UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Ayobami Ephraim Adewale

Link Travel Time Prediction Based on O-D Matrix and Neural Networks

Master's Thesis (30 ECTS)

Supervisor: Amnir Hadachi, PhD

Tartu 2018

Link Travel Time Prediction Based on O-D Matrix and Neural Networks

Abstract:

In public transportation system, commuters are often interested in getting accurate travel time information regarding trips in the future in order to plan their future schedules effectively. However, this information is often difficult to predict due to the irregularities in travel time which are caused by factors like future weather conditions, road accidents and fluctuations in traffic demand. With the introduction of Intelligent Transportation System into public transport system, it has been easy to collect data regarding bus trips such as travel times data. The data collected can be used to make predictions regarding trips in the future by applying scientific methods like Kalman filter, machine learning and deep learning neural network.

The goal of this thesis is to develop a neural network model for predicting travel time information of a busy route using Origin-Destination matrix derived from a historical GPS dataset of the same route. The prediction accuracy of the NN model developed in this thesis was measured using Root Mean Square Error (RMSE). Analysis of the result showed that the model is sufficient for making predictions of travel time for trips in the future.

Keywords: Public Transit System, Intelligent Transportation System, travel time, Neural Network, Origin Destination matrix, Kalman filter

CERCS:P170

Reisiaegade Ennustamine Kasutades algpunkt-sihtpunkt Maatriksit ja Tehisnärvivõrke

Lühikokkuvõte:

Ühistranspordi kasutajad on tihtipeale huvitatud täpsest reisiajast seetõttu, et tõhusalt aega planeerida. Kuid ebaregulaarsete reisiaegade tõttu on seda üsna keeruline teha. Reisiaegade muutused võivad olla põhjustatud näiteks ilmastikuoludest, liiklusõnnetustest ja liiklusnõudlusest. Intelligentse transpordisüsteemi kaasamisega ühistranspordi süsteemi muutus hõlpsamaks bussireisi andmete kogumine, sealhulgas ka reisiaegade kogumine. Kogutud andmeid on võimalik kasutada tulevaste reiside prognoosimiseks, rakendades erinevaid teaduslikke meetodeid, näiteks Kalmani filtrit, masinõpet ja tehisnärvivõrke. Antud lõputöö eesmärgiks on luua tehisnärvivõrgu mudel, mis ennustab tiheda liiklusega teekonna reisiaega. Selleks kasutatakse algpunkt-sihtpunkt maatriksit, mis on koostatud sama teekonna kohta kogutud GPS informatsioonist. Ennustustäpsuse arvutamiseks kasutati antud lõputöös ruutkeskmist viga (RMSE). Tulemuste analüüs näitas, et antud mudel on piisav tegemaks tulevaste reisiaegade ennustusi.

Võtmesõnad:
ühistransport, intelligentne transpordisüsteem, reisiaeg, tehisnärvivõrk, algpunkt-sihtpunkt maatriks, Kalmani filter.

CERCS:P170

Contents

Abbreviations

APTS Advanced Public Transportation System

ATIS Advanced Traveler Information System

AVLS Advanced Vehicle Location System

GPS Global Positioning System

ITS Intelligent Transportation System

LSTM Long Short-Term Memory

MLP Multiple Layer Perceptron

NN Neural Network

OD Origin Destination

OSRM Open Source Routing Machine

RMSE Root Mean Square Error

List of Figures

List of Tables

1 Introduction

1.1 General Overview

The birth of public bus transportation system has contributed immensely to the growth of cities in developed countries like New York in United State of America, Tallinn in Estonia, Beijing in China and Munich in Germany. It has been used to reduce traffic congestion, reduce the emission of greenhouse gas that are harmful to the environment, improve access to opportunities within connected cities, boost economy growth and lastly, improve quality of life in general. However, most people are still reluctant to take public bus and thus, prefer to go around with their private vehicles. This is quite understandable because the system is easily affected by weather, traffic signals, traffic fluctuations, peak hours and road incidents which often leads to delay in set schedules, irregularities in journey times and bus arrival times. In some countries, the difference between the set schedule and actual arrival times could be up to ten minutes or more. This does not only affect the plans of the commuters but also reflects on the economy growth. Based on this, there have been increasing demand for scientific techniques to solve the lingering problems.

1.1.1 Factors Affecting Travel Time

There are a lot of factors that lead to the variation of public bus travel time and some of these factors are not measurable. Some of the factors also depend on the type of route being considered. *Bargegol et al* in [3] identified some factors that do affect travel time and some of them include passenger boarding time, changes in average speed, pedestrian walking speed and bus capacity.

Fosgerau et al in [4] defined Travel time as a combination of free traffic flow travel time and delays which contribute heavily to the variation in travel time. These delays are then broken down into two types, systematic delay which is the expected delay and unexplained delay which is regarded as unforeseen delay [4].

Unforeseen delays are unpredictable delays that are non-recurrent. That is, these delays do not happen every time and there is no available pattern that can be traced or matched to predict such delays. Type of delays that fall into this category are delays caused by accident since most accidents are always unexpected. Delays caused by unannounced on-road workers is also an example of unforeseen delays. This kind of scenario often happen when road maintenance are announced and the trip schedules does not reflect such delays to the commuters who regularly ply the affected route. Delays caused by traffic light are also in this category because throughout the lifetime of a trip, a bus might arrive at a junction when the traffic light is red on more than one occasion. In busy cities, a red traffic light might take up to 2-3 minutes before switching to a green light and this often reflect on the bus travel time. The experience or psychological behavior of a driver can also lead to a delay and this cannot be measured. In summary, unforeseen delays are not part of a trip day to day characteristics and thus, are always very difficult to measure.

Systematic delays are delays that can be explained by observing the characteristics of the trip [4]. Delay time of systematic delays can be predicted just by observing patterns of previous trip along the same route. Example of delays that fall into this category are delay caused by number of passengers boarding or alighting the bus at bus stops along the route. High number of passengers boarding or alighting the bus will increase the dwelling time at that bus stop and will also reflect on the total travel time. *Bargegol et al* in [3] mentioned that it is possible to reduce the delay caused by dwelling time at stops by increasing the number of entry and exit doors on a bus.

In addition, delay caused by traffic congestion along the bus route is also an example of systematic delay. During peak and non-peak periods, travel time always varies and that is due to time spent in traffic congestion during rush hours of peak periods which are not always accounted for in the bus trip schedule. Traffic congestion can be measured by observing the average speed of a trip. If the average speed is low, we can conclude that there was a traffic congestion at a point in the trip. Therefore, changes in average speed will surely contribute to the variation in bus travel time.

Number of stops observed in the life time of a given trip is strongly correlated to the average travel time [5]. Stops are made by buses only if there are passengers who want to alight or board the bus from the list of scheduled stops. If there are no passengers at a stop, the bus continues its journey, however, if there are passengers, the bus has to stop. These stops can not be predicted, although it can be estimated based on studied pattern. For example, during peak hours, the number of stops observed are always more than that of non-peak hours.

Lastly, weather changes can also affect the travel time of a bus. For example, due to global warming, we might experience sunshine immediately after a rainfall on the same day and it is expected that public buses maintain specific speed for different weather conditions. In this case, there will be variation in travel time for any trip within the two weather conditions.

Although some systematic delays discussed thus far can be measured and used in making accurate travel time prediction, they can not all be captured due to the type of data available.

Table 1 gives a summary of all factors that have been discussed so far and the delay group each belongs to.

Table 1. Factors affecting bus travel time variability

Factor	Delay group
Road accident	Unforeseen delay
Change in average speed	Systematic delay
Number of observed stops	Systematic delay
Driver's experience/behavior	Unforeseen delay
Number of passengers boarding and alighting the bus	Systematic delay
Change in weather condition	Unforeseen/Systematic delay
Road accident	Unforeseen delay
Traffic light at junction traversed	Unforeseen delay
Road workers	Unforeseen delay
Change of driver	Systematic delay
Traffic congestion	Systematic delay

1.1.2 ITS influence on Public Transport

Technological advancement has reflected heavily on the effective operation of public bus transportation system and of notable mention is its enhancement through application of Intelligent Transportation System (ITS). ITS is defined as the use of state of the art technologies to improve transportation system of any kind. With ITS, the system has been made faster, efficient, safer, convenient and understandable to all by providing systems like Advanced Public Transportation System (APTS). APTS as a system consists of different subsystems such as Intelligent Traffic System, Advanced Traveler Information System (ATIS) and Advanced Vehicle Location System (AVLS).

AVLS are used to monitor the current position of buses which is reported by Global Positioning System (GPS) that are installed inside the buses. The information provided can then be used to report any significant delay or further analyzed to provide optimized route options. ATIS is used by transport agencies to broadcast information regarding link arrival time and journey times to commuters through web platforms, mobile applications or sophisticated devices [6]. In particular, commuters are interested in the information relayed by ATIS as they want to know the actual travel times of buses and also the arrival time of the next bus at a particular stop while transport agencies are interested in the information relayed back by the AVLS to provide better service to commuters.

Although ATIS is capable of providing near-accurate trip information in real time, the demand of commuters has changed from getting accurate real-time trip information to been able to get near-accurate information about future trips. That is, given a future date and a particular trip between two or more stops, the system should be able to give near-accurate trip information while taking it consideration different patterns that might occur. In its actual sense, these values are often difficult to predict as it cannot be measured directly but if achieved, it will be a massive boost to the acceptance of public bus transportation in larger cities.

This research area is a popular research field in ITS because researchers are trying to come up with different scientific methods that can be used to improve accuracy of previous research. However, results have shown that more work still needs to be done in order to meet commuters' accuracy demand. Past researches have focused on predicting arrival times of public bus at different stops while some focused on predicting traveling time between stops. In this thesis, the focus is on developing a model for predicting near-accurate link travel time information using an Origin Destination matrix (OD) generated from historical GPS dataset obtained from public buses in the city of Dublin, Ireland. The OD matrix allows us to represent travel time distribution between different combinations of stops.

1.2 Objectives

The main objective of the research described in this master's thesis is to develop and test a model that is capable of providing accurate prediction of link travel times distributed across a given trip in the future. To achieve the main objective, the following objectives had to be met:

- Understand different state-of-the-art methods applied to travel time prediction. The goal here is to identify the most suitable model to use for predicting travel times based on historical GPS data.

- Identify popular data sources that currently exist and determine their usefulness to the research based on factors like ground truth etc.

- Understand and analyze the obtained data to identify values that affect travel times under normal traffic conditions and abnormal conditions. Also, to identify values that could affect the reliability of predicted result.

  Finally, the result of the tested model is summarized and presented in a comprehensive format.

1.3 Scope

Travel time prediction problem is a broad research topic in the field of Intelligent Transportation System. A wide variety of prediction methods have been proposed in literature in order to provide accurate travel time predictions in public transportation system. The proposed methods have been applied to data obtained from public bus, taxis, trains or trams. In this thesis, the research is limited to the following:

First, the work done is applied to a selected bus line of an urban route. That is the scope is limited to a particular transportation type and route in an urban network. However, the concept can be easily extended to different public transport mode like taxi or train.

Secondly, it has been mentioned that there are a lot of factors that leads to the variation of bus travel time like weather, bus dwelling time and traffic signals. In this thesis, the focus is limited to bus scheduled departure time at the origin stop with focus on both the hour and the minute, the distance between stops, speed, and lastly, the day of the week. This is mainly because of the limitation of available dataset.

Finally, the prediction is done on a historical database of sampled GPS dataset offline.

1.4 Contribution

The methodologies discussed in this thesis are mainly built on Neural network and Data analysis. The use of O-D matrix alongside Neural network for link travel time distribution and prediction was investigated. The use of O-D matrix for travel time prediction might be an interesting concept since it presents a new approach towards travel time prediction for public transportation system. With O-D matrix, we can present the travel time distribution across all links in a given trip by considering all possible combinations of links throughout the trip. We believe that the approach discussed can be adopted by transport agencies and also a good starting point for further research.

1.5 Road Map

This thesis is arranged as follows:

Chapter 2 covers state-of-the-art methods that have been used in solving travel time prediction problem with focus on data collected from GPS installed in transit bus in major cities. For each reviewed literature, the test data, method and travel time factors considered are discussed.

Chapter 3 describes the methodology and the neural network model used in solving the travel time prediction problem.

Chapter 4 describes the dataset and how it was collected. Since the dataset is a GPS dataset, the filtering algorithms used in removing outliers in the dataset was also discussed. The chapter also describes the result obtained after analyzing the filtered dataset.

Chapter 5 presents the result of the proposed method after being applied on a test GPS dataset.

Finally, in chapter 6, the recommendations for future works and conclusions of the study is presented.

2 State of the Art

Over the years, different state-of-the-art prediction methods have been introduced and applied to travel time prediction problem. The most popular type of methods have been based on Kalman filter algorithms, artificial neural networks (ANN) and historical data analysis. Some proposed methods have also combined two models together either two artificial neural networks or an ANN model with Kalman filter algorithm. The focus of each proposed model is to increase the accuracy of travel time prediction while putting into consideration factors that often affect travel time. In this chapter several prediction methodologies from previous research studies that were applied to GPS type bus dataset will be reviewed. The prediction method, type of data and the criterion used to achieve prediction accuracy will be discussed for each reviewed literature. The summary of all reviewed literature is shown in table 3. It contains information about the type of method, the description of the dataset used and lastly, the factors considered.

