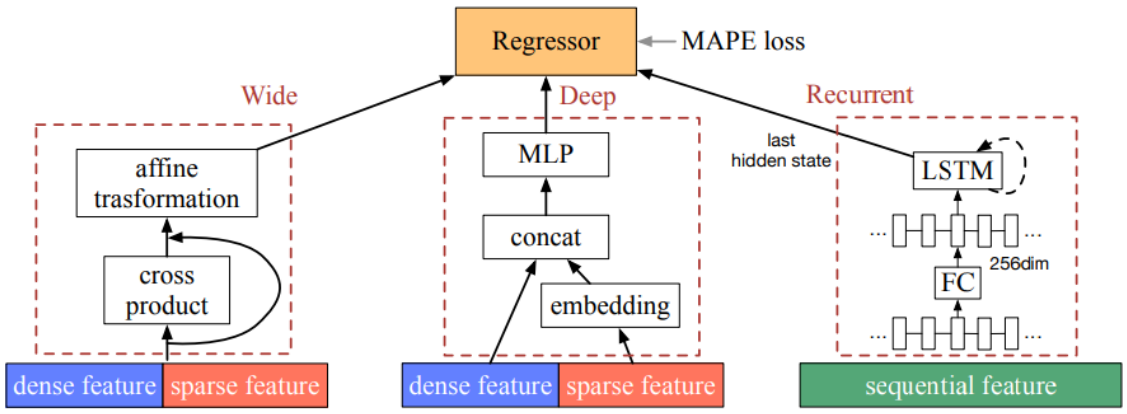

Figure 5. Wide-Deep-Recurrent network architecture [7].

The motivation behind using such complex model is to overcome the issues, which each of these models separately have. This model balances memorization with the recurrent network, which also handles sequential data. Generalization and representation abilities are covered with the deep and wide learning models, which all are orchestrated by single Regressor using mean absolute percentage error (MAPE) loss function. They claim that their proposed solution is used on real time online learning setup and been deployed on Didi Churching platform, which is servicing millions of users every day.

The aforementioned article provided comparison between their model and the competitive solutions. It was shown that their method outperformed all existing solutions, but the difference between the results is relatively small. The difference between mean absolute percentage error (MAPE) in their proposed method and the gradient boosting was less than 2.4%.

Another approach of ETA calculation that can be considered as state of the art is proposed by scientists from Pennsylvania State University and from Twitter Inc. [8]. This method is similarly to our proposed method relying on raw GPS data mostly collected by taxi drivers in New York City, but it takes different approach. Most of the proposed solutions, including ours, are route based methods, which take the designated route and calculate travel time for each of the segments or sub sections and then extrapolates based on this information. This approach, on the other hand is inspired more by human thought process, which means if we are asked how much time does it take to drive to any destination, we are not dividing this problem into numerous smaller problems, but rather relying on our previous experience on this field and trying to guess, how much time would it take to go somewhere similar, where we have been and then adjust it according to the original problem.

Authors of the article are calling this method a neighbor-based method. However, taking just average from all past trips to the neighboring area would yield quite bad results, because it does not take into account any traffic density. In order to overcome this issue, all of the neighboring trips were weighed either according to the distance between the origin and destination points or average speeds across the distance. As the experiments showed, distance between source and destinations for neighboring rides were both very computationally expensive and did not give any additional accuracy. Average speed on the other hand was much better correlated with travel time and yielded better results. The average speed for the neighboring trips is found by averaging all trips near the timestamp of neighboring trip.

Authors of the aforementioned article used speed reference table to adjust the traffic density according to the time difference between trips (Figure 6). The speed reference values are plotted on the figure below where the estimated average speed is denoted with blue dotted line and the actual measured speed is denoted with red line. Relative average speed computation is efficient because it could be done offline and reference values have to be calculated periodically.

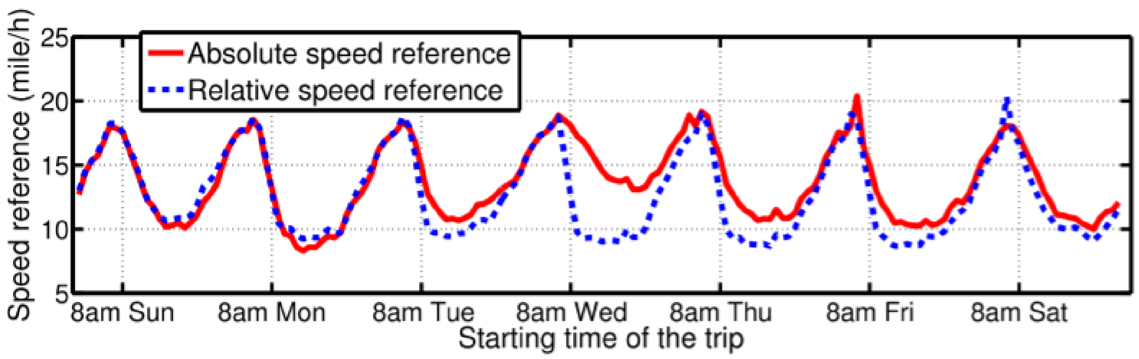

Figure 6. Average speed reference table for neighbor-based ETA method. Red line is denoting speed measured from traffic. Blue line is the reference used in the article gathered from the training data. This graph is covering Christmas week, where the traffic density is much lower on Wednesday because people are celebrating Christmas [8].

This method, however, as shown on the Figure 6. does not capture irregularities, such as national holidays. In order adapt better to the abnormal situations, the authors of the article proposed another method, which divides the whole time range into smaller time slots. Average speed of each slot is calculated by taking into account previous slots, seasonality and random noise. This information is fed into ARIMA model, which is predicting future values for the series. Finally, region information is added to the model, which is important if area is large and traffic density is not uniform across the whole area.

Results show that this method is a good baseline and in many cases can outperform far more complex methods which are route based. Also this method is simpler implementation wise than route based solutions and does not depend on map matching quality and suffer from data sparsity as much as most route based solutions. This method however has also some major limitations, which prevents it from being used in certain situations. For example, in a situation where two parallel routes lead to the same destination but only one of the routes is congested, would this method predict similar travel time for both. This limitation can be considered fairly important for services which quality and client satisfaction heavily depend on travel time estimation. The next section will discuss thoroughly our proposed method, which is also addressing this problem and therefore could be more suitable for ride sharing service application.

3. Methodology Adopted

This paragraph gives an overview of the data used for ETA calculation and validation, as well as describes in detail all the steps needed to pre-process the raw GPS data before applying it on our proposed method for arrival time prediction.

3.1 Introduction

There are various ways to predict the travel time. Our method belongs into a family of route based methods. This means that after specifying source and destination, the route is first identified and then travel time of each of the segments is calculated by aggregating historical data. Our proposed method relies on simpler baseline model and comparison of different feature vectors is used to estimate how much different features are influencing the prediction accuracy. For the evaluation of various features, different metrics, such as mean squared error (MSE), mean average percentage error (MAPE), mean absolute error (MAE) and other custom metrics were used.

In order to provide accurate estimation for the travel time, the floating GPS data has to be coupled with external data sources in order to take into account environmental factors. The largest environmental factor, which cannot be extracted directly from raw GPS data is the actual route in the road network. In order to recover original route from noisy GPS data the pre-processing step called map matching has to be carried out on the data to match it to the underlying base map.

The initial research plan of this thesis can be roughly divided into six steps (Figure 7). The first stage is pre-processing and cleaning the dataset. The second step is querying external data and matching the original raw GPS points to the underlying map. The underlying map used in the current thesis is from Open Street Map (OSM) database [11]. The third stage is reconstructing the routes in case the data density is too low. Fourth step would be estimating the traffic status for each segment in order to enhance the accuracy of the predictive model. Fifth step is for finding the optimal solution to fill the missing data and avoid data sparsity which is a very common problem with geo-spatial data, where center of the city is very densely covered and some parts of the suburbs does not have any sample coverage. The final step is to estimate traffic density and estimate travel time based on the knowledge gained in previous steps.

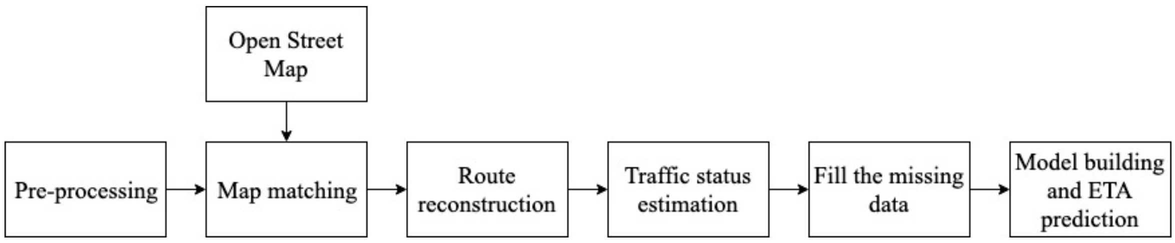

Figure 7. Workflow of our proposed method for estimating time of arrival.

3.2 Data used

The data used in this paper was provided by Bolt (former Taxify), an Estonian ride-sharing company through the collaboration with the Institute of Computer Science in the University of Tartu. The GPS data was collected from taxis and only by handheld devices such as mobile phones. The raw data used for this research contains only GPS coordinates, timestamp, and order id to link samples into sessions. The quality of the raw data was inconsistent as the handheld devices used for collection vary largely in quality and manufacturers. In addition, there could be fluctuations of sampling frequency and quality even during single collection.

Sample dataset provided by Bolt contained 3.7 million GPS samples which were grouped using order id. Sequence of the dataset was defined by the timestamp. Dataset contained 5500 rides which were later divided into training and testing sets. In addition, it contained device speed in kilometers per hour recorded by the device GPS. This information could be also extracted from the data, if timestamps are provided, although it is less accurate because of GPS fluctuations.

To validate, if our proposed method of ETA is applicable with different quality of data, additional data of T-Drive Trajectory Sample dataset published by Microsoft in Shanghai was used [12], [13]. The main difference between Bolt’s and Microsoft’s data is frequency of sampling. In addition, the raw data of the T-drive dataset is missing some of the crucial information, which is widely used in order to enhance the precision of preprocessing stages, such as bearing and the original order of the GPS samples. The bearing could be inferred from the existing data using timestamps and further preprocessing stages such as map matching.

3.3 Pre-Processing

Pre-processing stage is the first step of the ETA prediction method described in current thesis. This stage starts with grouping the collected samples by order id and then sorting the samples by timestamp in ascending order, which enables to form a trace of the initial route. Secondly, bearing information is added, which can be obtained by looking the next sample point and calculating the angle between the points. In this calculation, it is important to consider the earth curvature, if the distance is larger than five meters. The earth curvature can be calculated using Haversine formula [14]. With smaller distances the error is relatively small and is outweighed by the overall noise. Therefore, the earth curvature can be excluded from the calculation, in case of smaller distances, in order to decrease the computational complexity.

3.4 Map Matching

In this section, the floating GPS data is coupled with external map data by mapping the original GPS points to the underlying map. In our case, Open Street Map (OSM) platform was used. It is an open source community driven project to create free editable maps [15]. This platform provides a convenient API to download map layers according to the predefined bounding box. Unfortunately, there are some limitations of the file size which can be queried at once. Therefore, smaller pieces had to be used for combining the topology of the map. For larger areas, such as Shanghai or Bucharest the topology had to be downloaded in parts with small sleep timer between the requests to prevent OSM server blacklisting the IP address as it resembles denial of service attack. There are libraries, which make the whole process easier, such as Overpass API which has Python wrapper named Overpy [16]. Unfortunately, tested libraries were not well optimized and in case of querying a large dataset it was more efficient to include timeouts of one second in fixed interval of time. Tool for downloading OSM map in tiles and combining them later to form comprehensive map was developed by Intelligent Transportation Systems lab.