2.1 Historical Database model

In this class of prediction model, prediction is made by studying the pattern of historical travel time of past journeys within the same period. This model assumes that the traffic condition at the selected route is always stable at any given time period. In this section, studies based on historical database are reviewed.

Weiping et al in [7] introduced a novel travel time prediction model referred to as Historical Trajectory based Travel/Arrival Time Prediction (HTTP). The framework is capable of making real-time travel time prediction of next segments of an active bus journey based on large collection of historical trajectories. It was divided into three modules: Bus Status Monitoring module (BSM), Travel Time Prediction (TTP) module and Similar Trajectory Search (STS) module [7]. The prediction made by TTP module is based on similarities that exist in the trajectories collected by the BSM module. The study took advantage of traffic patterns that exist in road segments which when observed and analyzed can be used in making future prediction.

The framework combines two prediction schemes called Passed Segment Scheme and Temporal Feature Scheme to form a single prediction scheme called hybrid prediction scheme that was further divided into two. The passed segment scheme makes prediction by identifying pattern similarities in trajectories observed on previous segments while temporal feature is based on temporal journey features like hour of the day and day of the week. These temporal features are used in order to find similarities in trajectories for travel time prediction.

The proposed framework was applied on a trajectory dataset collected from Taipei City in Taiwan to evaluate the prediction accuracy and efficiency. The result showed that the framework significantly made better prediction than state-of-the-art techniques like TransDB scheme and average travel time scheme.

In [8], *Vanajakshi et al* predicted travel time by studying the historical travel time and computing the average. This method was compared with the actual proposed model which was based on Kalman Filter algorithm. Travel time based on historical average is often used for comparison purposes when a novel model is introduced. Travel time prediction based on historical average was also used in [7], [9].

2.2 Kalman Filter

Kalman Filter (KF) technique have been used in some studies to solve travel time prediction problem. It is a recursive method that consist of set of mathematical equations which implement an optimal predictor-corrector type estimator, that is, it tries to minimize the estimated error after some predefined conditions have been met [10]. With its recursive ability, KF makes an estimate of the current state based on previous estimates referred to as the posteriori. KF ability to make near-accurate estimate from sequence of observed data has made the method popular in different literature relating to travel time prediction and estimation.

In [11], *Yang et al* applied kilman filter and estimation algorithm to travel time prediction. In the study, the focus was on solving travel time prediction problem during peak periods like traffic congestion caused due to special events (e.g concerts, graduation ceremonies and conventions) at a specific route. The traffic flow at Duluth Entertainment and Convention Center (DECC) during a graduation ceremony on April 25 and May 22, 2004, was used as case study.

The prediction model was modeled as follows,

$$x_{k+1} = \Phi_k x_k + w_k$$

Where:

x_k is the travel time to be predicted at *k*

Φ_k is the state transition variable relating x_k to x_{k+1}. The value is obtained through historical data.

w_k is a zero mean Gaussian noise sequence with a covariance value denoted as Q_k.

The travel time prediction was done every three minute interval and the data is obtained every ten seconds through GPS device installed on three test vehicles running the specified routes back and forth till the traffic congestion elapsed (45 Minutes in this case). That is, the KF model was applied recursively every three minutes on the observed

data to predict the next travel time.
The observation equation associated to the value of x_k is given as:

$$z_k = x_k + v_k$$

Where:

z_k is the average travel times reported by the three test vehicles

v_k is the measurement noise represented as a zero mean Gaussian sequence with a co-variance value R_k.

The performance of the model was analyzed and it was reported that the prediction accuracy depends on how long the traffic congestion lasted, that is if the traffic congestion duration is short then average error will be large. Also, if there is a sudden change in the actual travel time, there will be a huge difference between the predicted travel time and the observed travel time.
The model prediction error in MARE was initially reported as 17.61% but after applying a two-point data interpolation technique, it was observed that the MARE can be reduced to 4.40%. The two-point data interpolation approach involves introducing two artificial data points by calculating the average of any two consecutive data points. This makes the prediction less likely to be affected by fluctuation in observed travel time [11].

The author also studied the effect of prediction time interval and noise variance on the prediction error. The study showed that increasing the prediction time interval will increase the prediction error while increasing R_k and decreasing Q_k will reduce the value of MARE. However, the best MARE value(4.40%) was obtained after applying the data interpolation technique.

Although the prediction error of the model is said to be acceptable by the Minnesota traffic engineer, the model is still affected by the limitation of KF techniques. Since KF estimation at each time step is a linear combination of previous measurements, therefore, it can only be used in linear system and most real world systems are non-linear [12]. In addition, only travel time parameter were considered in the study, that is factors that might affect travel time such as weather, speed and time of the day were not considered.

In [8], *Vanajakshi et al* used a Kalman Filter algorithm to predict travel time under heterogeneous traffic condition on an urban route in the city of Chennai, India. The approach involves dividing the entire route into subsections and predicting the travel time of each subsection.
Three buses were used in this research:probe vehicle 1 (PV1), probe vehicle 2 (PV2) and test vehicle (TV). The probe vehicles were used to gather data fed into the KF algorithm used in predicting the travel time of the test vehicle as it travels along the subsections in

the route. That is, when the test vehicle is at *k* subsection, the travel time for the next subsection represented as *k+1* is predicted. The predicted TT is then compared to the actual travel time observed by TV at *k+1*.

The travel time distribution across the different segment is given as:

$$x(k+1) = a(k)x(k) + w(k)$$

where,

$x(k)$ is the observed travel time while travel subsection *k*.

$a(k)$ value that relates travel time observed at *k* and *k+1*. Obtained using PV1.

$w(k)$ zero mean Gaussian sequence.

We can see that the equation is similar to the prediction model used in [11]. In addition, the same measurement model used by *Yang et al* [11] was also used in the study.

The algorithm prediction accuracy was computed using absolute prediction error (APE). The result showed that for the two subsections considered while making prediction for a single day, the APE was 13.75% and 17.09% respectively. The prediction of the KF algorithm was also compared against the TT average observed by both PV1 and PV2. The result of the comparison showed the proposed KF algorithm performed better than the TT average.

Although, the algorithm discussed in [8] can be used in APTS applications built specifically for heterogeneous traffic conditions, the model still suffers from the limitations of KF models.

Huifeng et. al in [13] used Kalman Filter to solve dynamic travel time prediction problem. Dynamic travel time prediction involves updating the travel time predicted at segment *k+1* using both observed and predicted travel time at segment *k*. With this approach, changes that might have occurred while on the route like varying speed, accidents etc. would be considered in the prediction. In this study, the selected route was divided into 13 segments and data was collected for different time periods (considering both peak and non-peak hours). The prediction of the model was compared to the observed travel time and the result showed that the model made near accurate predictions that is sufficient enough.

2.3 Artificial Neural Networks

Neural Network is a concept that was inspired by the operation of the brain and it as been successful in solving problems related to pattern recognition. A detailed description on how neural network works is presented in chapter 4 of this thesis.

NN models used in solving travel time prediction problem often differ by the architecture, input-output combination and the training algorithm used. In this section, different studies that focused on using NN solve travel time prediction are reviewed.

Zegeye et al in [9] developed a Multi-Layer Perceptron (MLP) neural network model which is capable of predicting accurate travel time information. The study was motivated by the need to provide travel time information to public bus users using only travel time data collected through GPS. The model is based on studying both the historical and real time arrival and departure time patterns at different stops in the selected route. Unlike KF methods discussed in [11], [8] and [13], NN model used in the study is able to capture non-linear relationship between travel times. The NN model has one hidden layer with 15 neurons and was trained using Levenberg-Marquardt algorithm.

The prediction algorithm was given as:

$$TT_{cj}^{k} = TT_{ij}^{k} - TT_{ic}^{k}$$

Where:

TT_{cj}^{k} is the predicted travel time from stop c to stop j.

TT_{ic}^{k} is the observed travel time from stop i to a point c.

TT_{ij}^{k} is the neural network predicted travel time from a stop i to j.

After analysis of the dataset used in the study, it was observed that the time of the day and the day type often contribute to travel time variability. Due to these factors, the study considered 4 variables which are, the time of the day, the id of the origin and destination stops and the travel time as input to the proposed model.

The performance of the proposed model in [9] was analyzed and it showed the Mean Absolute Percentage Error(MAPE) increases when the number of stops between the origin and destination stop is too large or too small. That is the model performs poorly when the travel time is less than 5 minutes or greater than 50 minutes. However, if the number of stop between the two stops is up to 5 stops where the travel time is between 20 and 50 minutes, then the model makes better prediction. The prediction accuracy and the robustness of the model was compared to an historical average model which involved finding the average of historical observed travel times. The result showed that 70% of the time, the model makes better prediction.

Yangie et al in [1] applied a different NN architecture called Long Short-Term Memory (LSTM) to solving travel time prediction problem. The problem was transformed into a time series and the LSTM model was used in making the travel time prediction. The approach is first of its kind and the reason for using LSTM is because the model is able to automatically store historical sequence that are useful for making accurate travel time prediction. LSTM architecture is regarded as a specific type of recurrent neural network, where hidden layer of the NN in different time sequence are connected to each other. Each hidden layer has an LSTM cell with different gate type, namely: input gate, forget gate and output gate. Figure 1 shows the different gates of an LSTM architecture. The main purpose the gates is to control the flow of information within the LSTM cell and NN.

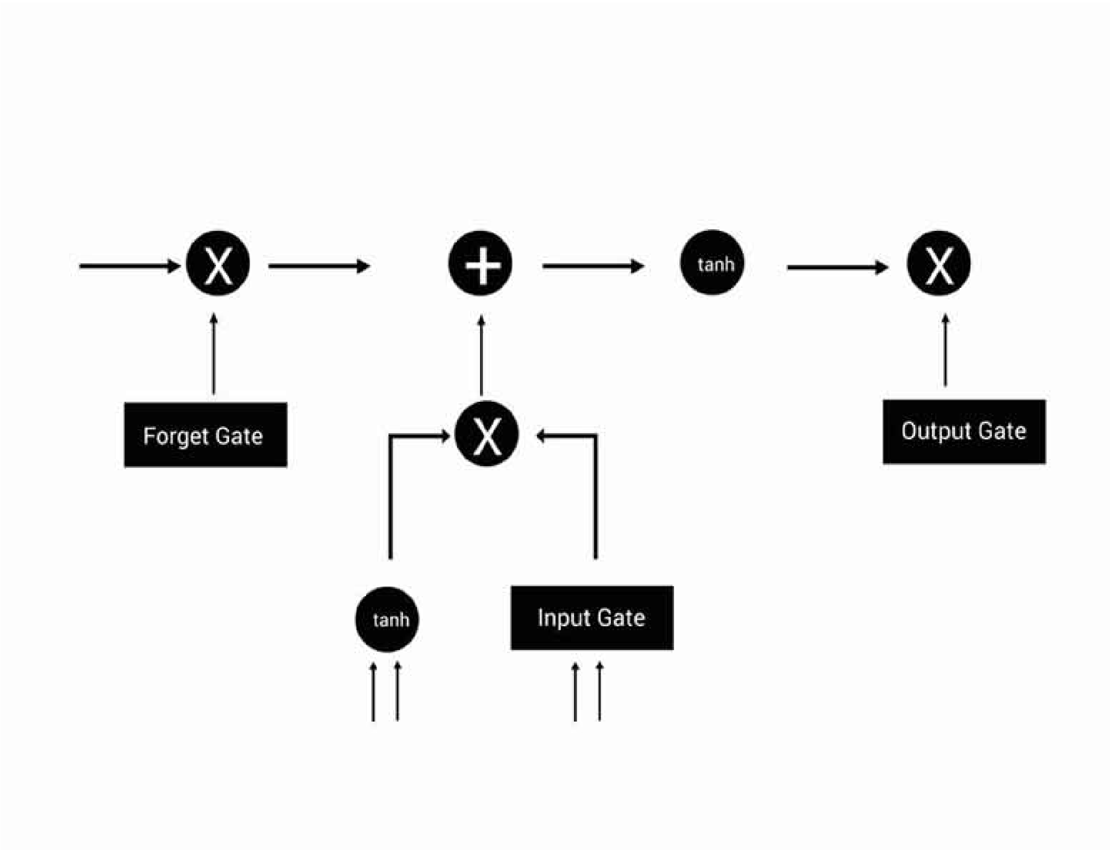


Figure 1. LSTM gates [1]

The dimension of the input layer and output layer of the LSTM model was 1, which means that factors causing variation in travel times were not considered. The number of hidden layer used in the study varies with different links. However, for any given number of links, the highest number of hidden layers that can be used is 5. The performance of the proposed model was tested by making predictions over 4 time periods in the future labeled as 1-step ahead, 2-step, 3-step and 4-step. The study showed that as the number of steps increase the prediction error also increases. However, the median of the MRE was reported as 7.0% when the model was applied on the test data.

Furthermore, in [14], *Johar et al* applied neural network model to solve travel time prediction problem on a selected route in the city of Delhi and achieved a sufficient level of accuracy. In the study, travel times were observed for both peak and none peak periods. The observed data showed that the number of passengers boarding and dropping off a bus, the number of stops observed by a bus during a given trip and the dwelling times at each stop all contribute to the variation of bus travel times. These three parameters were used as input to the NN model proposed. The NN architecture is a simple MLP structure with 1 hidden layer having 9 neurons, 1 input layer with 3 neurons and an output layer with 1 neuron.*Johar et al* made use of Levenberg-Marquardt algorithm with Bayesian regularization as the training function.

The accuracy of the proposed model was computed by comparing the predicted travel time and the actual travel time using Chi-test. The result of the comparison showed that the proposed model is capable of making predictions with sufficient accuracy.

In [15], *Wichai et al* took a different approach by using deep neural network model on travel time prediction problem. The study was motivated by the need to improve the prediction accuracy when predicting travel times of trips with varying long distance. The architecture is able to capture more complex relationship and features that could not have been captured with models discussed in [9], [1] and [14] . Seven parameters that affect travel times were identified in the study, namely: origin and destination location, the distance between the two points, the instantaneous speed, average speed, hour of the day and lastly, the day of the week.

The NN architecture used in the study has 11 neurons at the input layer, 4 hidden layers with 7 neurons each and a single output layer. The model was trained using adaptive gradient optimizer (adaGrad).

The prediction model proposed was compared with an existing solution which is referred to as Ordinary Least Square (OLS) method. Analysis of the prediction result showed that the model makes more accurate prediction and is more feasible than the existing solution.

2.4 Hybrid Methods

Hybrid method involves combining two or more methods from historical database model, Kalman filter or machine learning to solving travel time prediction problem. The motivation of such combination is to improve the accuracy achieved when using a single method by employing the advantages of the combined methods.

Jianying et al in [16] proposed a hybrid type method which is the combination of Kalman Filter and Elman Neural Network which is a type of Recurrent Neural Network model to solving travel times prediction problems. The model was applied on a dataset collected over five days period on a link from Wenhui Bridge to Mingguang Bridge, Beijing China.

The model proposed in [16] made use of two type of data, the historical dataset and the real time dataset. Historical dataset is fed to the Elman Neural Network and the predicted output plus the observed real time data is fed to the Kalman filter model. The neural network consist of four layers namely, input layer, hidden layer, context layer and output layer. The Kalman filter model applied at the later stage outputs the predicted result of the hybrid model. The prediction accuracy of the model was compared to the prediction accuracy observed when each method are applied separately. The result of the comparison showed that the proposed hybrid model makes better prediction than the separate methods.

Furthermore, in [17], *Zhihao et al* developed a hybrid neural network model for travel time prediction. The model was created by fusing a 2-dimensional convolution neural network (CNN) and long-short term memory (LSTM) together. The idea was motivated by the dominant performance of both CNN and LSTM [17]. In the study, CNN is responsible for identifying the spatial features of traffic conditions from an image input with spatiotemporal characteristics while LSTM identifies the correlation of the travel time series problem. The features identified by CNN is then used as input to the linear regression layer. The proposed model was trained using RMSprop optimizer.

The proposed model was used to predict the travel time on a 35km urban expressway in Beijing, China. The prediction accuracy was then compared to other classical prediction algorithms like instantaneous travel time, historical average and naive K-Nearest-Neighbor (KNN). The result showed that the model outperformed the classical algorithms during peak and none-peak periods. However, the model made most-accurate predictions when making short-term travel time prediction. Although the dataset used in this study was obtained through probe data, it will be interesting to analyze the result of the model when applied on a GPS data.

Avigdor et al in [18], introduced a novel hybrid model which involved combining Machine Learning algorithm with methods from Queuing Theory. Given a scheduled route, the proposed model is capable of predicting the travel time between an origin and destination pair. In the proposed model, journey segments are represented as network queues, while buses are interpreted as clients that go through the queue. Snapshot principle from Queuing theory is then applied on the modeled data to make predictions.

In the study, the snapshot principle follows the logic that a bus passing through a particular segment will observe same travel time observed by another bus that just passed through that same segment. The prediction accuracy of the snapshot principle was tested and the result showed that it is able to capture any delay in the system. However, it was also reported to suffer from outliers and in order to fix the outliers, machine learning technique called regression tree which learns from example observed from an historical and real time dataset is applied. The method acted like a boost to the non-learning snapshot principle [18]. The prediction accuracy of the two combined model is report to be sufficient enough in making accurate travel time prediction.

2.5 Summary

In this chapter, state-of-the-art travel time prediction models were reviewed with focus on models applied on GPS dataset obtained from public transit vehicles. The literature reviewed have shown that most travel time prediction model often rely on historical travel time patterns by taking into consideration factors that can lead to variation in travel time such as number of people boarding and leaving the bus at a given stop, the distance between the origin and destination stop, dwelling time at stop, speed, time of the day, accidents and lastly, weather condition. Although most of the review models tried to consider some of the highlighted factors, unfortunately, the type of data available often limits factors that can be considered when making travel time prediction. Table 3 shows the different factors considered by each model reviewed in this chapter with the description of the dataset used as test bed.

Author	**Approach**	**Variables**	**Data**
Weiping et al [7]	Historical and Real time trajectories analysis	Day of the week, time of the day, speed and travel time	1 year bus GPS dataset, Tapei town
Vanajakshi et al [8]	Historical average	travel time	One month dataset, Chennai city
Yang et al [11]	Kalman filter	travel time	45 minutes peak hour dataset
Vanajakshi et al [8]	Kalman filter	travel time	One month dataset, Chennai city
Heifang et al [13]	Kalman filter	travel time and varying speed	One day GPS dataset, Beijing, China
Zegeye et al [9]	NN, MLP Levenberg-Marquardt	Origin and destination stop, time of the day and travel time	6 months GPS dataset, Macae, Brazil
Yangie et al [1]	NN, LSTM	travel time	One year GPS dataset, high M25, London
Johar et al [14]	NN, DNN Levenberg-Marquardt	Number of passengers leaving and boarding, dwelling time and average non stop trip time	6 months GPS dataset, Macae, Brazil
Wichai et al [15]	NN, MLP adaGrad	Origin and destination stop, time of the day, distance, average speed instantaneous speed and day of the week	1 month GPS dataset, BMTA-8, Bangkok

Jainying et al [16]	Kalman filter and Elman NN	travel time	5 days GPS dataset, Beijing, China
Zhihao et al [17]	CNN and LSTM	travel time and speed	45 days GPS dataset, Route Ring 2, Beijing, China
Avigdor et al [18]	Queueing theory and Regression tree	travel time	1 month GPS dataset, Route 46A, Dublin

Table 3. Literature Review Summary

3 Methodology

In this thesis, a neural network model is developed to predict link travel times based on an OD matrix generated through GPS data obtained from a public bus transit system of the city of Dublin. The method is divided into two parts:

1. OD matrix Extraction : This part involves transforming the GPS dataset obtained into an Origin Destination matrix. An OD matrix is a matrix which has origin stops represented as rows and destination stops represented as columns. In previous research like in [19], OD matrix was used to present people's trip in a given location and also, in [20] the concept of OD matrix was applied to a ferry dataset to understand the passenger trip distribution. However, in this research, the approach is different. The matrix is used to present the distribution of travel time across links in a given journey. That is, the values in the cells will represent the travel time between any two stops.

2. Neural Network model : This part involves identifying independent variables that affect travel time values and building a model that is capable of making near accurate travel time predictions based on the independent variables. The input of the developed model is the OD matrix generated from the GPS dataset and the values of all identified independent variables.

   Figure 2 and Figure 7 gives a step by step break down of the prediction approach used in this research. The GPS dataset is first analyzed, then transformed into an OD matrix and fed into the developed NN model for training. After training, prediction is made using the test dataset. Finally, the result of the prediction is analyzed by comparing the value predicted with the ground truth travel time.

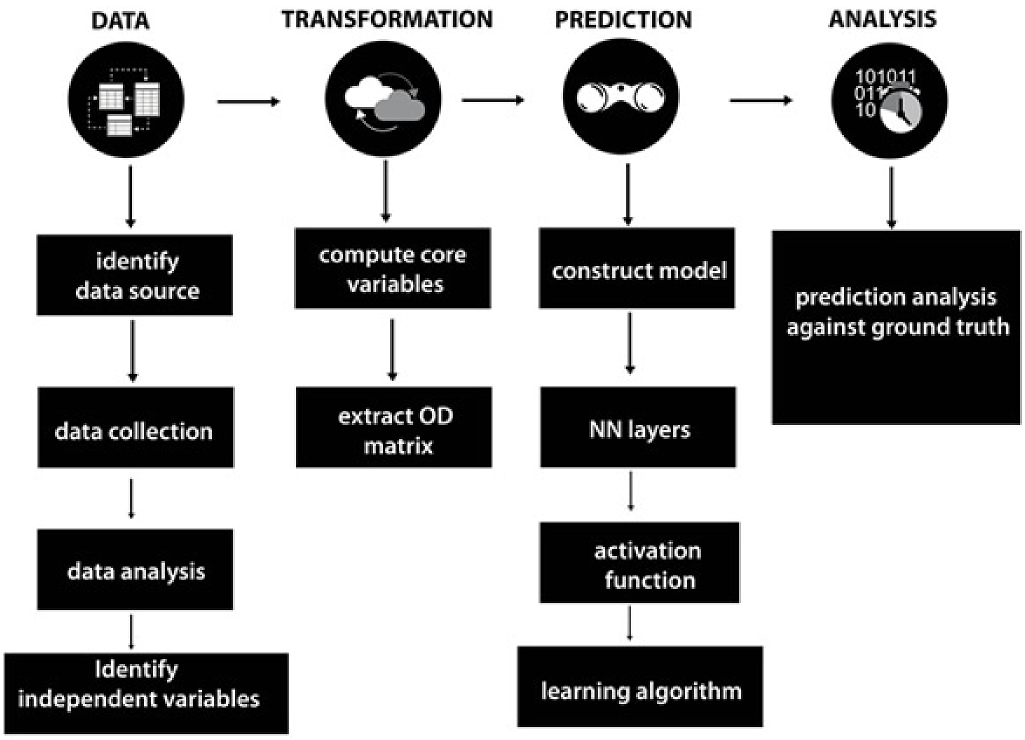


Figure 2. Methodology

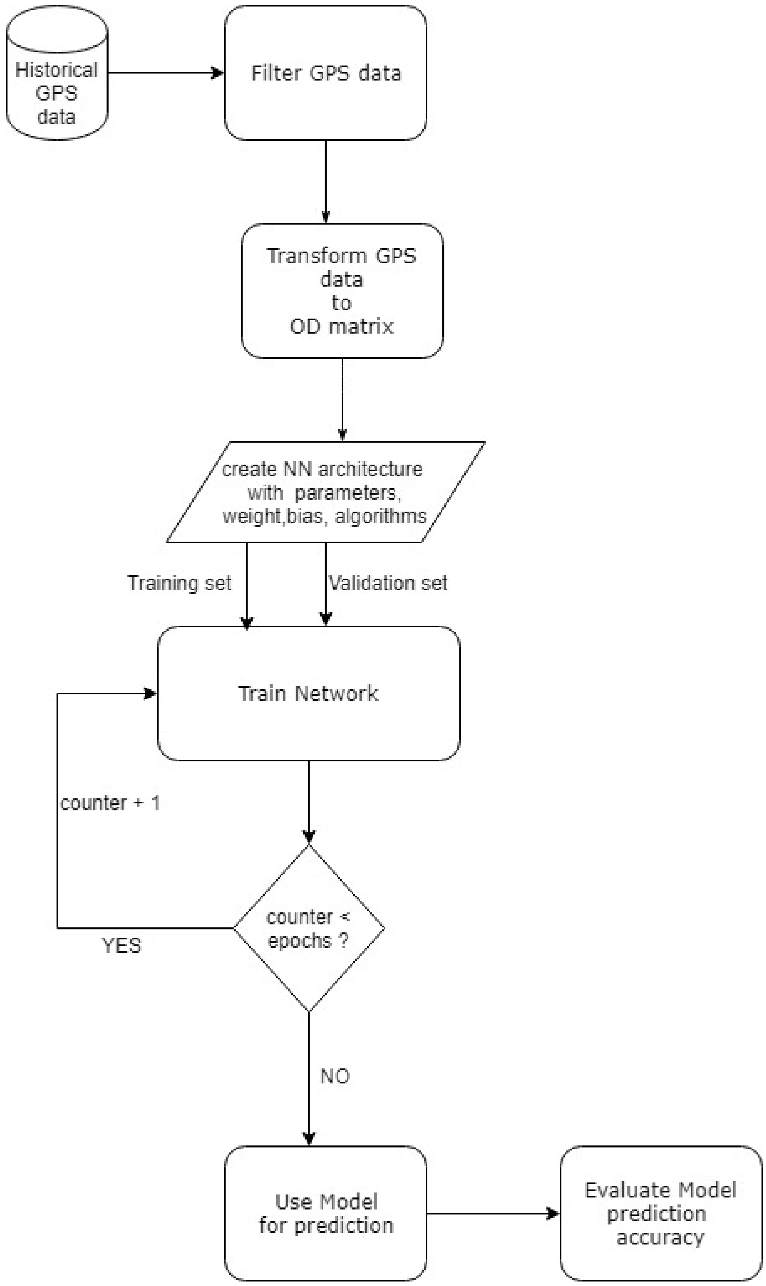


Figure 3. System Flowchart

3.1 OD Matrix Extraction

In public transportation system, an O-D matrix is used to present trip distribution in a given geographical area for effective transportation planning. The concept is adopted in this research to present the travel time for all possible origin and destination pairs along the same route for a specific bus line, in this case bus line 46A.