OSM represents map topology using a graph notation [17]. Each road is split into smaller pieces called segments, roads with curvature are split into multiple segments. Segments correspond to edges in our topology graph. Between the segments are nodes, which connect multiple segments together. Nodes might be intersections, but do not have to. Nodes and segments can have additional information, for example information about the speed limit or the length of the segment. Nodes have also information weather the node corresponds to crossing in the traffic and weather the crossing is regulated with traffic lights. On the Figure 8 there is a sample of OSM graph where each segment is color coded depending on the length of the segment.

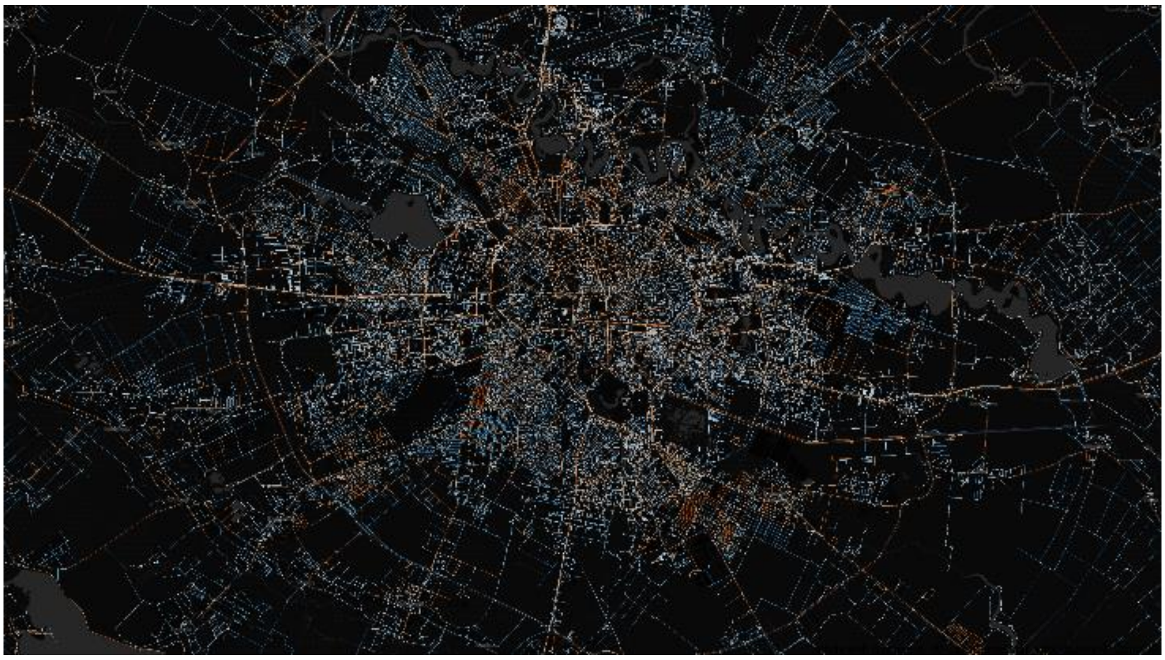

Figure 8. Plot of the Bucharest OSM graph where segments are color coded based on the length.

Map matching is a process, where each sample pair latitude and longitude location will be matched to the base road segment [18]. It might seem like an easy task, which could be solved by brute force, but reckoning each device is recording the location with approximately 2Hz frequency, it becomes infeasible solution in the real-world scenario. The computational complexity exceeds any real world use case time boundaries and in this case even with small datasets.

Additional complexity of this task comes from the ambiguity of the process, if some of the samples are in between two feasible routes. In some cases, the noise makes it very difficult to determine, if the samples, which are close to each other in space, but have time difference, belong to the same segment. Such situations occur relatively often, for example during U-turns or in the districts where streets are very narrow and close to each other. Such case is depicted on Figure 9, where rightmost sample P3 could be matched both branches B and C.

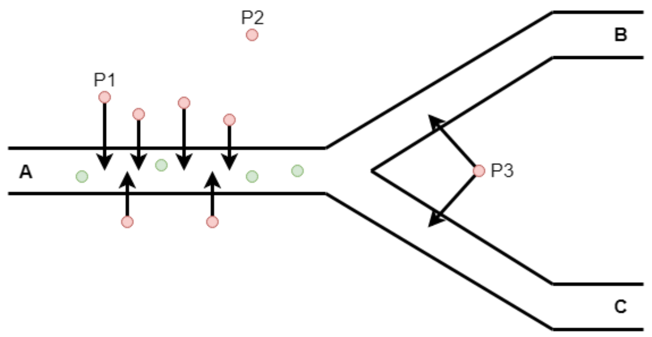

Figure 9. Map matching process where GPS samples, which are not on the road, are matched to the underlying road network. P2 is not matched due to exceeding the orthogonal distance threshold to the road. P3 introduces ambiguity as it could be matched to B and C.

To overcome the problems mentioned above, a relatively simple algorithm, developed by Intelligent Transportation Systems (ITS) lab from the University of Tartu, was used in current thesis. This method was favorable, because of the efficiency and computation complexity. Although, this method was more efficient, but less accurate, it was used in order to maintain relatively short feedback loop, as most of the computations were carried out on a personal computer.

Aforementioned method uses R-tree to index spatial data in order to avoid calculating orthogonal distance from every sample to every segment [19], [20]. The road segment with smallest orthogonal distance is matched to the sample. In this experiment, distance threshold of 10 meters was used in order to prevent from matching very noisy samples and generating larger errors. For example, in the case depicted on Figure 9, the point P2 was not matched to the existing road network due to exceeding the orthogonal distance threshold.

As state-of-the-art solutions, there are more sophisticated approaches, which are using either machine learning or additional external data sources and sensors to determine the correct road segment for matching the sample. For example, one of the solutions that ITS lab members have proposed uses mobile phone sensors together with GPS signal to determine vehicle location more precisely and improve map matching accuracy [21]. This method considers vehicle speed, bearing and both longitudinal and latitudinal gravitational force to enhance the quality of GPS signal and predict precise location of the device in the road network. Adding additional sensors enables to foresee some of the future actions. For example, changing to the leftmost lane before the left turn is very difficult to detect using GPS signal, but can be extracted using accelerometer and gyroscope. This method uses Kalman Filter in order to extract important information from very noisy gyroscope signal [22].

3.5 Route Reconstruction and OSRM

After matching the data to an underlying map, route reconstruction stage is needed to fill in the missing data, because at some cases data can be sparse. Route reconstruction is a pre-processing procedure for recovering the original route from a set of possible routes [20]. For example, when using the aforementioned Microsoft dataset, the gap between the GPS samples is relatively large, reaching to even more than six minutes [12], [13]. If the data would be dense, the assumption that the vehicle trajectory is similar to straight line, could be used. In case of sparse data, such as the Microsoft dataset, this assumption cannot be used, because in six minutes the car can travel considerable amount of distance and make numerous turns. Without knowing the trajectory of the vehicle, it is also difficult, if not impossible, to estimate the correct distance and speed. On the other hand, without knowing or estimating speed per segment it is difficult to estimate traffic density, which is one of the most important features in the case of this paper.

In order to resolve the problem mentioned above and get more accurate estimation of the route, additional information about the traffic is needed. In an example depicted on Figure 10, the information about one-way traffic helps to reconstruct the route of a vehicle with sparse data.

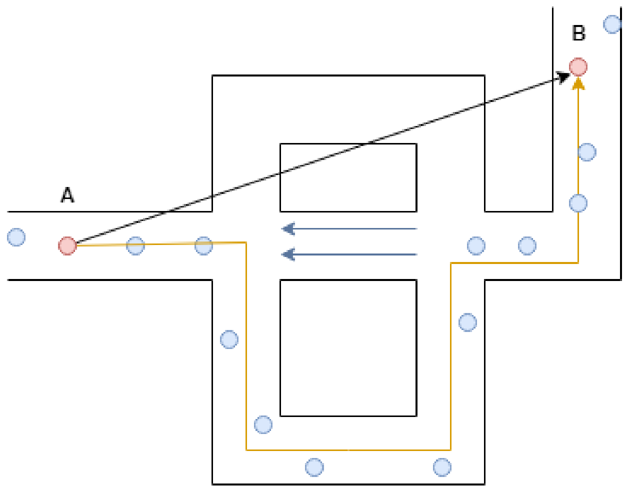

Figure 10. Route reconstruction of sparse data. The route reconstruction is the process of predicting the original route based on the map data and samples. Blue circles represent dense data samples and red circles represent sparse data samples. The orange line resembles reconstructed route. Double arrows mark the one-way traffic.

In order to get more accurate estimation of the actual route, a routing algorithm on top of the OSM map data is needed for trajectory estimation. The more thorough description of the Open Street Map (OSM) platform was provided in the previous sub-section 3.4. In case of our proposed method, route reconstruction was performed using Open Source Routing Machine (OSRM) [15].

OSRM is an open source routing engine for the Open Street Map platform and provides API to host routing server and query routes between two or more coordinates [15]. It provides implementation of various routing protocols, such as Multi Level Dijksta (MLD), which is widely used [23]. In addition to the fastest route, OSRM provides alternative routes as well. It is important because the exact route and distance between the samples is not known, but the travel time information is available. For example, on the Figure 10, there are multiple available routes and it is important to pick the one which is most probable to be the original one. In order to increase the likelihood of picking the original route the distance and estimated time of arrival is calculated for each of the alternatives and the one which matched the closest time to original samples is picked. This method drastically increases the data density and helps to estimate speed and distance of the vehicle more precisely.

Unfortunately, the aforementioned OSRM platform does not provide Open Street Map (OSM) way id-s which are important for later stages for augmenting the data. The following sub section will cover retrieving the information about the road segments, which carry the additional features such as speed limit information and whether the road is only one way.

3.6 Traffic Status Estimation

One of the most important factors in travel time prediction is the traffic status estimation. Traffic status depends on many external and environmental factors such as weather, specific route and time. There are many alternative ways to estimate traffic density and status. Simpler methods use analytical approach, while more complex state-of-the-art methods use, for example, graph neural networks [24], [25].

Graph neural networks are particularly well fitting for this specific task because of their nature. One of the most natural ways for representing the road networks, as discussed before, is the graph representation. The graph neural networks are a special kind of neural networks which operate on graphs [25]. The main idea behind graph neural networks is to provide a way to take the input graph and predict the next state of the graph vertex attributes and edge attributes using the knowledge of global state. This does not allow to modify the graph by adding new nodes or edges, but enables to learn and predict new states of the current graph. Such setup fits perfectly for predicting the traffic status in urban conditions where network itself is fixed, but the status of edges and nodes changes to reflect the current traffic density.

At the current test setup, the dataset did not have the labelled data about the traffic density with necessary fidelity to use it with graph neural networks described in the articles mentioned before, but it may be possible to adopt this method in the future [24], [25]. Therefore, an analytical approach using speed performance index to calculate congestion status was chosen [26]. This method could be optimized by vectorising all computations and to perform well, even on low compute power devices.

Our proposed method uses travelling speed from the data and maximum permissible road speed, which can be obtained from the OSM platform. However, the OSM map does not have speed limit information about all areas. In order to use this method, it was decided to fill missing values with default speed limit in urban areas in this region. Only urban areas were considered and therefore the induced error was within acceptable bounds.