With this approach, it is easier to understand the travel time distribution for any origin and destination pair along a given link.

The values in the OD cell are estimated travel time between a given origin stop and a destination stop pair. This is computed using:

$$TT^J_{o,d} = T^J_{Do} - T^J_{Ad} \tag{1}$$

Where:

$TT^J_{o,d}$ is the travel time in minute between any origin stop and destination stop pair.

T^J_{Do} is the departure time at any given origin stop.

T^J_{Ad} is the arrival time at any given destination stop.

Table 4. Origin Destination Matrix

	Destination	A	B	C	.	.	Z
Origin							
A		0	$TT_{a,b}$	$TT_{a,c}$	.	.	$TT_{a,z}$
B		0	0	$TT_{b,c}$	.	.	$TT_{b,z}$
C		0	0	0	.		$TT_{c,z}$
.		.	.	.	.	.	.
.		.	.	.	.	.	.
Z		0	0	0	.	.	0

Table 4 describes the OD matrix table and how it was constructed for each journey. From Table 4, we can see that the O-D matrix is an *n x n* matrix that is used to represent a trip graph where n represents the vertices, in this case, all observed stops along the trip. The weights of edges between two vertices is the travel time between the two stops computed using equation 1. That is, if a stop is reachable from another stop, then the travel time between the stops is placed at the *ith* row and *jth* column, if it is not, then 0 is placed. The label of the vertices are labeled in the order in which the stops are transversed. The diagonal of the matrix from the first vertice represent the distributed link travel time as we transverse from the origin to the final destination. The trip total travel time can then be calculated by finding the summation of the distributed link travel times.

Along the route 46a, there are 60 stops from Dun Laoghaire To Phoenix Park, with OD matrix we are able to create 1770 combinations of stop pairs. However, in its real sense, not all stops will be observed because the bus did not make a stop at that bus stop.

3.2 Neural Network

Neural Network is a concept that was inspired by the operation of the brain and so far, it has been successful in solving prediction and estimation problems. The interest in NN as grown over the years due to its ability to find complex non-linear relationships in any given input. It has been successfully applied to different problem domain in diverse areas, ranging from finance, medicine, engineering, geology and physics [21].

3.2.1 Neural Network Architecture

There exist different architectures of neural network, and they are differentiated from each other by how the neuron are arranged in relation to other neurons in the network. A typical neural network is divided into three layers and each layer performs specific task:

- **Input Layer:** The input layer interacts with an external source, accepts data in form of signal or features. These features are then normalized to achieve better numerical precision when a mathematical model is applied at the hidden layer [22]. The number of neurons at the input layer is equivalent to the number of features to be considered in the input dataset, in this case, the number of independent variables that affect travel time.

- **Hidden Layer:** This layer consist of neurons that accept input from previous layer, and pass the output to the next layer in the network. These layer performs most of the processing in the network by extracting correlating patterns from the features passed in by the input layer [22]. The number of neurons at the hidden layer and number of hidden layers corresponds to the complexity of the problem.

- **Output Layer:** The output layer consist of neurons that are responsible for presenting the final result of the network from the methods applied in the previous layer. The number of neurons in the output layer corresponds to the output values of the problem that is being analyzed.

The simplest neural network architecture is referred to as perceptron which can be seen in Figure 4. It has two layers, the input layer with n input neurons and the output layer with one neuron. The input neurons read in the features into the network and the neuron in the output layer applies an activation function on the inputs. The output neuron c performs the simplest output function on the inputs by multiplying the values with randomly chosen weights and adds a bias.

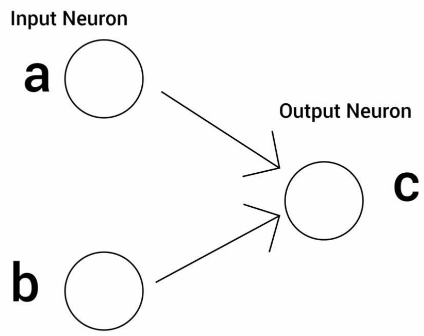


Figure 4. Perceptron

Mathematically:

After the dataset is read into the perceptron network in the form (x_1, y_1) where y_1 is the expected output relating to input x_1. Node *C* performs a simple computation using *equation 2* and *equation 3*.

$$Z = W^T x + \theta \tag{2}$$

Where,

W is the weight represented as a 1 by 2 matrix $[w_1, w_2]$ in this case and W^T it transpose.

x is the input features read into the perceptron represented as a vector $\begin{bmatrix} a \\ b \end{bmatrix}$

θ is the bias for each nodes in the output layer which is also a vector $\begin{bmatrix} \theta \end{bmatrix}$

The final computation done by output node c involves applying an activation function on output Z to get the prediction $\overline{y}$.

$$\overline{y} = \sigma(Z) \tag{3}$$

Perceptrons are very limited in what they can represent thus they are often used for representing linearly separable functions. When the relationship between the input data and the output becomes complex, other neural network architectures with more layers are used. Example of such architectures are Multi Layer Perceptron (MLP), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) neural network. These types of architectures are best at identifying patterns and trends in data for example in pattern recognition problems and time series problems.
Figure 5 is an example of an MLP Neural network, and it has n neurons at the input layer, h neurons at the hidden layer and lastly, two neurons at the output layer.

In MLP, the weight for each input is initialized and the bias is added as discussed above. The result of the computation at each layer is passed forward to the corresponding layer till its get to the output layer. The result at the output layer is compared with the target and if the result differs from the target with a huge margin, the error is propagated back to previous layers in the network to adjust the previously used weights and bias. The backward propagation algorithm used in MLP tries to find the minimum and maximum of a function by iterating over the direction of the negative of the slope of the function to be minimized or maximized [21]. The error can be computed using different methods like Root Mean Square Error (RMSE), Mean Square Error (MSE) or Mean Absolute Error (MAE) or Mean Absolute Percentage Error (MAPE).

Using Figure 5 as case study, the mathematical computation at each layer becomes:

$$Z^{[1]} = {W^{[1]}}^T x + b = \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} \\ w_{1,2} & w_{2,2} & w_{3,2} \\ w_{1,3} & w_{2,3} & w_{3,3} \\ w_{1,4} & w_{2,4} & w_{3,4} \\ w_{1,5} & w_{2,5} & w_{3,5} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \tag{4}$$

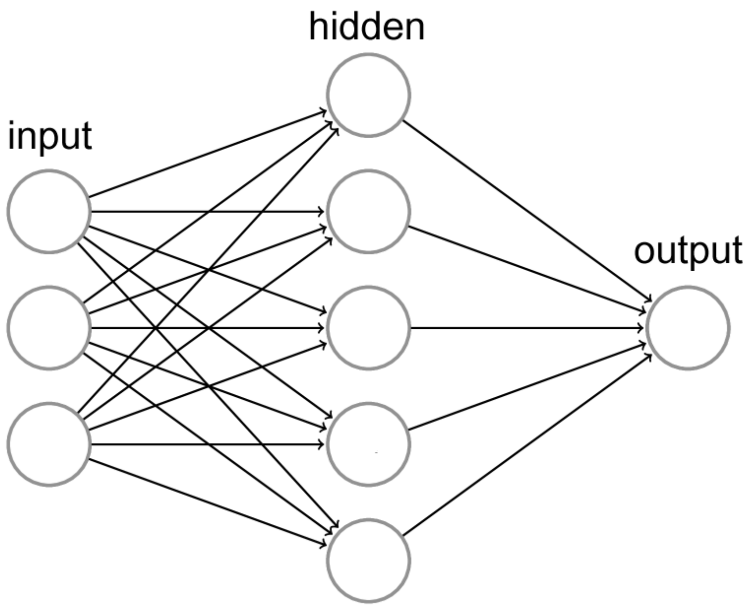


Figure 5. Multi Layer Perceptron

Applying the activation function which can be Relu, Sigmoid or TanH. In this case, a Sigmoid function is used as example:

$$a^{[1]} = \sigma(Z^{[1]}) = \sigma(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}) \tag{5}$$

Where σ used to introduce non-linearity and output boundaries are (0,1)

$$\sigma = \frac{1}{1 + e^{-x}} \tag{6}$$

The result is passed as input to the next layer and the same computation is applied has shown in equation 4.

$$Z^{[2]} = {W^{[2]}}^T a + b = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} + \begin{bmatrix} b^{[2]} \end{bmatrix} \tag{7}$$

Again the activation function is applied to obtain the prediction:

$$\overline{y} = \sigma(Z^{[2]}) \tag{8}$$

The backward propagation algorithm computes the loss function which defines the error between the target and the predicted output of the neural network for all sets of input-output pairs.

$$E = 1/N \sum_{i=1}^{N} (y_i - \overline{y}_i)^2 \quad MSE \tag{9}$$

Where $\overline{y}$ is the predicted result and y is the target.

and the algorithm tries to minimize error with respect to the weights of the neural network:

$$\frac{\partial E}{\partial W_{ij}^k} \tag{10}$$

Using the gradient descent strategy, the result of the loss function is backward propagated to the NN for the purpose of adjusting the weights and bias using chain and product rule of differential calculus:

$$\frac{\partial E(n)}{\partial W_{ij}(n)} = \frac{\partial E(n)}{\partial z_j^k(n)} \frac{\partial z_j^k(n)}{\partial w_{ij}^k} \tag{11}$$

Where a_j^k is the result before the activation function is added to generate the predicted output.

The partial derivative of the weight is defined as the product of the error at a node in layer k and a node in layer $k-1$ [23]. Represented mathematically as [23]

$$\frac{\partial E(n)}{\partial W_{ij}(n)} = \partial_j^k * o_j^{k-1} \quad (12)$$

The derivatives of the activation function as we backward propagate into the network is also derived. However, the computation of the derivation depends on the type of activation function that is used.

In summary, the backward propagation is broken down into the following four steps:

1. Read in values and targets
2. Feed forward computation into the network
3. Get prediction y and compute error $y - y_1$
4. Backward propagates the error into the network to update weight and bias according to input output deviation.

3.2.2 Model

MLP neural network architecture was chosen as the model to use in this study. This was done because it is the most popular architecture and also, it is capable of approximating any function given that there are enough neurons in the hidden layer [24]. The MLP network used is made of three layers; input layer, hidden layer and output layer. In order to achieve better prediction result, the number of hidden layers used was increased to two.

Although there have been different research that has adopted the use of Neural Network for predicting travel time, their input-output combination differs. In this thesis, a unique input-output combination was also used based on the result of the data analysis and state-of-the-art models reviewed in previous section. Six input variables were used in combination with one output variable which is the predicted travel time. The first two input variables X_1 and X_2 is the hour and minute the bus is scheduled to start from the origin bus stop. These two variables were selected to maintain schedule adherence because scheduled departure time also affects the total travel time observed. In addition, non-peak hours with minutes closer to peak-hours always behave exactly like trips started during peak hours.

The third (X_3) and fourth (X_4) variables are the origin and destination stop that the travel time needs to be predicted. The distance between the origin and destination stop is the fifth input variable tagged X_5. The distance between two stops affects travel time because for short distance, it is expected that the travel time observed will be different from that of long distance. X_6 is the average speed and as discussed in chapter 1, changes in average speed often reflects on the travel time observed.

Other variables that affect travel time like weather, traffic light and bus dwelling times at bus stops are not considered because of the limitation of the dataset. However, it is expected that the model will capture the effect of dwelling times at different stops on the total travel time.

The dataset used in this study was divided into two sets namely: the training dataset and test dataset. In order to avoid data over-fitting, the training dataset was further divided into two sets namely: training set and validation set. The training set is used during the training process for model development to capture the relationship between the input and output by adjusting the weight and bias values [25]. In contrast, the test set is used to test the ability of the model that is to know for sure if the model is making accurate predictions and to also compute the difference between the predicted and actual value. The test set is used only after the model is already trained. The validation set is used to fine tune the parameters of the model during the training process.

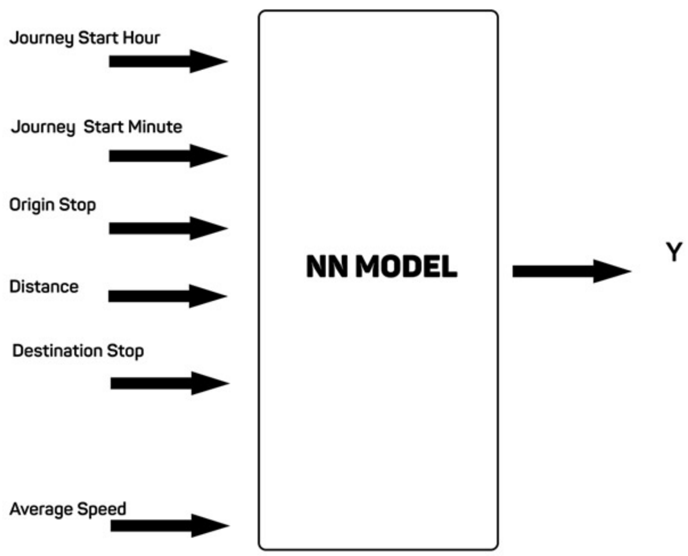


Figure 6. Neural Network Model

Reviewed literature have shown that there is no general procedure used to divide the dataset into training, test and validation sets. In this thesis, seventy percent of the dataset

(70%) was used as the training set while the remaining thirty percent (30%) of dataset was used as the test set. This corresponds to using first three weeks of the dataset for the training set and the last week as the test set. Twenty percent (20%) of the training set was used as the validation set.

An NN model needs to be trained with a training algorithm in order to attain the desired output according to the training patterns [26]. This algorithm is repeatedly applied to update the network and only stopped when a predefined criterion is met. The choice of criterion always lie between the maximum number of epochs and the minimum error gradient. The type of training function chosen often depends on the type of problem been solved either a classification or prediction problem, the size of the dataset i.e the number of available data points and the number of weights and biases in the neural network. With prediction problems, some of the most popular training algorithms that have been used are Levenberg-Marquardt algorithm, Bayesian Regularization , Steepest Descent algorithm and adaGrad algorithm.

In this research, the selected training algorithm is adaGrad. The algorithm is referred to as an adaptive learning rate algorithm that tries to improve the learning rate of features in the dataset by performing large updates for infrequent parameters and smaller update for frequent parameters [27]. Adagrad modifies the general learning rate *n* at every unique time step *t* for every parameter θ_i based on the past gradients that have been computed for θ_i [28]. The update is given as:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{n}{\sqrt{G_{t,ii} + \epsilon}} * g_{t,i} \tag{13}$$

Where:

$\theta_{t+1,i}$ = parameter at the next time step.

n = learning rate

$\theta_{t,i}$ = parameter at a previous time step

$G_{t,ii}$ = is a diagonal matrix where each diagonal value *ii* is the sum of the squares of the gradients

$g_{t,i}$ = previous gradient , given as:

$$g_{t,i} = \bigtriangledown_{\theta t} J(\theta_{t,i}) \tag{14}$$

Duchi et al showed in [29] that adaGrad performs best with sparse dataset, thus making making it well-suited for the neural network model used in this research.

3.3 Summary

In this chapter, the neural network model used for this research was introduced and extensively explained. Although neural network has been described as a black box over the years, implementing it is not that difficult as one might have thought. The availability of different NN libraries has made the construction of NN easier for researchers and thus, has allowed researchers to focus more on data collection and analysis in order to find the correlation between dependent and independent variables. However, it is still important for one to fully grasp the concept so as to avoid issues related to over-fitting and under-fitting.

4 Data

The dataset used in this study is an open dataset made available by Dublin City Council [30]. It consists of GPS data collected by Dublin City Traffic Control System from different buses plying different routes in the city. It corresponds to approximately one month of data from November 6, 2012, to November 30, 2012. The dataset is popular among researchers who have worked with problems related to estimation or prediction of either bus arrival time or bus travel time.

4.1 Data Description

To build the dataset, each active bus sends data to the traffic control system every twenty seconds (20 secs) throughout the lifetime of a specific journey. Figure 7 describes the data collection scheme in a nutshell. It shows that each bus has a GPS installed inside it and it is interfaced to Internet. The GPS tracks the location of the bus in latitude-longitude pairs, the current time and date. The GPS data and some other relevant information are then relayed through the interfaced Internet connection to the Traffic Control System. The data sent are :

1. The time-stamp of event, that is the time the data was observed.
2. Current location of the bus in Latitude and Longitude.
3. An identifier for the bus, termed as the vehicle id.
4. An identifier for the journey, termed as the journey id. The journey id specifies the direction of the journey, either it is south bound or north bound.
5. The vehicle journey id. This identifier is used to differentiate between journeys of the same bus.
6. The journey pattern. This specifies the sequence of bus stop for journey.
7. Bus stop id. This identifier defines the bus stop closest to a bus. That is, every data sent to the traffic control system will have a bus stop id even though the bus is not at the stop.
8. A binary identifier which specifies if a bus is at a particular stop or not. This identifier is called at_stop and it can only have a value of either 0 or 1.

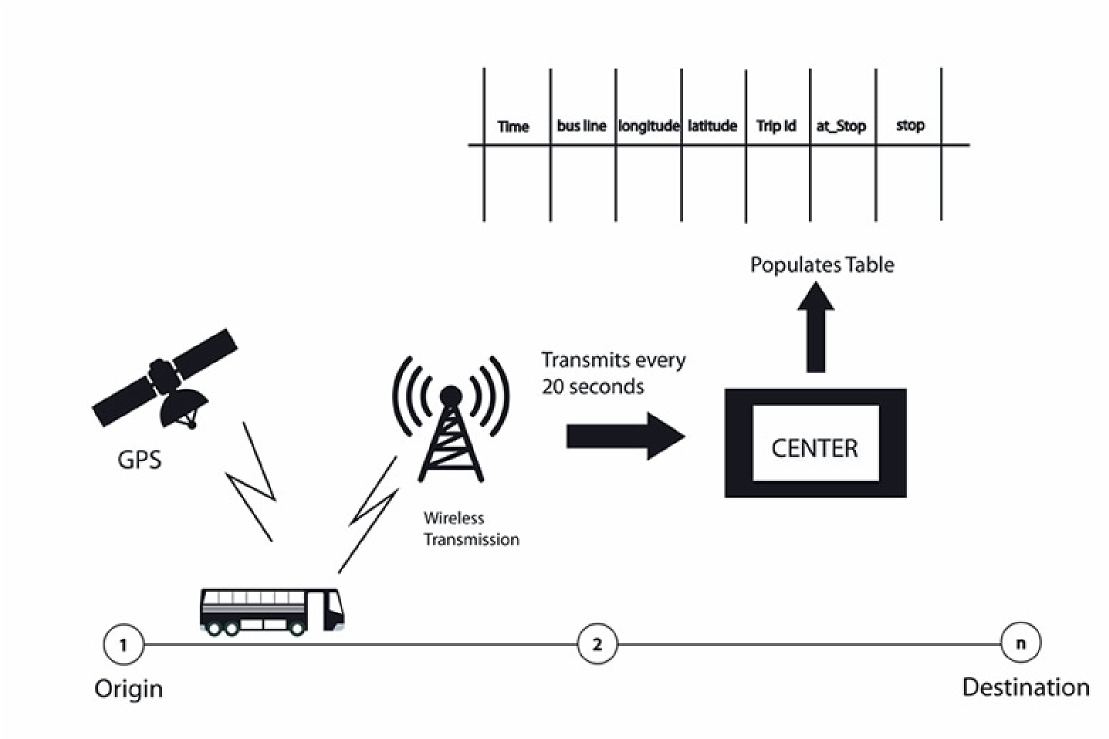


Figure 7. Data Collection Scheme

After careful analysis, bus line 46A was chosen for the research described in this thesis. The route was chosen because it has more lengthy journey pattern when compared to other routes in the dataset. Furthermore, the route passes through Dublin city center which means it is vulnerable to peak hour traffic and travel time variation. Figure 8 shows the route of bus line 46A, where the blue markers correspond to bus stops that can be observed along the route.

4.2 Data Reduction

Since it is a GPS dataset, it means the data is susceptible to errors and noise. To prepare the data for the model, all errors and noise had to be cleared. During data cleaning, each vehicle journey was considered separately and some filters were applied to prune out bad journeys. The following filters were applied to the dataset.

1. The dataset for line 46A comes with four journey patterns 046A0001, 046A1001, 401001 and 400001. While analyzing the dataset , only 046A0001 and 046A1001 were frequently visited, and they were also lengthy (covers up to 19km) . Based on this, journeys belonging to 401001 and 400001 were eliminated from the dataset.

Table 5. Route 46A details

Details	Value
Length	19km
Bus stops	59
Journey Time (Excluding Peak Time)	60 minutes
Journey Id	2

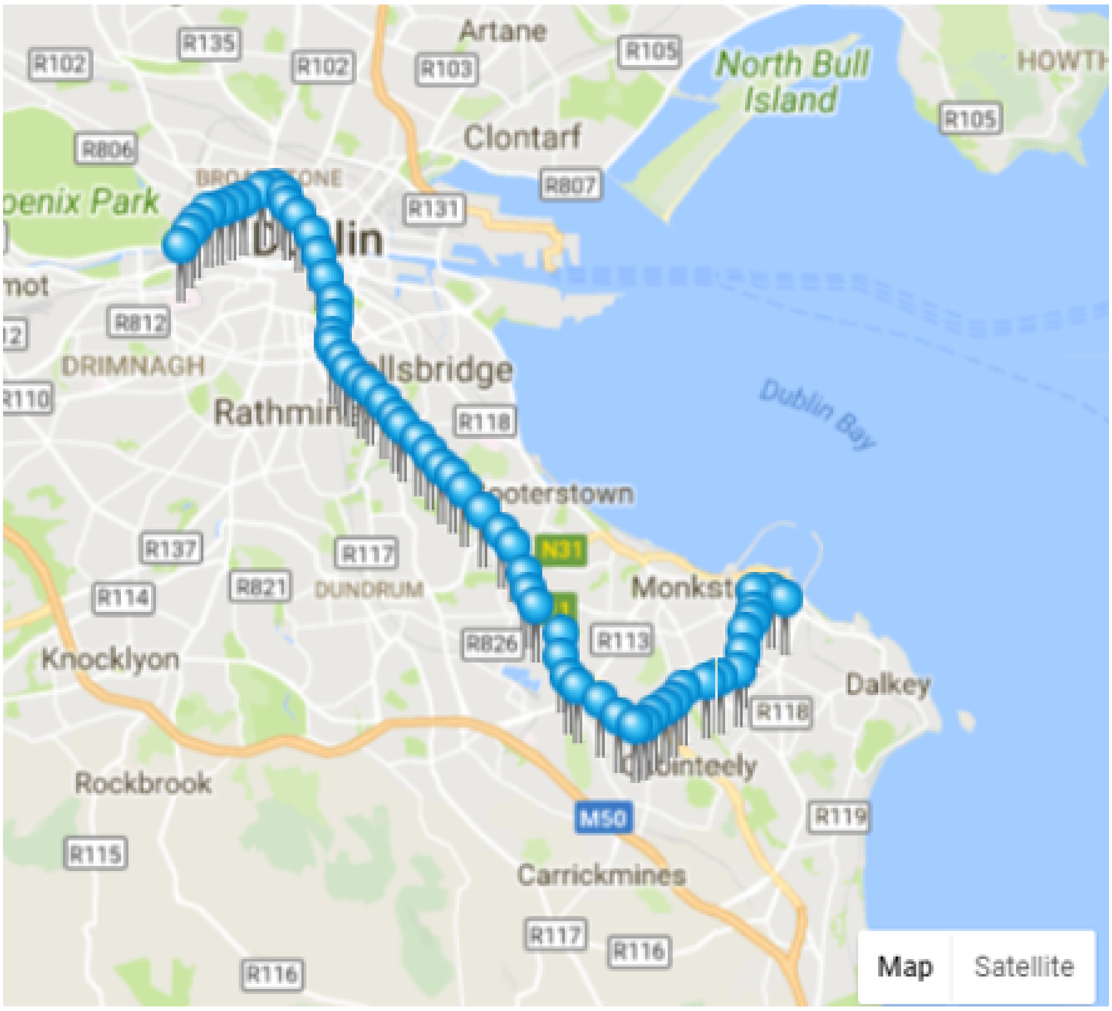


Figure 8. Route for bus line 46A [2]

2. All journeys that had their data points inside a 100m by 100m square were tagged bad journey.

3. Journeys that had data points with large time jumps in it were tagged bad journeys. That is if the time difference between the last data point and the next data point is above two minutes, the journey is tagged bad journey. This was done because, the expected time difference is twenty seconds, and we understand that it is possible for tall buildings and trees to cause delays in transmission. However, Anything above 2 minutes is tagged bad journey and deleted from the dataset.

4. Lastly, for each journey in the dataset, journeys with less than fifteen stops throughout the lifetime of the journey were tagged bad journey.

The dataset also lacked some values that are important to the research being carried out, for example, the average speed was missing, the journey time between each stop and also the distance between each stop. All three values were computed as follows:

- **Estimated Travel Time:** Two types of estimated travel time were computed, the estimated travel time between individual stops that is the time it took a bus to move from stop S_i to S_{i+1} and the estimated total travel time for any given journey referred to as the journey travel time. Therefore, the travel time between any two given stops a and b for a given journey J is defined as the difference between arrival time at stop b and arrival time at stop a. Defined mathematically as:

$$TT^J_{a,b} = T^J_{Ab} - T^J_{Aa} \tag{15}$$

  The total Travel Time for a particular journey J is defined as the difference between departure time at origin a and arrival time at final destination b. Defined mathematically as:

$$JT^J_{i,j} = T^J_{Ab} - T^J_{Da} \tag{16}$$

- **Distance:** The distance between any two given stops a and b was calculated using Open Source Routing Machine (OSRM) platform. OSRM is a platform equipped with high performance engine designed for calculating paths in road network. Its distance calculation accuracy was the reason OSRM was adopted. The distance was calculated by using the latitude and longitude of both arrival stops and destination stops as query for the OSRM server and the result is the distance between both stops while considering the public transit route only. The total distance is then computed by adding the values together. This is done to avoid measuring distance using the crow fly method.