The first step of our proposed method, was to group data by segments and calculate average travel speed per segment according to current timeframe. In this case, 60 minutes was chosen for the timeframe length according to the suggestion from the original article and by considering the amount of data and data density available for the test setup [26]. The second step was calculating the speed performance index R_v. The speed performance index was calculated as follows:

$$R_v = \frac{\bar{v}}{V\_max} * 100 \ (1)$$

In the formula $\bar{v}$ denotes average speed per segment at current timeframe and V_{max} denotes the maximal permissible speed at the segment. The next step was to divide segments into four categories according to the predefined rules from the Table 1 below [26].

Table 1: Evaluation criteria for determining traffic status on the segment [26].

Speed Performance Index R_v	**Traffic State Level**	**Description of the Traffic State**
[0, 25]	Heavy Congestion	The average speed is low; road traffic state is poor.
(25, 50]	Mild Congestion	The average speed is lower than usual; road traffic state is worse than average.
(50, 75]	Smooth	The average speed is higher; road traffic state better than average.
(75, ∞)	Free flow	The average speed is high; road traffic state is good.

In order to determine the road segment congestion, the next step was calculating the proportion of non-congested state, which was calculated using the following formula:

$$R_{NC} = \frac{t_{NC}}{T_t} \quad (2)$$

where R_{NC} denotes proportion of non-congested state, t_{NC} denotes duration of non-congested state in minutes. The segment was considered non-congested, if the speed performance index R_v was in range $(50, \infty)$. T_t denotes the length of the whole observation period in minutes. This gives the proportion of how much time during the observation period was this segment traffic state smooth or free flowing as stated on the table above.

The road segment congestion index was calculated using the speed-performance index and the proportion of the non-congested state of the road segment. The formula for calculating the road segment congestion index is following:

$$R_i = \frac{\overline{R_v}}{100} * R_{NC} \quad (3)$$

were $\overline{R_v}$ denotes average speed performance index of the road segment during the timeframe. This could be obtained by taking larger timeframe than for calculating the speed performance index and taking mean of the value. Road congestion index is a good estimate about the traffic density in the area. If the index value is small, then the road segment is heavily congested, larger value indicates better traffic conditions. This method was preferred, because it takes into account speed limit information. Average speed might not be very informative in most cases, because it also considers vehicles which are waiting for the client and does not consider the information about speed limits.

In case of Figure 11, the average speed was calculated by grouping the data by segment id-s and then taking the average of all speeds weighted with the length of segment. This however does not work well in the conditions of ride sharing platforms, because as seen on the Figure 11, the main roads have seemingly higher average speed than the side roads, indicating that side roads are congested, which is not actually the case.

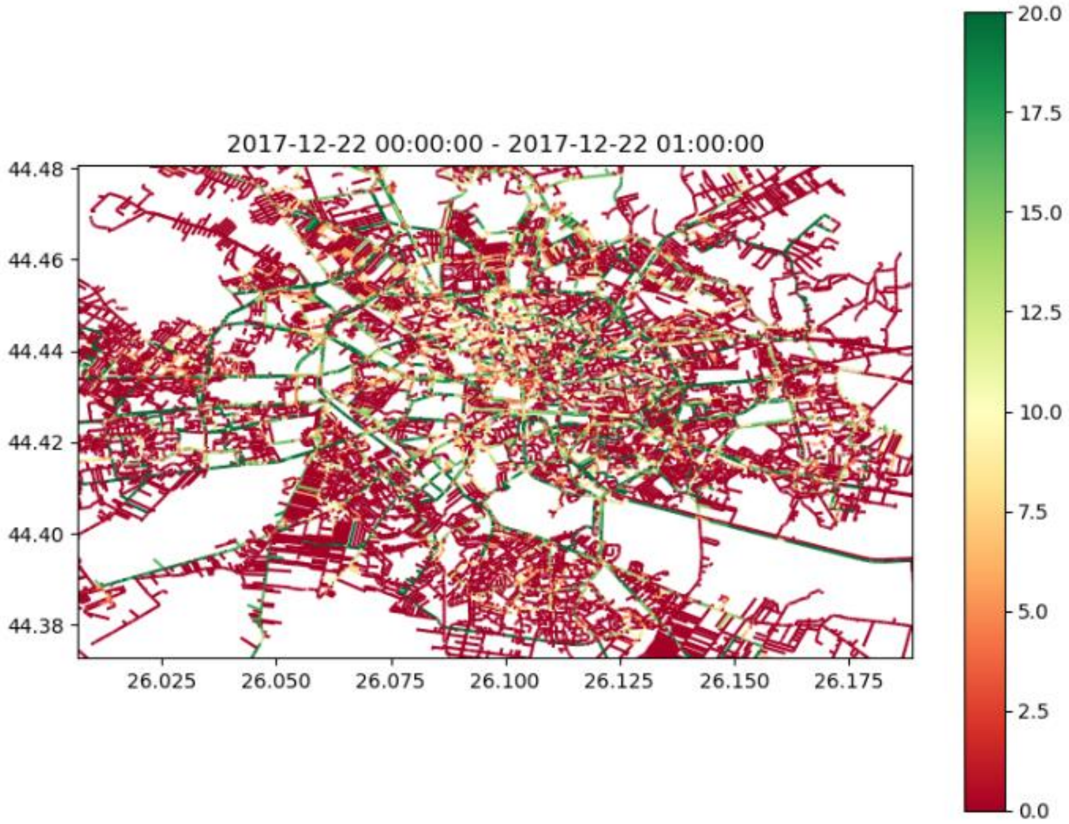

Figure 11. Average speed in the center region of Bucharest. Average speed in kilometers per hour is color-graded on the right side of the figure.

The reason behind it lies in the dynamics of ride sharing platforms nature. Clients source or destination is usually at the side road, which inherently makes the driver wait there for picking up customers or for dropping them off. This all is recorded as the ride and therefore considered in the average speed calculation process. The main road where the average speed is higher than the side roads, is not congested according to this plot. Unfortunately, it only tells that the relative number of samples, where vehicle is moving on this segment, is higher than the number of vehicles standing on the segment.

If the same area and roughly the same time frame is taken and the speed performance index is used to estimate the road segment congestion, the result is different (Figure 12). In this case, the speed limits in the area are also considered and the result is weighed by the number of the samples of the specific segment. As seen on the Figure 12, suburbs and the side roads of the city are not congested and there is more traffic in the center of the city. This corresponds more realistically to the actual situation of the traffic at the given time point.

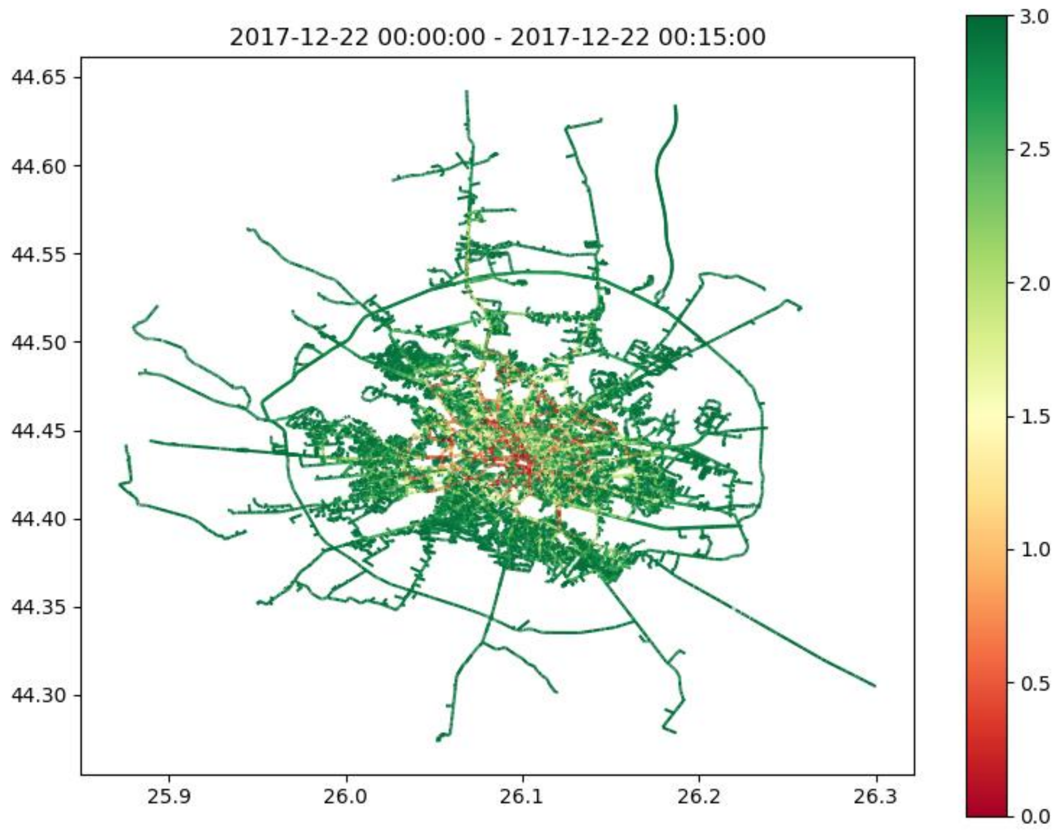

Figure 12. Segment congestion indexes of Bucharest center in broader view. Color scale on the right side of the figure is reflecting speed performance index.

Looking at the congestion indexes in more detailed view, the traffic is much more congested on the main streets and there are some slight signs of congestion on the larger roads which connect different districts (Figure 13). The timeframe in current cases, is taken around the midnight, so it makes sense that there is more traffic in the middle as more people would like to visit the bars and other attractions around the city at that time.

Road congestion index can be incorporated into the travel time estimation by calculating the average congestion index along the route as well as the average speed along the route.

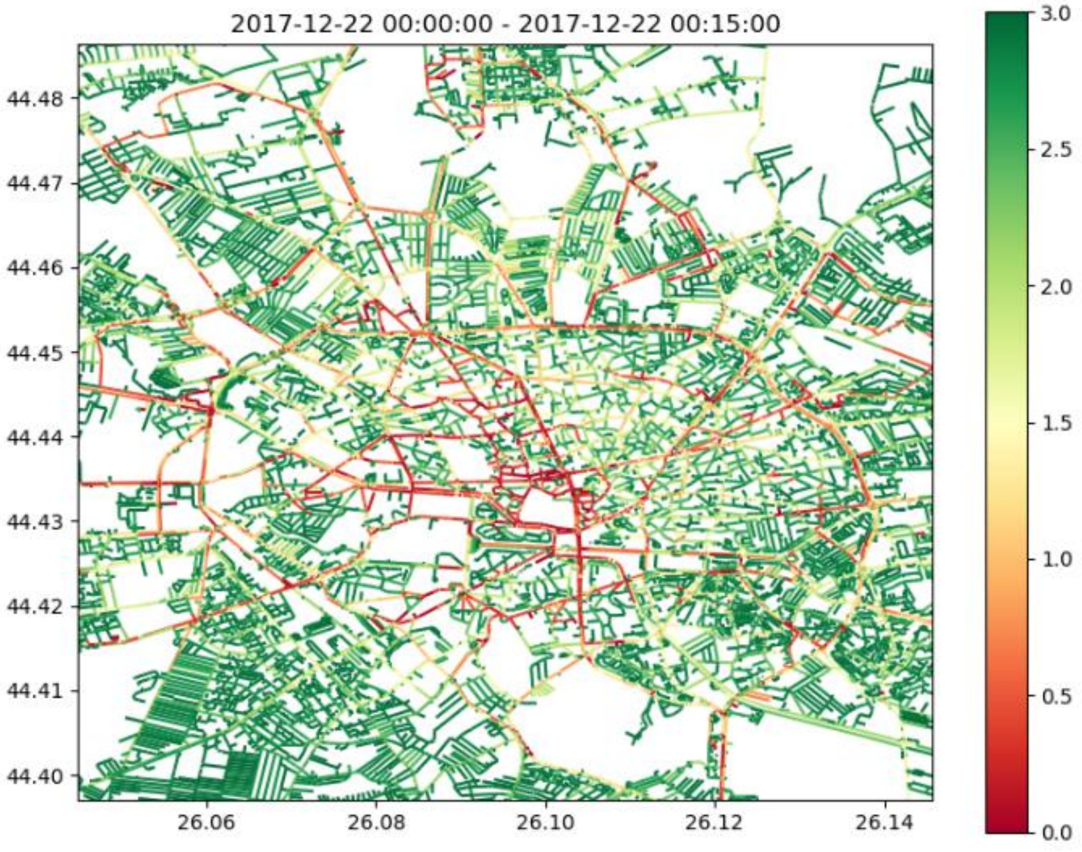

Figure 13. Congestion indexes of Bucharest center in more detailed zoom level.