- **Speed:** The average speed in kilometer per hour between any two stops a and b was calculated using:

$$AS_{a,b}^{J} = \frac{D_{a,b}}{TT_{a,b}^{J}} * 3.6 \tag{17}$$

Where:

$TT_{a,b}^{J}$ is the travel time between any stop a and b for journey J

$JT_{i,j}^{J}$ is the total travel time for journey J.

$D_{a,b}$ is the distance between any two stops a and b.

$AS_{a,b}^{J}$ is the average speed between any two stops a and b for journey J.

T_i is the time stamp observed at any stop i.

4.3 Data Analysis

To determine the correlation between the journey times and other variables in the dataset, it was necessary to analyze the filtered dataset. The distribution of the travel time across different hours of the day for different days of the week was first analyzed for both the south and north bound journey. Figure 9 shows that the peak hours are always different for weekdays and weekends. For weekends (Saturday and Sunday), the peak times are between 10:00 am to 12:00 pm and 14:00 pm to 16:00 pm. This is quite understandable because people only tend to move towards the city center around noon or afternoon to relax or have some fun.

For weekdays (Monday to Friday), the peak hours observed for days starting the week (Monday) and ending the week (Friday) can be seen to be slightly different for days within the week. The peak hours for Friday and Monday share some kind of similarities as seen in Figure 9. The peak hours for these days are between 07:00 am to 10:00 am, 12:00 pm to 14:00 pm and 16:00 pm to 18:00 pm. While the peaks hours for days within the week are 07:00 am to 08:am and 16:00 pm to 18:00 pm.

Peak hour observed between 07:00 am to 08:00 am on weekdays shows that people are moving to their work place while peak hour observed between 16:00 pm to 18:00 pm shows that people are moving to go home. The different peak hour observed on Monday and Friday is quite understandable because commuters often want to start the week on a brighter note and are also in a rush to start the weekend early on Fridays, thus leading to high traffic rate on Mondays and Fridays.

The same analysis was carried out for the south bound journey and Figure 10 shows that the peak hours are slightly different from the north bound journey. For weekends (Saturday and Sunday), no major peak hours were recorded while for weekdays, peak hours were between 07:00 am to 08:00 am and 16:00 pm to 18:00 pm and it remained the same for all weekdays.

From the analysis shown Figure 9 and Figure 10, a conclusion can be drawn that if a journey J is just before peak hour period, it is likely that journey J+1 plying the same route and direction will likely have a higher Journey Travel Time.

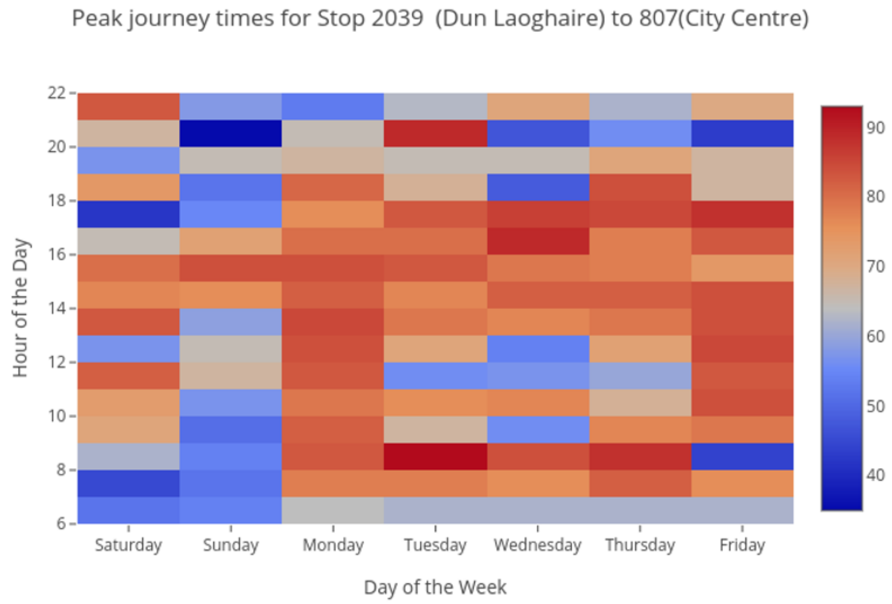


Figure 9. Heat Map for North Bound Journey

Furthermore, the travel time distribution for the south bound journey from Dun Laoghaire (Stop 2039) to Phoenix part (Stop 807) on weekdays was analyzed. Figure 11 shows the distribution and from the plot it can be seen that the travel times in the dataset follow a normal distribution curve. The left half of the plot represents the non-peak periods while the right half of the plot represents the travel time distribution at peak-periods.

From Figure 11, we can deduce that the average travel time in the dataset is 80 minute and with any particular trip starting at Stop 2039 to destination Stop 807, there is a 68% probability that it will be within 10 minutes from the average travel time.
In addition, the average number of stops per schedule from Dun Laoghaire To Phoenix

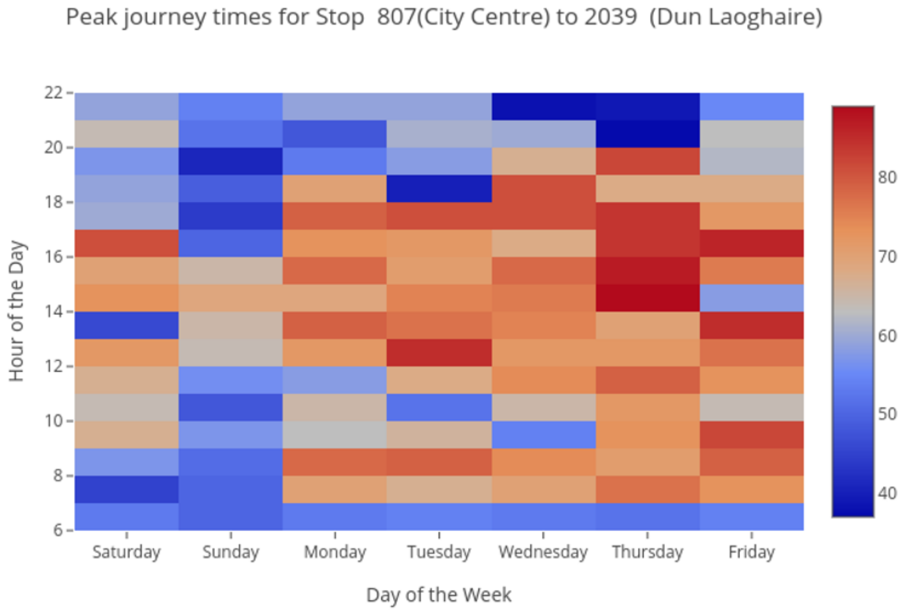


Figure 10. Heat Map for South Bound Journey

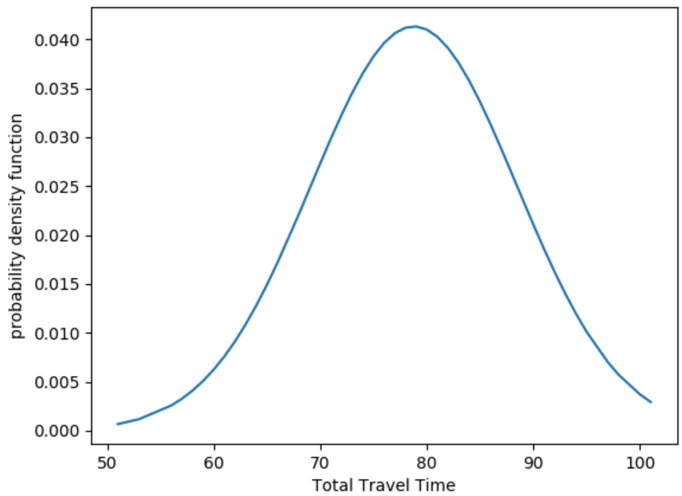


Figure 11. Travel Time distribution

Park was analyzed. The plot in Figure 12 is the result of the analysis and it shows that during the morning peak periods and evening peak periods between 07:00 to 08:00

and 16:00 - 18:00 respectively, the number of stops observed is high. Based on what was explained in chapter 1, this factor also contributes to the travel time variability experienced during this period. Also, has the hour approaches a peak hour, it can be seen that the average number of stops tends to increase. For example, the average number of stops observed tends to start increasing from 06:45 as we approach 07:00 which is a peak hour.

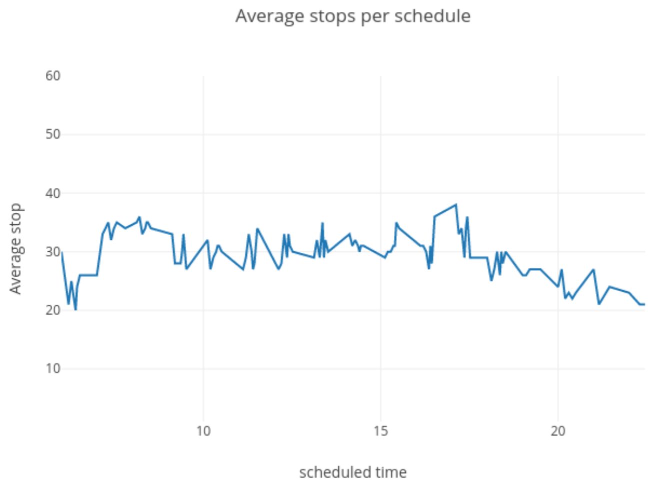


Figure 12. Average stop per schedule

Figure 12 also shows that on the average, only 60% of the total number of scheduled stop is always observed. This is quite understandable because a bus is only expected to stop when there is a passenger who wants to alight or board the bus.

4.4 Summary

In this chapter, the GPS dataset collected to be used as test bed in this study was introduced. The dataset collection method and columns in the dataset were all described. Also, the filtering algorithms that was used in removing outliers were described in detail. Finally, the preprocessed dataset was analyzed in order to understand the dataset and to easily identify independent variables that lead to travel time variation in route 46A.

5 Result and Analysis

5.1 Model Evaluation

As discussed in the previous chapter, a neural network was modeled and adaGrad is the learning algorithm of choice. The number of input neurons used is 6 which is same as the number of independent variables in the dataset while the number of neurons used in output layer is 1 which is the dimension of travel time T to be predicted. To determine the number of hidden layers to use and the number of neurons in each layer, we conducted an experiment to obtain the best combinations of values to use and these values are presented in Table 6.

Table 6. NN configuration for short jumps

Details	**Value**
Input Layer	6 neuron
First hidden Layer	12 neuron
Second hidden Layer	50 neuron
Output Layer	1 neuron
Epoch	1000
Training Algorithm	adaGrad

Furthermore, in order to improve the prediction accuracy of the model, the dataset was divided into two classes according to their estimated travel time before being used as an input to the neural network. Each class of the dataset is passed into a different neural network with different configurations at the hidden layers. Figure 13 gives a pictorial description of how this division was done.

For short jumps the NN configuration in Table 6 was used while for long jumps, the NN configuration in Table 7 was used. When prediction is done, the result of the two network is then combined.

Table 7. NN configuration for long jumps

Details	**Value**
Input Layer	6 neuron
First hidden Layer	9 neuron
Second hidden Layer	25 neuron

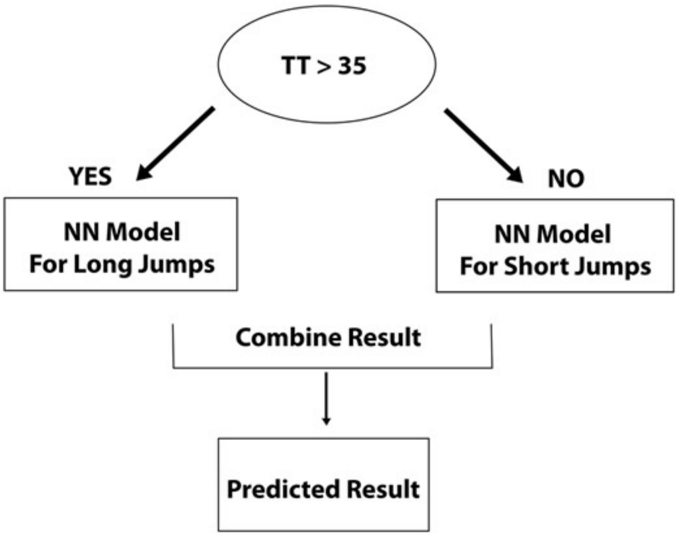


Figure 13. Dataset class division

The performance of the model was evaluated in terms of accuracy by using Root Mean Square Error (RMSE). That is the predicted travel time (TT) was tested against the ground truth value which is the estimated travel time discussed in section 4.

$$RMSE = \sqrt{\frac{\sum(tt^j_{a,b} - TT^j_{a,b})^2}{z}} \tag{18}$$

Where:

$tt^j_{a,b}$ = the observed travel time from stop *a* to stop *b* for a journey *j* in testing data.