3.7 Gathering Additional Data

There are many different factors, which influence the travel time as discussed before. The map matching in the section 3.4 and traffic evaluation in section 3.6 has already been covered. However, there are many other factors, which should be considered to provide good estimate for travel time.

One of the major factors, that should be considered is the weather. With heavy snow or even rainstorm, a short trip might take considerably more time than usual. In order to use that information, it has to be queried from the provider. There are many companies who are specialized for providing API for weather data, such as *AccuWeather*, *OpenWeatherMap* and *WeatherBug*. The query and response are encoded in JSON format and all of the aforementioned platforms provide convenient API to request the information.

In current work for additional weather information, it was decided to use the aforementioned *OpenWeatherMap*, that provides reasonable number of free queries, which is enough to validate if it is beneficial for providing accurate estimates. It was decided to use the ambient temperature in Celsius, rain volume for the last 3 hours, snow volume for the last 3 hours and wind speed in meters per second.

In addition to the weather information our hypothesis was that number of street crossings is correlated with travel time and thus has good predictive qualities. Number of street crossings was extracted from Open Street Map structure. Performance and predictive qualities for each of those features is discussed in the upcoming sections.

3.8 Building The Model and Predicting the Travel Time

All the steps of preprocessing described in the previous sub-sections are necessary to extract useful information from the raw GPS data in order to prepare the data to be in a suitable form for a machine learning model (Figure 14). As previously discussed the raw GPS data is coupled with external map data for map matching and for feature extraction. All the features describing traffic and route are extracted by combining the original input data to additional data source from Open Street Maps. After feature extraction, the next step is to build a model which is capable of producing accurate predictions based on those features.

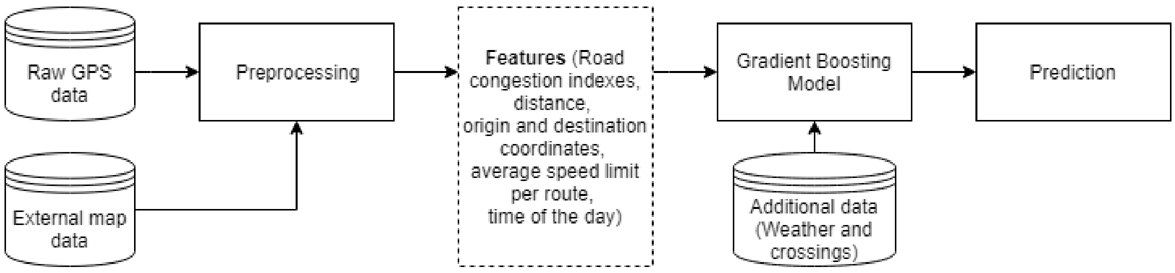
Raw GPS data
External map data
Preprocessing
Features (Road congestion indexes, distance, origin and destination coordinates, average speed limit per route, time of the day)
Gradient Boosting Model
Additional data (Weather and crossings)
Prediction

Figure 14. Pipeline for data moving from preprocessing to model.

Although there are many alternative ways for building the model for travel time prediction, for our proposed method, it was decided to use the out of the box solution of Gradient Boosted Regressor (GBR) from Scikit Learn and from LightGBM to focus on which features contribute more on giving a good prediction for the travel time [27], [28]. These models were preferred because they were also mentioned in the Didi Chuxing article and performed relatively well compared to their solution, which is considered to be state-of-the-art at the moment [7]. In addition, the motivation behind choosing Gradient Boosted Regressor was related to the amount of data available for this thesis and the motivation of analyzing feature importance, which is difficult with black box models, such as neural networks. In the case of such amount of data, the long short-term memory (LSTM) network coupled with multilayer perceptron (MLP) would over fit the data and perform worse on the new real world data. Experiment was run on a small LSTM model and it was found that the volume of data was not sufficient in order to make the model generalize and outperform simpler models. Some experiments were also conducted on smaller models such as two layered MLP with one thousand iterations. MLP model was capable of producing slightly better results than current LightGBM implementation of gradient boosting Regressor, but as our motivation was to compare also feature performances more descriptive models were preferred.

The LightGBM model was configured to have "gbdt" boosting type which is traditional gradient boosting decision tree. Mean average percentage error (MAPE) and mean absolute error (MAE) were used for optimization. MAPE was chosen because it is intuitive for time series measurement as all the values are positive and it is biased to low forecasts as by the definition of MAPE the under-forecast is bounded by the zero but over-forecast is unbounded. Other metrics, such as l1, l2 distance, were also used during experiments, but best results were achieved with MAPE. Other parameters for the model included number of leaf nodes which was bounded by 21 to prevent overfitting. Learning rate which determines impact of each tree in the final outcome was set to 0.05. Max depth of the tree was limited to 5 also in order to reduce overfitting. Feature fraction parameter which determines proportion of features that are selected randomly in each iteration of building trees was set to 0.9 which means 90% of parameters were set randomly in each iteration. Bagging fraction which specifies the fraction of data used in each iteration was set to 0.8 which means 80% of the data is being used. Bagging frequency was set to 5, which means bagging was performed at every fifth iteration.

The dataset was divided into two equal sets, training and testing. For resembling the real world scenario more precisely, the dataset was split by time, which means half of data was used for training and the second half for testing. Dataset was split temporarily into half to have similar distribution of congested hours and non-congested hours in both training and testing set. For the validation, several different metrics, such as MSE MAPE, MAE and R^2 score were used to evaluate each model's performance.

4. Results

For evaluating the work of the ETA prediction workflow and our proposed method, various types of analysis was performed, including analyzing histogram of errors and various metrics to understand the nature of mistakes for each model. The method described in this paper was compared to ETA prediction solutions by OSRM and one commercial API provided by major global maps provider service (MGMPS). All methods were configured to be as comparable to each other as possible, which included using additional traffic related data to enhance external model performance and setting additional waypoints to force routing algorithm to take as similar route as possible. The first part of this section will cover experiment setup describing both the data and models including external model setup. The second part will compare the performance of our proposed method to OSRM and MGMPS.

4.1 Experiment Setup

Benchmarking our proposed travel time estimation method involved analyzing dataset of 3.7 million GPS samples grouped into rides provided by Bolt. As described before map matching was applied, as well as filtering out all GPS samples, which distance to the nearest possible road was more than 10 meters. Those samples could not be matched to any segment. This resulted 3.5 million samples, dropping only 100 000 samples which were less than 3% of the original dataset.

The next filtering step was performed to remove the samples on the segments, which did not have speed limit information for the congestion index calculation purpose. Those samples were still used for the average speed analysis, but could not be used for the traffic density analysis method described in section 3.6. Congestion indexes could be calculated for roughly 2 million samples. Those samples were grouped to roughly 5500 rides.

Models were evaluated by dividing the whole dataset into temporarily half. This resulted two equal parts, first part was used for training and last part for testing. This split was made to have similar distribution of the periodic daily congestion flow in training and testing sets. All preprocessing steps were carried out separately on both sets in order to prevent data leakage from test set to the training set which would artificially improve the results. Cross validation was used on the training set in order to fine tune the model parameters and find the best performing model which was later evaluated on the test set.

As discussed in the previous section LightGBM implementation of gradient boosting regressor was chosen instead of DiDi Chuxching proposed deep learning architecture to evaluate feature performance and to understand more thoroughly why this method performs the way it does. LightGBM implementation of gradient boosting decision trees is both lightweight and capable, it managed to produce almost as good results as deep learning method proposed by DiDi while having much lower engineering complexity and better usability [7].

All models were fitted using the same features in the training data, containing traffic congestion information about last three hours and ongoing hour, average speed over all rides on the segments covered by this particular route, total length of the trip, speed limit information and information regarding the time of the day, year and location. In addition, latitude and longitude geographical coordinates of the starting and ending point was supplied to the model because this information is available also in the real world scenario.

The models were compared on the basis of mean absolute error (MAE), mean squared error (MSE) and minimal, maximal absolute error. In order to make both trained and pre trained models comparable, the traffic information was supplied to the external models whenever possible. Furthermore, additional waypoints were added in order to force external models to choose the original route. Additional traffic information improved OSRM method performance and made it more comparable.

Major global maps provider service API provided only three settings for the traffic model and several models for traffic mode such as driving, walking, cycling and transit. Three different models were available for traffic. The most pessimistic model from three performed best on our data. The aim was to make each model perform as good as possible within reasonable amount of work. Supplying our own data to the MGMPS model was not possible and therefore this model was adjusted to give as close predictions to the original situation as possible using other means. MGMPS API supports only present or future timestamps for the trip start time. Therefore, date from 2019 was used from the same month as the original ride and on the closest date with same day of week as the original ride was used. In order to force MGMPS API to take similar, if not the same route, additional ten coordinates were uniformly sampled from the original ride dataset and set as the waypoints. Furthermore, additional parameter was configured, in order to prevent routing algorithm from optimizing the sequence of those waypoints. These additional steps were taken to ensure that both the traffic load and routes are as comparable as possible for the experiments.

Additional tests on Microsoft dataset of Shanghai taxi rides were conducted [12]. Our pre-processing methods were applicable for even very sparse data where gaps between samples exceeded six minutes. This was achieved using route reconstruction methodology described in the section 3.5 before. Using our proposed route reconstruction method, the density of data was increased by 5.5 times. Unfortunately, the experiments did not provide comparable results due to lack of external data such as speed limit information and for some cases very large noise causing unsatisfactory map matching quality.

4.2 Results and Feature Performance

Performed experiments proved that using only average speed information can be considered a viable way to estimate traffic status and therefore travelling time. On the other hand, more complex methods, which take into account speed limits and other external factors are capable of outperforming simpler methods relying solely on average speed. On the Figure 11, it is apparent that the average speed was uniform in almost all areas and did not outline congested roads because this dataset is collected by taxis, which were either picking up or dropping down customers mostly at suburban areas. Also, as the maximum permissible speed is different in many areas, the average speed alone cannot be used to compare areas with different speed limit. In order to overcome those issues, speed performance index was used, which takes into account the speed limits and the time constraints, how long the traffic was congested or free flowing. The method works by finding speed performance index from the actual speed and speed limit and then finding non congested time duration of the segment to calculate road segment congestion ratio. This method was also more thoroughly explained in the section 3.6. As discovered later, the model which used speed performance index and road segment congestion index instead of average speed, did also outperform it in terms of prediction accuracy.

Each sample in the testing set, which corresponded to the original taxi ride was coupled with road congestion indexes on the same roads one hour, two hours and three hours before the start of the ride. In case of missing data, current hour road congestion index was used to fill the missing values. This time range was chosen by experimenting with range of values from one hour to twelve hours. Road congestion indexes more than three hours back from the starting time did not seem to have any impact on the travel time prediction most likely because of the nature of traffic which is ever changing.

Initial experiments show that congestion indexes heavily contribute in the precise ETA estimation. For estimating the feature importance, the model was fitted ten times with random seed and average importance was calculated. The most important feature with the average importance of 0.51 was the total trip distance which was measured as the total length of the OSM segments of designated route (Figure 15). The second and third most important features, latitude and longitude points of the destination, were related to the destination location. These features were so important probably because there are predefined and more popular drop off points which exist in both training and testing dataset. The road congestion index three hours before the start of the ride was fourth most important, leading over average congestion index two hours before the start of the ride. Average speed limit information of the designated route was next important for the model as it contained valuable information weather the route traversed highways or other similar areas. In our dataset average speed limit did not correlate to the average speed too well and therefore speed performance indices were more important. Speed performance indices of the current hour until the start of the ride and previous hour were also used but those did not have as high importance as the features listed above had.

Furthermore, some experiments were conducted using additional weather and traffic data, such as number of regulated crossings on the route. Features selected from the weather data were ambient temperature in Celsius, rain volume for the last 3 hours, snow volume for the last 3 hours and wind speed in meters per second from the time of starting the ride. However, those did not improve the result as the feature importance for all of these features were smaller than 0.005. Therefore, those features were left out from the final model. One of the reasons why those features were not as useful as traffic state describing features was because the time period in the training data was relatively short and according to the reports weather was very uniform over that period of time. Number of regulated crossings did not influence much as well, probably because during inactive times regulated crossings turn off traffic lights and become unregulated crossings. Also the traffic congestion information was already extracted.

To evaluate how much does each feature contribute to the prediction, experiments with different variations and combinations of the features were performed. As discussed before, the road congestion index is calculated for each segment per each timeframe and then averaged over the route. While including only the average speed and distance of the route, ETA was predicted with mean absolute error of 607 seconds. While using the speed performance index instead of average speed, the mean absolute error was reduced to roughly nine and a half minutes. Road congestion indices helped to reduce it further to the current best result of 227 seconds. Although, this is still more than four minutes off, it can be considered as a good result according to the sparsity of the dataset, which introduces huge amount of ambiguity and does not give precise information about the route.

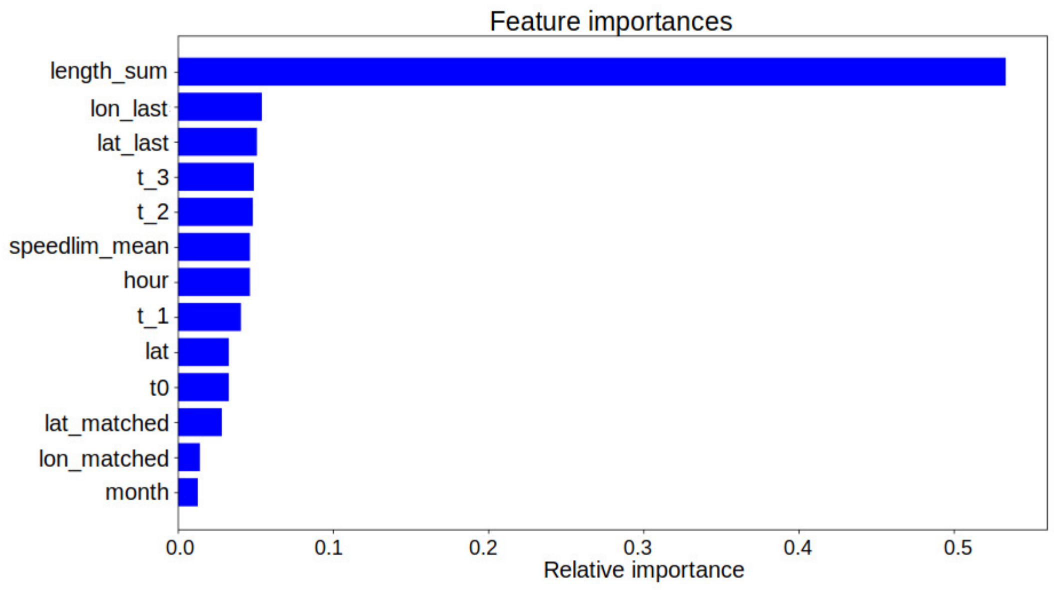

Figure 15. Average feature importance over 10 test runs.

4.3 Comparison with Existing Methods

Our proposed ETA prediction method was compared to two other well-known and proven solutions to verify, whether the result provides good estimations and to validate the efficiency of the method.

External methods were compared to the baseline method and our proposed method. The baseline method was linear equation between the total trip distance and average speed. The trip distance was taken as the sum of all the OSM segments' length. Average speed was estimated by averaging all samples per segment where average speed was larger than one kilometers per hour to exclude vehicles standing still. Then average was taken from all segments covering trip route to get the average speed as scalar value. Dividing total length to the average speed was already good estimate for the travel time. The graph below depicts histogram of errors for our baseline model (Figure 16). Errors were measured using mean absolute error metric. The distribution is unimodal and slightly right skewed, median value for the distribution is around eight minutes and 30 seconds. Mode value of this distribution is at around nine-minute mark. This model performs very well on this dataset because it is based on the same problem specific data and as this data is temporarily connected it has similar bias caused by environmental factors, which are very difficult to include in the external models.

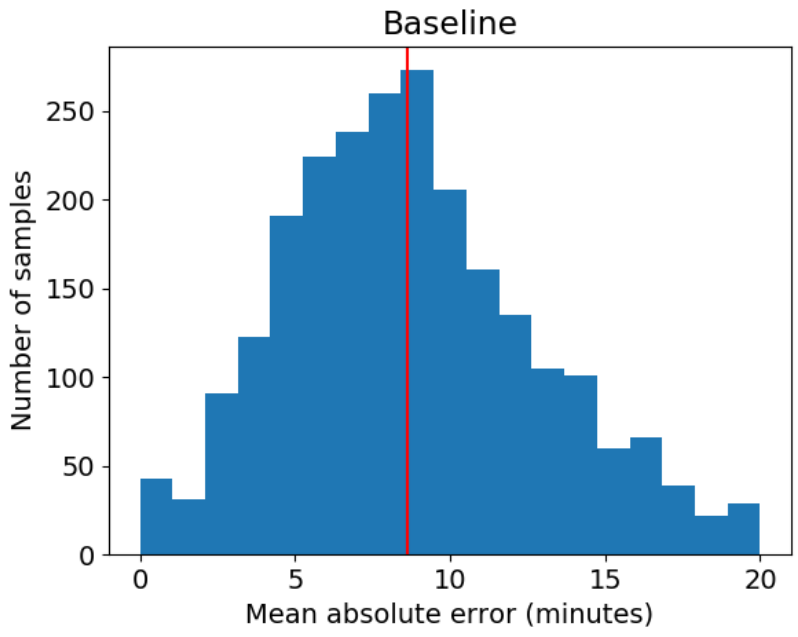

Figure 16. Histogram of errors for baseline model measuring mean absolute error (MAE). Red line denotes median value.

In addition to the baseline value, all methods were compared to our best performing model, which was LightGBM implementation of Gradient Boosting Machine. This model was more thoroughly discussed in a section 3.8 above. The error distribution histogram of this model measured as mean absolute error (MAE) is depicted below (Figure 17). This distribution is also unimodal and right skewed, but with much shorter tail and more samples on the left side. This indicates that the mean average error of the whole model is smaller than the baseline. The median value for this model is around three and half minute mark. The variance of this distribution is much lower than the baseline distributions variance and there are very few samples which have error larger than 15 minutes.

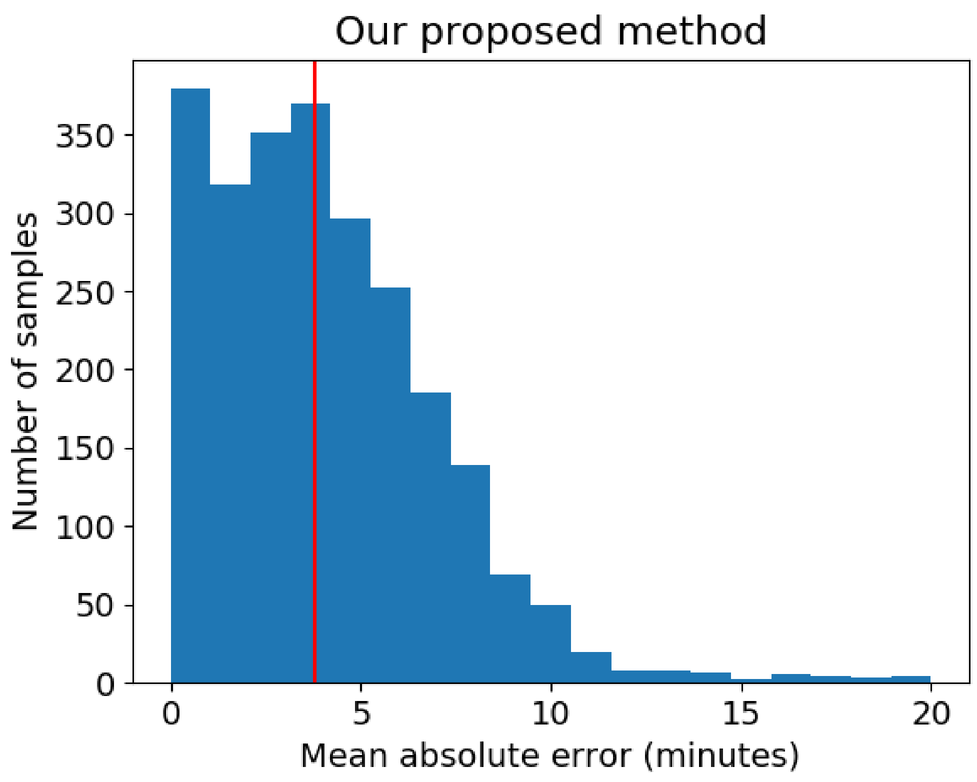

Figure 17. Histogram of errors for our best performing model (LightGBM Gradient Boosting Machine) measuring mean absolute error in minutes. Red line denotes median value.

The full error distribution of our proposed solution is unimodal normal distribution which is slightly skewed towards the left (Figure 18). The mode for this distribution is not perfectly aligned to the point zero as it is around five-minute mark. The median value for this distribution is two minutes and 30 seconds, which means that this model is on average too pessimistic and therefore predicts that arrival time to the destination is later than the vehicle actually arrived. Optimistic model in our interpretation would do the opposite by predicting the travel time to be shorter than the actual trip takes. Effect that there are more pessimistic estimations than positive estimations is most likely caused by outliers in the data. As this distribution is left skewed there are some errors, which indicate that the model has made errors which are larger than 20 minutes. Such large errors are existing only on the optimistic side of the distribution meaning that in case of large errors, less travelling time is predicted than it actually takes to travel that distance. This behavior might be caused by MAPE loss function which is biased towards low forecast as the error of under forecasting is limited by zero.