$TT^j_{a,b}$ = the ground truth travel time from stop *a* to stop *b* for a journey *j* in testing data.

z = the total number of links in the journey.

5.2 Result

The first part of the prediction involves querying a specific day of the week over the one month dataset and predicting all journeys of the last occurrence of the selected day in the dataset. For example, the journeys for last Tuesday of the month was predicted by using the first 3 Tuesdays of the month as input into the NN model.

Figure 14 presents the behavior of the model when used to predict a single journey outside peak hours. It can be seen that the difference between the estimated travel time between two links is approximately equal to two minutes. That is, the model either accurately predicts the travel time or is off by approximately two minutes.

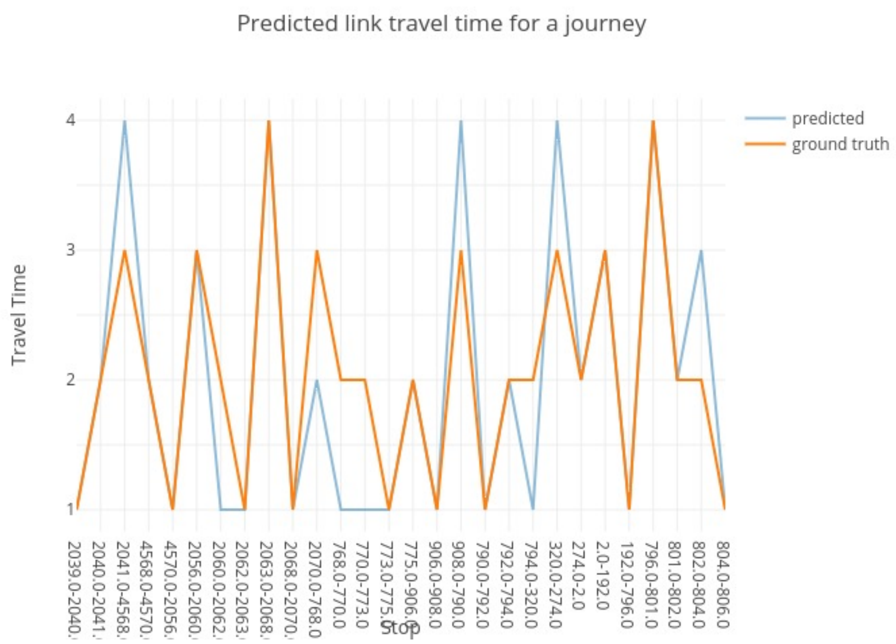


Figure 14. Predicted link travel time for a journey outside peak hour

In Figure 15, the result of using the model to predict a journey inside a peak hour (16:00 is an example of peak period as shown in the analysis done in chapter 3) is shown. It can be seen that the model is not affected by peak hour as it also gives the same output has the non-peak period.

In Figure 16, the graph for RMSE observed for both peak hours and non peak hours for predictions made for the last weekdays of the month by querying each of the days is presented. The figure shows that the model performed better for peak hours compared to normal hours while on Sunday the error between the peak and non-peak periods is high. This is because the peak hour on Sunday is short, thus making it difficult for the model

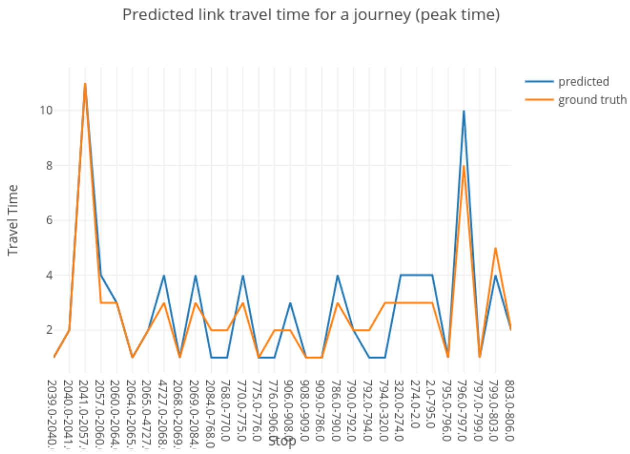


Figure 15. Predicted link travel time for a journey inside peak hour

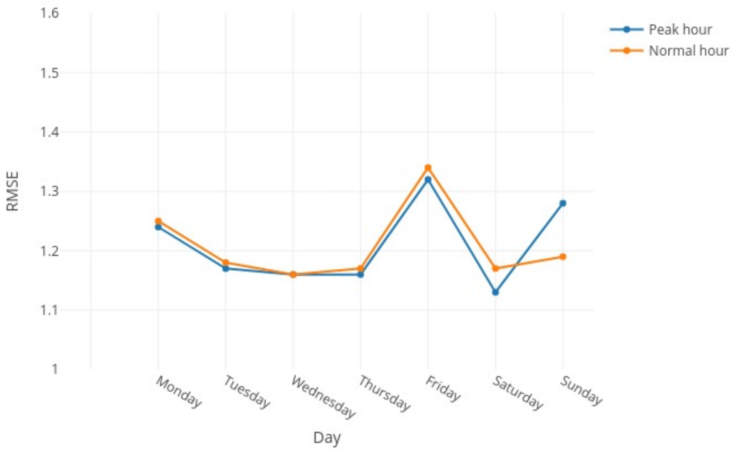


Figure 16. RMSE error observed by day for Non peak and peak periods

to learn the peak periods during short trips.

From the result obtained for the short jumps, we can conclude that the model does not predict accurately enough for short trips. This often occur for stops where the travel time is between 1 minute and 3 minutes. The inaccurate predictions are mostly due to travel time variability caused by stops made at traffic lights during short trips. The time spent at traffic lights are often unpredictable and thus, increases the level of uncertainty for this type of trips in the dataset.

For long jumps, Figure 17 presents the RMSE observed per day. As expected, the RMSE for long jumps are larger than that of short jumps, however, the prediction accuracy at 88% accuracy is still better than short jumps. The difference between the predicted and the ground truth is often between 3 and 4 minutes, which is acceptable given the type of data used and the factors considered. The model is also able to capture but non-peak and peak periods as shown from the plot. The error margin between peak hour and non-peak hours on Sunday for long trips is very low compared to that of short trips.

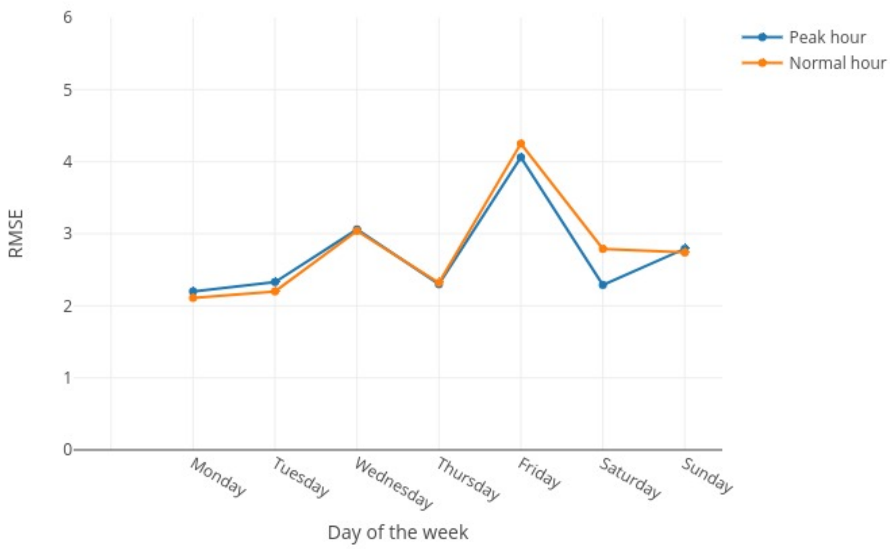


Figure 17. RMSE error observed per day for long jumps during peak and non-peak periods

To understand the behavior of the model when used with a bigger dataset, we feed the NN model with all the dataset except the last five days which equate to the last 5 working days of the month. To achieve good performance, another input variable X_7 called day of the week had to be introduced. The new NN model is shown in Figure 18 below.

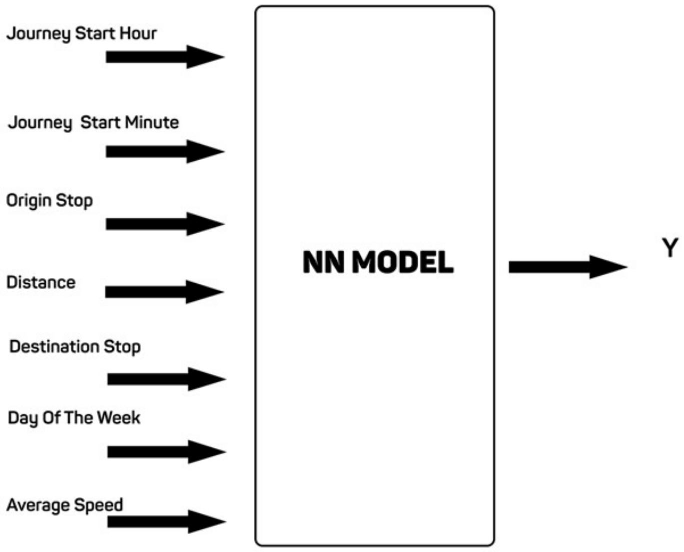


Figure 18. Neural Network Model with Day of the Week as input variable

The prediction error computed using RMSE is presented for each day for both long and short jumps for different time periods (peak and non-peak periods). It can be seen in Table 8 that there is little difference between the prediction error for the two different time periods. This means that the model captured both peak and non-peak periods and was also better than the previous model for both periods. However, the model still had difficulties in predicting accurately short trips with travel time between 1 and 3 minutes.

Table 8. RMSE for short jumps

Days	**Nonpeak**	**Peak**
Monday	1.17	1.18
Tuesday	1.18	1.19
Wednesday	1.20	1.16
Thursday	1.17	1.20
Friday	1.17	1.15

In Table 9, the prediction error for long jumps during non-peak and peak hours is presented. Again, the table shows that the model is able to predict long trips with low error margin. The prediction made by the model are accurate to within 3 minutes of the ground truth for trips exceeding 60 minutes. For the type of route considered in this research, this margin is acceptable and can be improved if more data are available.

Table 9. RMSE for long jumps

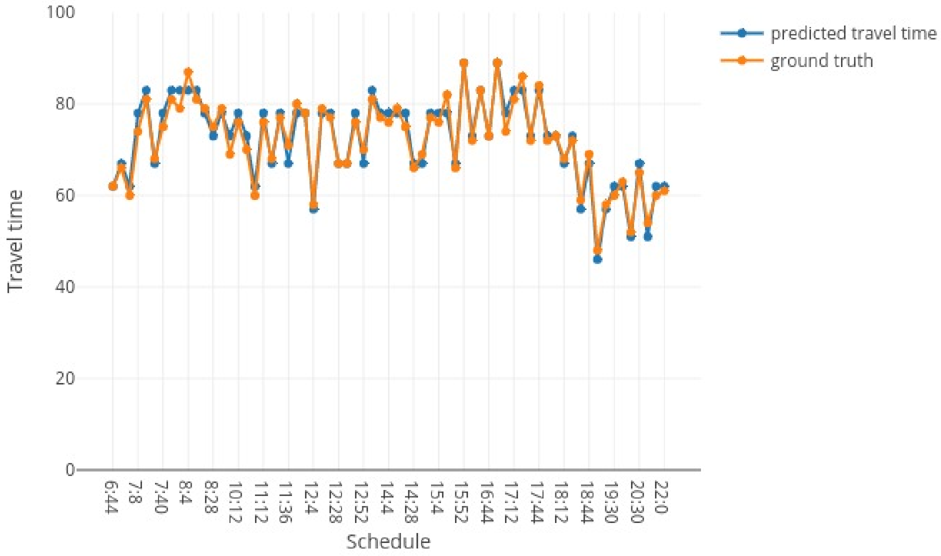
Days	**Nonpeak**	**Peak**
Monday	2.48	2.56
Tuesday	2.53	2.58
Wednesday	2.53	2.60
Thursday	2.60	2.61
Friday	2.48	2.52

Finally, the performance of the model when predicting the total travel time of trips in the future is analyzed. In Table 10 the RMSE for both peak and non-peak periods are presented and from the table, it can be seen that the RMSE for non-peak is in most cases lower than peak periods. However, the differences are not much, which makes us conclude that the model is able to capture both peak and non-peak periods sufficiently.