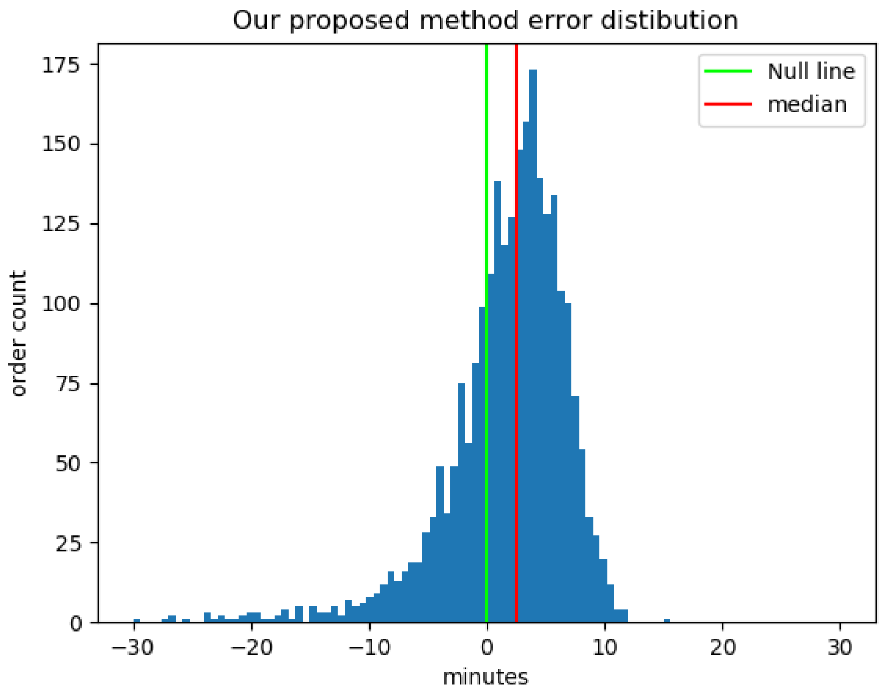

Figure 18. Error distribution for our best performing model (LightGBM Gradient Boosting Machine) measuring the error in minutes as difference between our predicted arrival time and the target value. Red line denotes median value and green line denotes the point zero.

4.4 Comparison to Open Source Routing Machine (OSRM)

The first comparison was performed with a solution provided by OSRM (Open Source Routing Machine) [29]. As mentioned before OSRM is high-performance routing engine, which also provides travel time predictions and various other services. This routing engine uses map information with the .osm or .pbf extension and converts it to its native format. The native format uses preconfigured vehicle type to adjust the road network to the selected vehicle type. Therefore, this routing engine can be used both for cars and pedestrians. In addition to the topological information, the OSRM routing engine takes source and destination coordinates as latitude-longitude points and configuration file, which determines the routing algorithm and traffic information. It was decided to define also the traffic information to make the comparison more even. Traffic information, in this case, was consisting of pairs for segments and their average speeds at certain time frame. This functionality is not configurable by default but can be defined using additional command line tool "osrm-contract".

In this experiment setup, the OSRM engine was configured with default car configuration and the same traffic data as in our proposed method. The main difference was between the features where the latter used speed performance indices instead of just average speeds. In addition, as the OSRM provides both the route and estimated travel time, the route had to be aligned as much as possible with the actual GPS trace. The most similar route was determined by the proportion of aligned segments in the OSM map and total distance.

Firstly, mean absolute error was calculated to compare ETA of our proposed method and the one provided by OSRM (Figure 19). Samples were sorted by MAE in ascending order on both results.

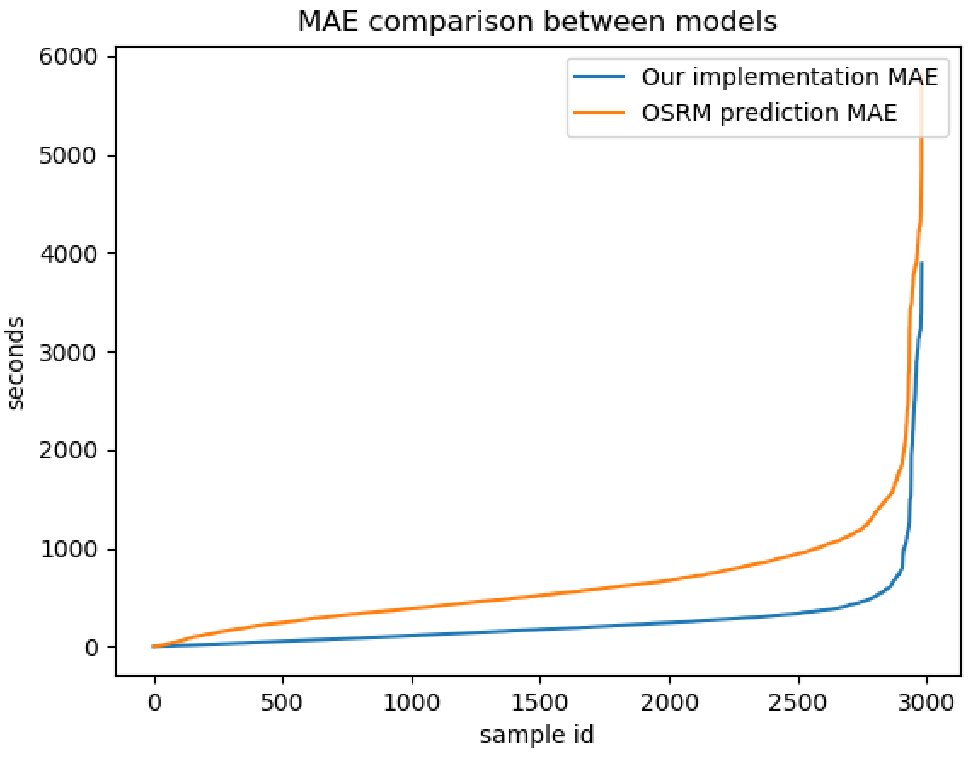

Figure 19. Travel time comparison between the implementations of OSRM and our proposed method. Samples are sorted by the value of mean absolute error in ascending order on both results.

Secondly, mean absolute error was calculated for both methods, which is visualized as histogram on the Figure 20. The scales between Figure 20A and 20B are synchronized in order to make them more directly comparable. The mean absolute error distribution in case of OSRM is resembling unimodal normal distribution, having the mean absolute error roughly 9 minutes and 20 seconds (Figure 20B). The mean absolute error in case of our proposed method is smaller for most of the samples compared to the OSRM prediction error, having the mean absolute error about 3 minutes and 48 seconds (Figure 20A). Distributions clearly outline that our proposed method is capable of predicting travelling times with better accuracy than OSRM. This is mostly caused by the additional information that it uses, such as speed limit information and information about the traffic density in specific regions and times. Also the dataset used for training our proposed method was specifically related to this problem and therefore we had much more information to train our model.

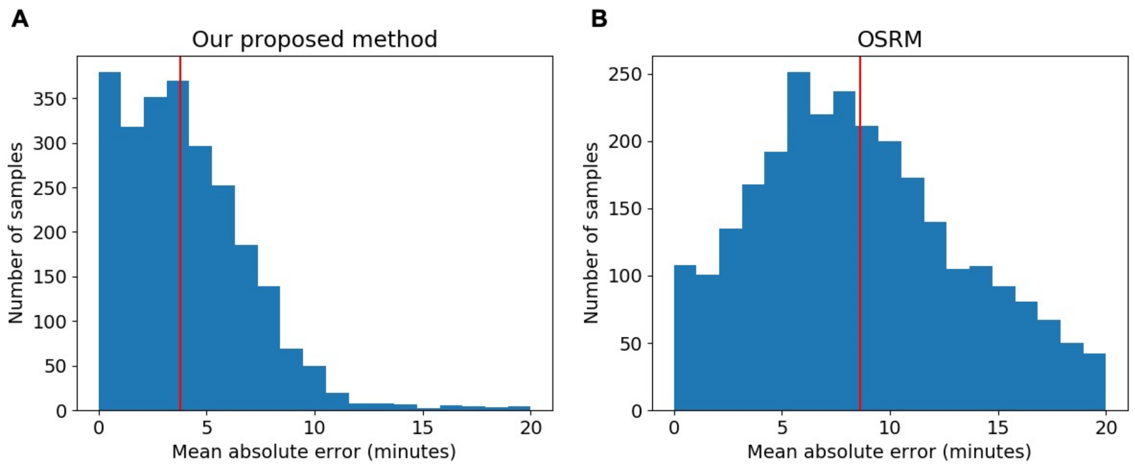

Figure 20. Distribution of mean absolute error of travel time prediction, measured in minutes. Red line denotes the median of the distribution. **A** Mean absolute error in our proposed method. **B** Mean absolute error in case of solution by OSRM.

The reason why OSRM does not outperform our solution is most likely based on the dataset and the base map. The figure below depicting the full error distribution of OSRM predictions follow Gaussian distribution and the mode of this distribution is around minus eight minutes, which means that the model is on average eight minutes too positive towards the prediction (Figure 21). This means that on average the model will predict that the travel time is 8 minutes shorter than the actual target value was. The median value of this distribution is around minus 9.3 minutes.

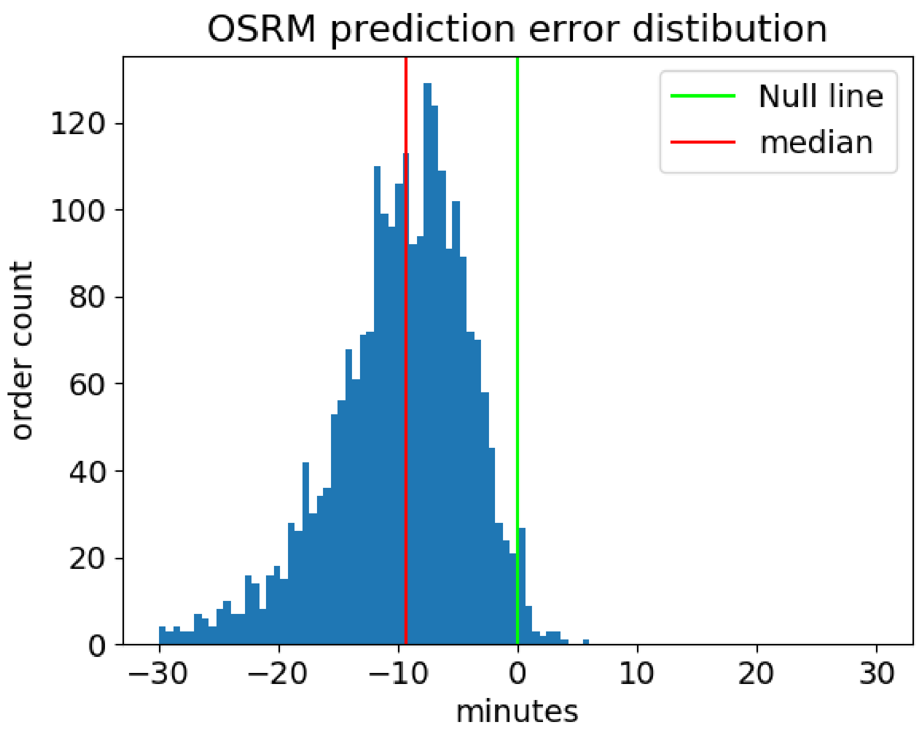

Figure 21. Error distribution for OSRM measuring the error in minutes as difference between OSRM prediction and the target value. Red line denotes median value and green line denotes the point zero.

4.5 Comparison to major global maps provider service

In the previous chapter our proposed method was compared to OSRM which is well known and proven open source solution. To make the comparison against competitive solutions more holistic, our proposed solution was compared to major global maps provider service. This service has been commercially used in many logistics based applications.

Major global maps provider service (MGMPS) was evaluated similarly to OSRM using same metrics and methods. The only real difference was that MGMPS method could not be configured to the same extent as OSRM and therefore it was more complex to make this method comparable to our proposed solution. Details of configuring this method is discussed in the experiment setup section 4.1 above.

Based on our experiments the major global maps provider service predictions are very similar to the OSRM to the extent that the MAE between those two is under two minutes. MGMPS was expected to outperform our proposed method as well as OSRM, but as a surprise, it outperformed OSRM only by a small margin, beating mean absolute error by exactly half a minute. Error distribution histogram depicting mean absolute error of MGMPS is shown on Figure 22B. The distribution is less skewed than our solution and has shape of Gaussian distribution with mode around seven-minute mark and median value denoted by the red line at 8.8-minute mark.

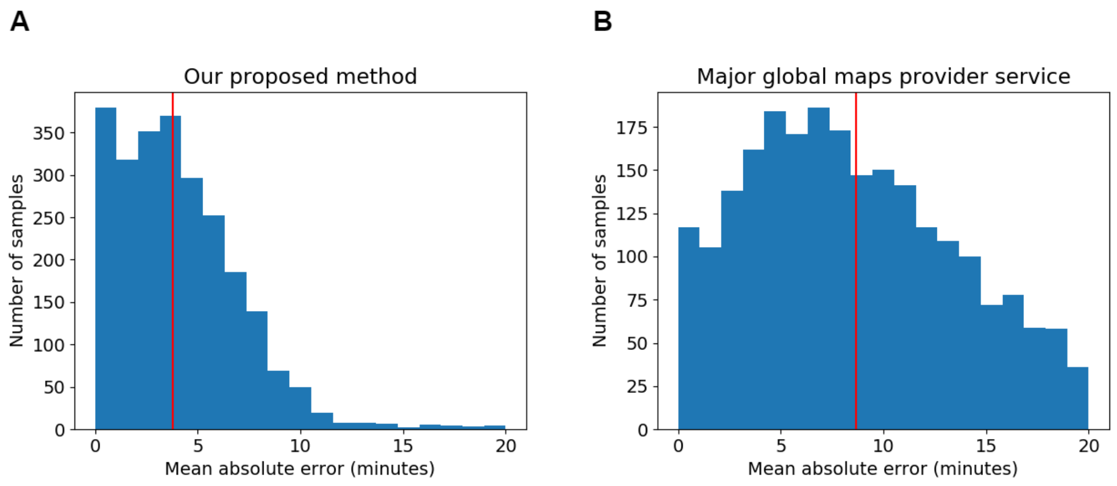

Figure 22. Distribution of mean absolute error of travel time prediction, measured in minutes. Red line denotes the median of the distribution. **A** Mean absolute error in our proposed method. **B** Mean absolute error for major global maps provider service.

At first such large error by the industry leading API provider seems odd. However, when looking the full error distribution, as difference between the prediction and target, it becomes much more clear, why this method did not perform that well on our data. The Figure 23 below depicts that the error distribution for MGMPS does also follow Gaussian form and the mode is around eight minutes. This means that this distribution is similarly shifted as the OSRM distribution discussed above (Figure 21). The reasoning behind it was also discussed in the previous subsection and it involved mostly base map changes and usage of target or problem specific dataset.

As mentioned above, major global maps provider service was configured to be as comparable to our proposed method as possible. However, as the time difference between the original ride on the supplied data and MGMPS ride starting time exceeded almost two years, it is still difficult to compare the two methods, as there is very high probability of extensive changes in the road network. Additional ten waypoints and constraint restricting to optimize waypoint order were added to force competitive algorithm to take similar route, but this did not guarantee that the same route was actually taken. Also, the traffic flow and roadworks heavily influence congestion rate and travelling time both, which are also not taken into account while comparing MGMPS, OSRM and our proposed method.

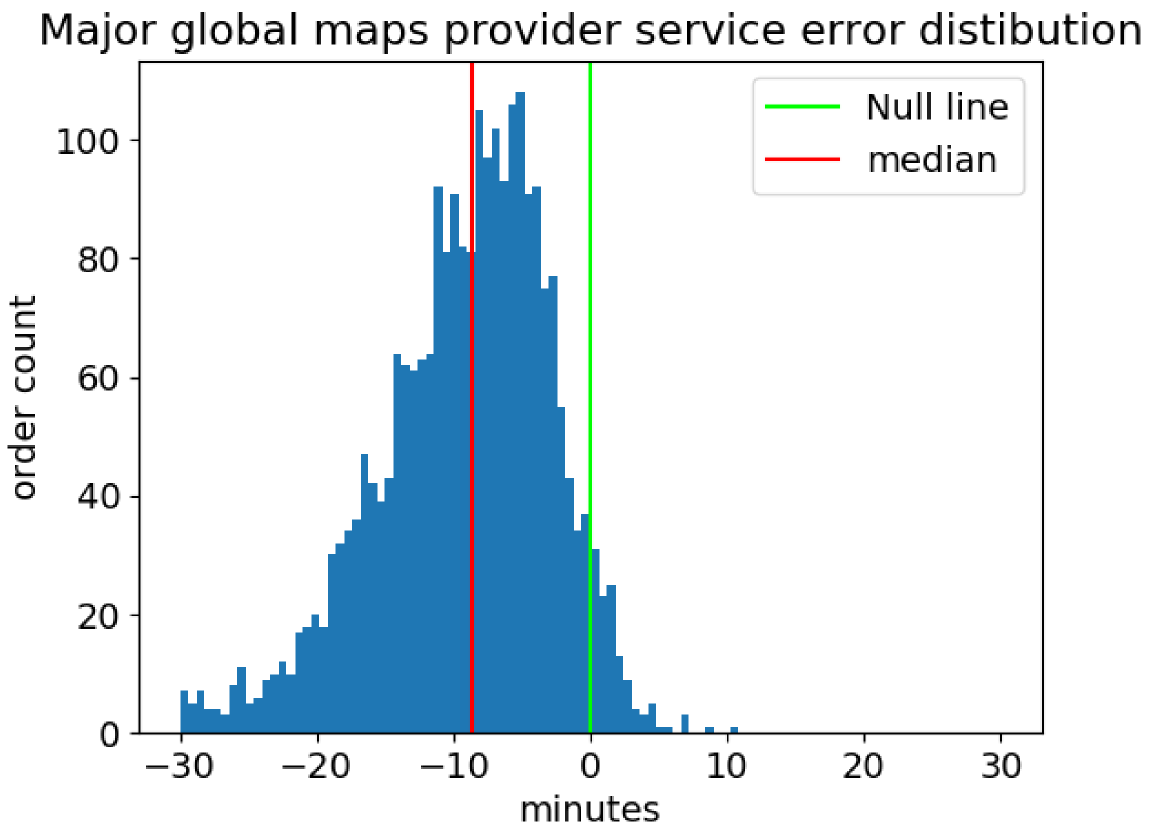

Figure 23. Error distribution of major global maps provider service measuring the error in minutes as difference between our estimated time of arrival and the target value. Red line denotes median value and green line denotes the point zero.

Comparison between all aforementioned methods (baseline, our proposed, OSRM and major global maps provider service) can be found from Figure 24 below. Our proposed method is outperforming all other methods, when comparing the methods by mean absolute error. The error lines between our proposed solution and other solutions are crossing each other at the leftmost part of the graph where errors are smallest. This means that in some cases OSRM, MGMPS and even baseline are capable of predicting with very high accuracy, but as the blue line does not cross with others, our method is almost constantly better on average than the others. On the same time, we can see OSRM prediction and major global maps provider service line crossing at 1800 which means that the commercial solution managed to outperform OSRM on the samples before that but slightly falls back at the worst cases.

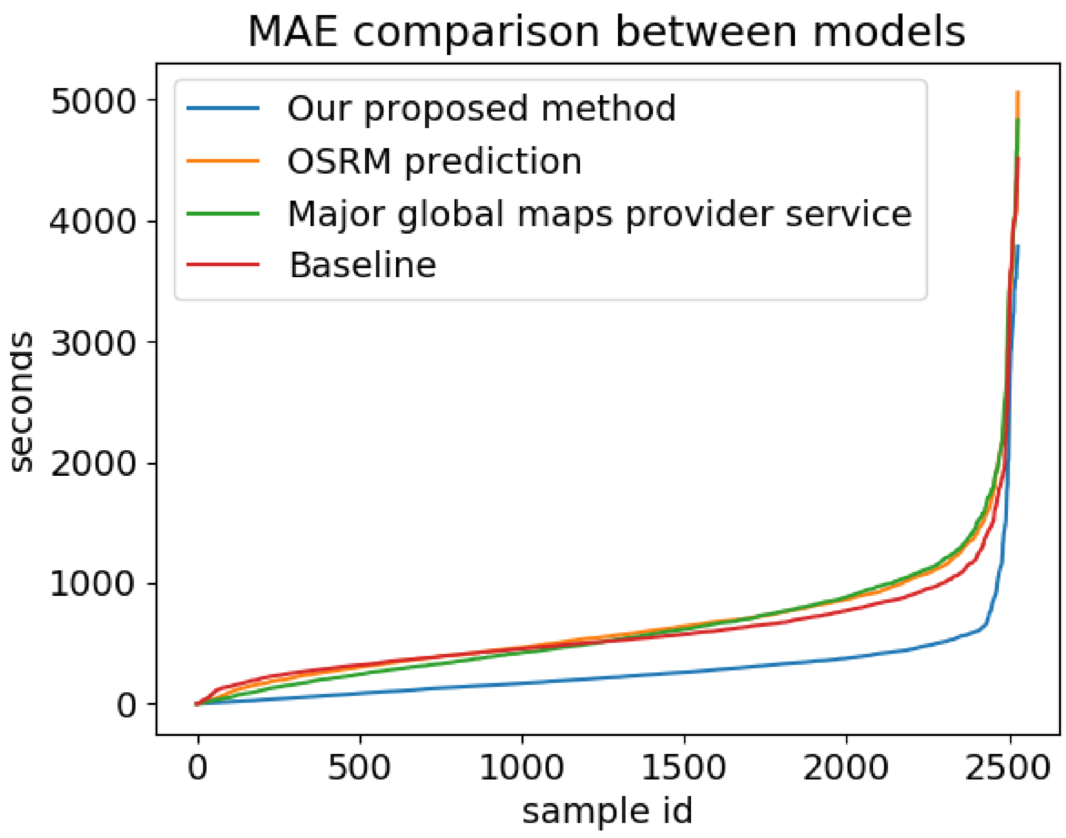

Figure 24. Mean absolute value comparison between all aforementioned methods including LightGBM implementation of Gradient Boosting Machine, baseline model, OSRM and major global maps provider service.

The baseline method performed fairly similar to OSRM and MGMPS. On the Figure 24 the range of small errors for the baseline model is still larger than competitive solutions almost to the midpoint of the x-axis where the increase of the error slows down and exceeds the accuracy of both external competitors.

In conclusion to this section, our proposed method has managed to outperform other competitive and widely used methods, but it has been done using specifically targeted dataset and has been evaluated on our own data. This means that our proposed method is more efficient for estimating travel time for certain types of transports in this particular location using both original and external data.

5. Discussion

Travel time estimation has become essential part of any logistics based business and therefore it has become a popular research topic. Our proposed method for travel time prediction belongs to route based travel time estimation methods family and covers all of the intermediate processing steps from raw GPS data to the travel time estimation based on historical data of the route subsections. The aim of this thesis is to propose a method for travel time prediction and to describe each of the preprocessing steps and their impact to the overall prediction quality.

Route based methods are highly sensitive to map matching quality and data sparsity, which makes the input data quality very important. Data quality used in this thesis can be considered very good and therefore simpler methods for some preprocessing stages could be used without sacrificing too much on travel time estimation quality. For example, simpler map matching technique using only orthogonal distance to the road was used in this thesis instead of ITS lab proposed method, which is using multi sensor fusion to overcome ambiguity [21].

Travel time is influenced by different kind of factors, that all contribute to the prediction accuracy. In our proposed method, total length of the ride was most predictive feature, as it is directly connected to the travel time. Average speed in the urban situation is relatively uniform and therefore the model could have just found the optimal average speed constant which is closer to the original ride speed than average speed in our dataset over all rides. Division between the total distance and this scalar provided very good estimate for the trip durations as it is directly connected through the laws of physics.

Road congestion index in this application was also performing well as predictive feature, because it managed to capture more information than other features such as average speed which was not used by the model. This feature already contained average speed information, speed limits and even temporal data about the congestion duration. The most important feature among speed performance indices captured traffic state three hours before the trip start, which reasoning is difficult to guess as the current hour congestion information seems to be more important and contain better information. Some parts of the reasoning might lie in the method of constructing this feature by filling missing values with current hour speed performance index.

For evaluation purposes our proposed method was compared to two other well-known and proven solutions, OSRM and commercial solution provided by major global maps provider service. Our method was able to outperform both of the aforementioned methods on the taxi ride data provided by Bolt. Both of the external methods were configured to be as comparable as possible. However, some variables could not be configured to be exactly on par for comparable methods. The biggest difference of prediction quality is most probably caused by the base map including roadworks changing the traffic pace. The base map has changed during the time period of roughly two years, which means extensive changes in the road network are highly probable. In addition, the major global maps provider API was configured to estimate traffic density but the distribution of errors might be shifted because the traffic estimation part of the model is insufficient or misconfigured by the author.

Another error for travel time estimation might be caused by the fact that taxi drivers pick up and drop off clients, which might take significant amount of time and is factored in the dataset. Unfortunately, there is no way of verifying this hypothesis as we had only records about the raw GPS data without any annotations about the driver whereabouts.

Both OSRM and major global maps provider service API-s was provided with ten additional waypoints and additional parameters to not optimize the route in order to force those methods to pick the same route. However, due to changes in the road network, the traffic flow density and accessible routes might have changed. In addition, external factors such as roadworks and public events are influencing traffic density. Unfortunately, due to the timeframe difference, those factors cannot be taken account of and it might introduce some ambiguity into methods.

Although, our proposed method outperformed the two other methods, it has its downsides as it is specific to input data, which was further discussed in the section 1.1. This means that the proposed method will be effective on taxi ride data on the specific region but might underperform in other types of applications. The compared major global maps provider service on the other hand is capable of performing well on wide variety of situations and locations without such preconditions.

Our proposed method cannot be considered optimal as it can be definitely improved. Nevertheless, it provides a solid baseline for problem specific datasets and could be used for commercial services as it outperforms more generic well known solutions. More detailed discussion how to improve this method is following in the next section.

5.1 Future perspectives

Currently some external factors that could be used were not included in the research, because of the lack of time. For example, Didi Chuxing article describing the estimation of arrival time suggests using external factors such as large public events, driver and passenger profiles and vehicle data [7]. The driver information coupled with vehicle data could be very important as the drivers have very different driving styles which could be integrated into the model [30]. In addition by using additional sensory information such as gyroscope, accelerometer and magnetometer the map matching quality can be further increased [21]. As discussed in the previous sections route based solutions are highly sensitive to map matching quality and therefore this improvement might produce significant increase in prediction accuracy.

In addition to introducing the new features for the task, the accuracy of the current model can also be improved. For example, deep learning models have proven to be efficient in such cases and could outperform our current version [7], [31]. The research in this field will be continued to further improve the results and get better accuracy. Besides improving the model, some further work could be done on the feature analysis by comparing model trained using only the total distance and temporal features to the external model which could give us additional information about this feature predictive properties.

Furthermore, for the future research, similar approach could be used on some other external dataset, that does not comprise of taxi ride data, in order to validate if our method can be applied to other kinds of transportation types as well. In addition, it would be very interesting to train the method based on some other type of vehicle such as bicycle and compare the results with well-known competitors such as OSRM or some other commercial service such as Here Maps.

6. Summary

Estimating travel time and time of arrival has become essential part for any logistics-based business which makes algorithms working with geospatial data popular topic in the industry. This thesis has been conducted in collaboration with Bolt which is one of the largest ride hailing companies.

This thesis is describing route based travel time prediction algorithm based on raw GPS data. The goal of this thesis was to analyze each of the preprocessing steps and to develop a coherent method to predict arrival time based on floating car GPS input supplied by Bolt. Our approach consisted of six distinct stages which were all more thoroughly described in section 3. The first part of this approach coupled the GPS data with external map information provided by Open Street Maps. The second part of the preprocessing was important to overcome data sparsity which is one of the main issues of route based approaches. Our proposed method used external routing engine to find potential routes and then chose the most plausible option. The most extensive preprocessing step was the traffic status estimation which estimated traffic density based on input data coupled with some external data such as speed limit information and each of the segments (sub-paths) lengths. Last part of our approach enriched the processed data and described the machine learning model which was used in this thesis for travel time prediction.

We achieved the original goal for predicting the arrival time based only on the raw GPS recordings. Furthermore, we found a way how to deal with varying quality of data and overcame the issues which were introduced by the sparsity of the dataset. In the thesis all preprocessing steps together with detailed descriptions of the problem were described.

In addition, to validate our proposed method efficiency it was compared to two other well-known and proven solutions. Our method was compared to Open Source Routing Machine and a commercial method by major global map provider service. External methods were configured to be as comparable to our solution as possible to the extent of supplying our preprocessed traffic data to comparable methods in order to improve their performance. Our method managed to outperform both of the external competitors on the provided input data.

In conclusion, our proposed method can be considered as a good route based estimation baseline for specific applications and even used for commercial purposes but in order to make it applicable to wider range of application future research is required.

7. References

[1] D. O. D. USA, “Global Positioning System Standard Positioning Service - SPS Service,” 2008.

[2] O. Montenbruck, A. Hauschild, P. Steigenberger, U. Hugentobler, P. Teunissen, and S. Nakamura, “Initial assessment of the COMPASS/BeiDou-2 regional navigation satellite system,” *GPS Solut.*, vol. 17, no. 2, pp. 211–222, Apr. 2013.

[3] C. Cai and Y. Gao, “A Combined GPS/GLONASS Navigation Algorithm for use with Limited Satellite Visibility,” *J. Navig.*, vol. 62, no. 4, pp. 671–685, Oct. 2009.

[4] C. Vicek, P. McLain, and M. Murphy, “GPS/dead reckoning for vehicle tracking in the ‘urban canyon’ environment,” in *Proceedings of VNIS ’93 - Vehicle Navigation and Information Systems Conference*, 2002, pp. 461,–34.

[5] Rohde and Schwartz, “Simulation of real-world conditions.” [Online]. Available: https://www.rohde-schwarz.com/pk/product/gnss-powerslave_63491-11461.html. [Accessed: 04-May-2019].

[6] L. Vanajakshi, S. C. Subramanian, and R. Sivanandan, “Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses,” *IET Intell. Transp. Syst.*, vol. 3, no. 1, p. 1, 2009.

[7] Z. Wang, K. Fu, and J. Ye, “Learning to Estimate the Travel Time,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD ’18*, 2018, pp. 858–866.

[8] H. Wang, Z. Li, Y.-H. Kuo, and D. Kifer, “A Simple Baseline for Travel Time Estimation using Large-Scale Trip Data,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–22, Jan. 2015.

[9] S. K. S. Fan, C. J. Su, H. T. Nien, P. F. Tsai, and C. Y. Cheng, “Using machine learning and big data approaches to predict travel time based on historical and real-time data from Taiwan electronic toll collection,” *Soft Comput.*, vol. 22, no. 17, pp. 5707–5718, Sep. 2018.

[10] “Is $80 Billion Valuation Achievable For Didi Chuxing’s IPO?” [Online]. Available: https://www.forbes.com/sites/greatspeculations/2018/12/24/is-80-billion-valuation-achievable-for-didi-chuxings-ipo/#206033ee6211. [Accessed: 15-May-2019].

[11] J. Topf and F. Ramm, “GEOFABRIK.” [Online]. Available: http://www.geofabrik.de/geofabrik/geofabrik.html. [Accessed: 08-May-2019].

[12] J. Yuan *et al.*, “T-drive,” in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS ’10*, 2010, p. 99.

[13] J. Yuan, Y. Zheng, X. Xie, and G. Sun, “Driving with knowledge from the physical world,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11*, 2011, p. 316.

[14] C. C. Robusto, “The Cosine-Haversine Formula,” *Am. Math. Mon.*, vol. 64, no. 1, p. 38, Jan. 2006.

[15] M. Haklay and P. Weber, “OpenStreet map: User-generated street maps,” *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[16] R. M. Olbricht, “Data retrieval for small spatial regions in OpenStreetMap,” in *Lecture Notes in Geoinformation and Cartography*, no. 9783319142791, Springer, Cham, 2015, pp. 101–122.

[17] OSM Community, “OpenStreetMap Wiki,” *OpenStreetMap Wiki*, 2015. [Online]. Available: http://wiki.openstreetmap.org/. [Accessed: 15-May-2019].

[18] N. P. Deasy, A. R. Gholkar, T. C. S. Cox, and M. A. Jeffree, “Tentorial dural arteriovenous fistulae: Endovascular treatment with transvenous coil embolisation,” 1999.

[19] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R*-tree: an efficient and robust access method for points and rectangles,” in *ACM SIGMOD Record*, 2005, vol. 19, no. 2, pp. 322–331.

[20] B. Y. Chen, H. Yuan, Q. Li, W. H. K. Lam, S. L. Shaw, and K. Yan, “Map-matching algorithm for large-scale low-frequency floating car data,” *Int. J. Geogr. Inf. Sci.*, vol. 28, no. 1, pp. 22–38, Jan. 2014.

[21] S. Plangi, A. Hadachi, A. Lind, and A. Bensrhair, “Real-time vehicles tracking based on mobile multi-sensor fusion,” *IEEE Sens. J.*, vol. 18, no. 24, pp. 10077–10084, Dec. 2018.

[22] E. J. Krakiwsky, C. B. Harris, and R. V. C. Wong, “A Kalman filter for integrating dead reckoning, map matching and GPS positioning,” in *IEEE PLANS ’88.,Position*

Location and Navigation Symposium, Record. "Navigation into the 21st Century"., 2003, pp. 39–46.

[23] D. Delling, A. V. Goldberg, T. Pajor, and R. F. Werneck, "Customizable Route Planning in Road Networks," *Transp. Sci.*, vol. 51, no. 2, pp. 566–591, May 2015.

[24] X. Wang, C. Chen, Y. Min, J. He, B. Yang, and Y. Zhang, "Efficient Metropolitan Traffic Prediction Based on Graph Recurrent Neural Network," Nov. 2018.

[25] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," Jun. 2018.

[26] F. He, X. Yan, Y. Liu, and L. Ma, "A Traffic Congestion Assessment Method for Urban Road Networks Based on Speed Performance Index," *Procedia Eng.*, vol. 137, pp. 425–433, Jan. 2016.

[27] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Nips '17*, p. 9, 2017.

[28] P. Prettenhofer and G. Louppe, "Gradient Boosted Regression Trees in Scikit-Learn," 2014.

[29] D. Luxen and C. Vetter, *Real-time routing with OpenStreetMap data*. Chicago, Illinois: Association for Computing Machinery, 2012.

[30] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1609–1615.

[31] Yanjie Duan, Yisheng Lv, and Fei-Yue Wang, "Travel time prediction with LSTM neural network," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1053–1058.

I. License