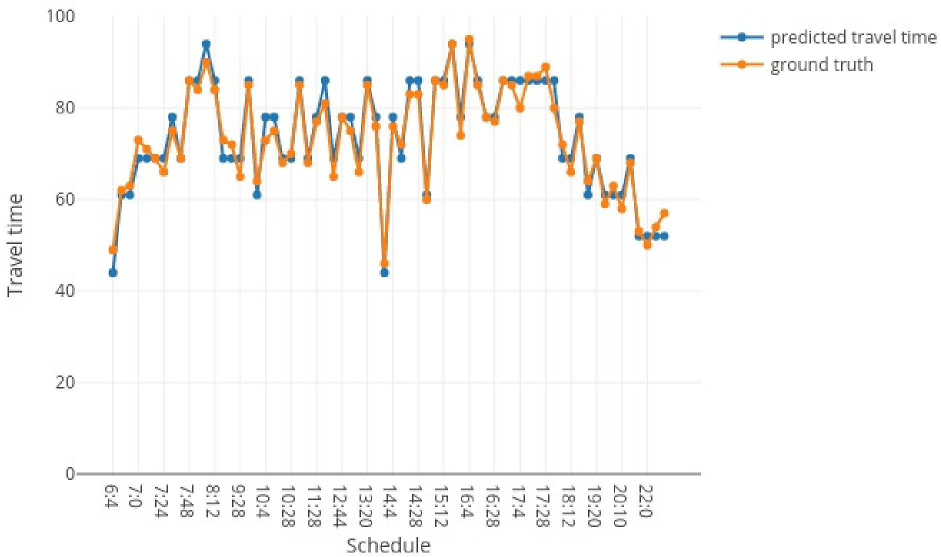
Table 10. RMSE for predicted trip travel time

Days	**Non-Peak**	**Peak**
Monday	2.50	2.76
Tuesday	2.55	2.72
Wednesday	2.75	3.12
Thursday	2.44	3.24
Friday	3.01	2.66

In Figure 19, the plot for the predicted travel time against the ground truth travel time per schedule is presented for all week days. From the plot, we can deduce that the NN model is able to predict the total travel time efficiently with low error margin. The model was also able to identify peak and non-peak periods without using input variables to differentiate between both periods. Lastly, the missing schedules in some of the days are due to the filter applied on the dataset discussed in previous chapter.

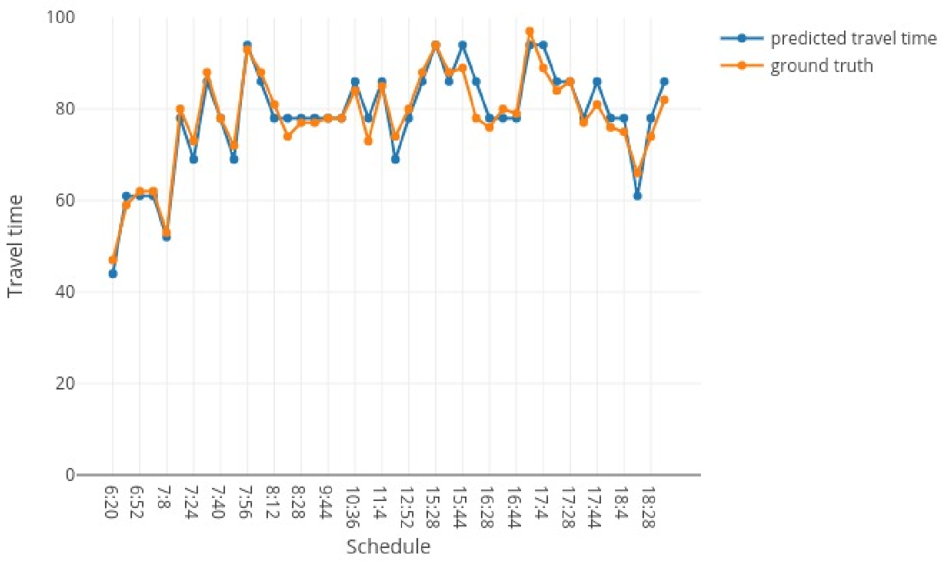


(a) Monday

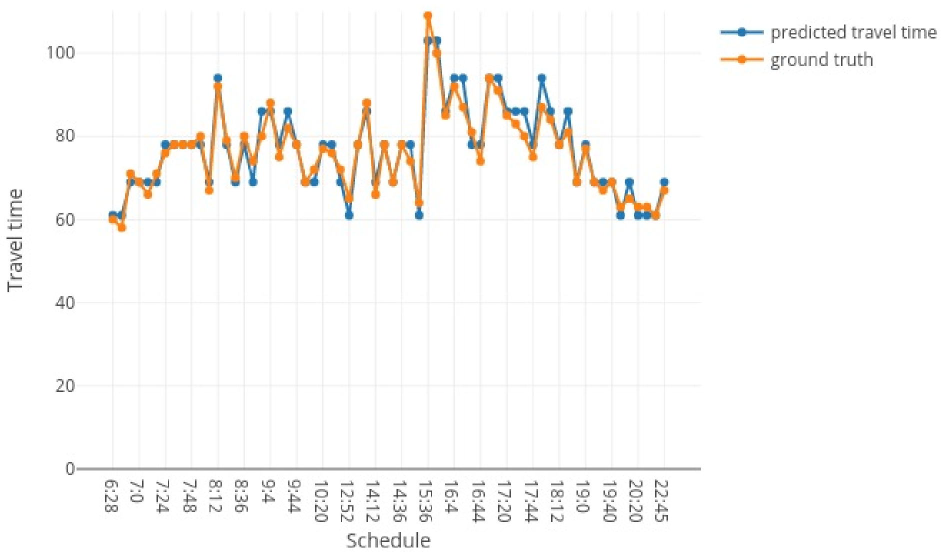


(b) Tuesday

Figure 19. Predicted travel time vs Ground truth per Schedule

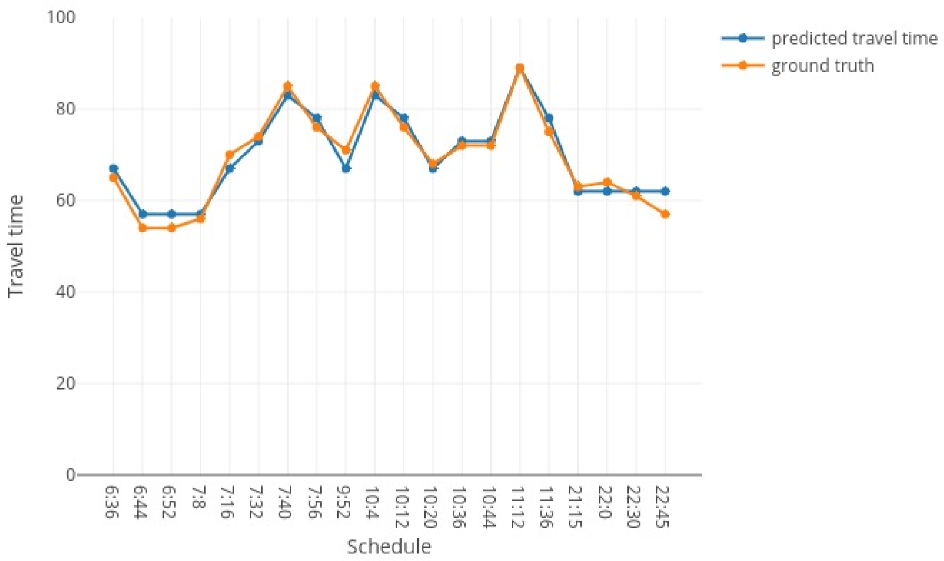


(c) Wednesday



(d) Thursday

Figure 19. Predicted travel time vs Ground truth per Schedule



(e) Friday

Figure 19. Predicted travel time vs Ground truth per Schedule

5.3 Summary

In this chapter, the prediction accuracy of the model measured using RMSE was analyzed and presented in tables and graph plots. From the result analyzed and discussed in this chapter, we can conclude that the model performs best with long trips for both non-peak and peak periods. The result showed that the NN model is capable of making near accurate predictions when the number of stops between the origin and destination is at least 4 stops. However, if the travel time between two or three stops is between 1 and 3 minutes, the model gives poor prediction with prediction accuracy close to 70%. This is because the variability for short trips are often caused by time spent at traffic lights at intersections which are often unpredictable. Although, given a bigger dataset of two to three months, the NN model might be capable of learning variability caused by traffic light on very short trips. Additional analysis also showed that when the travel time is between $40 - 100$ minutes, the average difference between the predicted and ground truth is between $3 - 5$ minutes for both peak and non-peak periods.

The result analyzed in this chapter showed that the model is sufficient enough for making travel time prediction on bus routes in cities like route 46a.

6 Conclusion and Future Works

This chapter marks the conclusion of the thesis, presents the limitation of the thesis and lastly, presents what extra work needs to be done to achieve better prediction result.

6.1 Conclusion

In this thesis, we discussed the factors affecting travel time variability in public bus transit system and also reviewed different methods that have been applied to solve travel time prediction problem based on data obtained through GPS. The use of NN and OD matrix for travel time prediction was explored in this research and the choice of route was route 46A in the city of Dublin. The route is popularly known for its lack of lane discipline and travel time variability.

Although this is not the first time NN has been applied to travel time prediction problem, however, for each NN model used in the past, a unique input-output combination is used.

The travel time predicted by the model was compared with the ground truth value and the result of the comparison was analyzed using RMSE to determine the model's prediction accuracy. The result of the analysis presented in chapter 5 showed that the model is promising and can be used by public transport agencies to build a prediction system that is capable of providing travel time information of trips based on schedules in the future. We are confident that the NN model implemented in this research is sufficient enough to improve the reliability of public transit system if applied.

6.2 Limitations and Future perspectives

Unfortunately, the prediction accuracy of the model was affected by the size of the dataset and type of variable available. The dataset used in this research was approximately equal to a month dataset and we believe that using a bigger dataset of 5-7 months would have helped achieve better accuracy in prediction. Also, factors like weather conditions, driver swap periods, traffic light and accident time periods were not captured in this research due to the structure of the dataset used, thus contributed to the prediction error of the model.

The research done in this thesis and the prediction accuracy achieved by the NN model is a good starting point for future works. A future work based on the application of Bayesian methods can be inspired by the work done by *Jianying et al* in [16]. *Jianying et al* in [16] applied Kalman filter to the predicted output of their proposed neural network to achieve better result. Likewise, Bayesian methods like Monte carlo and Markov chain can be applied on the travel time predicted by the NN model used in this research to reduce the prediction errors in the absence of large dataset.

Bradford et al in [31], applied Markov chain Monte Carlo method to estimate the travel time of ambulances and the estimation accuracy showed that Bayesian methods are good tool for travel time prediction. Although, the case study used in the study is not related to public bus transit system, we believe that exploring the use of Bayesian methods alongside the NN model used in this research will be worth it.

References

[1] Y. L. Yanjie Duan and F.-Y. Wang, "Travel time prediction with lstm neural network," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (•, ed.), 2016.

[2] C. Ltd, "46a bus timetables," 2011. `https://hittheroad.ie/bus/46a#to=Phoenix+Park`.

[3] M. G. Iraj Bargegol, Mahyar Ghorbanzadeh and M. Rastbod, "Evaluation of effective factors on travel time in optimization of bus stops placement using genetic algorithm," in *IOP Conference Series: Materials Science and Engineering 245 042002*, 2017.

[4] C. B. Mogens Fosgerau, Katrine Hjorth and D. Fukuda, "Travel time variability: Definition and valuation," tech. rep., DTU Transport, 2008.

[5] C. Goodwin, "Bus journey time variability in urban areas: Pilot study and analysis report," tech. rep., Department for Transportation Framework for Transport-Related Technical and Engineering Advice and Research, 2015.

[6] S. V. H. Christopher Hedden, Dan Krechmer and A. Toppen, "Strategies for improving traveler information," 2011.

[7] L.-J. C. Wang-Chien Lee, Weiping Si and M. C. Chen, "Http: a new framework for bus travel time prediction based on historical trajectories," in *SIGSPATIAL '12: Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pp. 279–288, 2012.

[8] R. S. L. Vanajakshi, S.C. Subramanian, "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses," vol. 3, 2009.

[9] T. N. Zegeye Kebede Gurmu and Perkins, "Artificial neural network travel time prediction model for buses using only gps data," *Journal of Public Transportation*, vol. 17, 2014.

[10] G. B. Greg Welch, "An intriduction to kalman filter," tech. rep., University of North Carolina at Chapel Hill Department of Computer Science Chapel Hill, NC 27599-3175, 2004.

[11] J.-S. Yang, "Travel time prediction using gps test vehicle and kalman filtering techniques," in *American Control Conference. Proceedings of the 2005*, IEEE, 2005.

[12] D. J. Simon, "Using nonlinear kalman filtering to estimate signals," 2006.

[13] S. X. J. Huifeng, X. Aigong and L. Lanyong, "The applied research of kalman in the dynamic travel time prediction," in *Proc. 18th Int. Conf. Geoinformatics* (•, ed.), 2010.

[14] J. S. Johar Amita and G. P.K, "Prediction of bus travel time using ann: A case study in delhi," in *Transportation Research Procedia*, pp. 263–272, Elsevier B.V., 2016.

[15] W. P.-A. Wichai Treethidtaphat and S. Khaimook, "Bus arrival time prediction at any distance of bus route usiing deep neural network model," in *20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 988–992, 2017.

[16] X. G. Jianying Liu, Wendong Wang and X. Que, "A hybrid model based on kalman filter and neural network for traffic prediction," in *Cloud Computing and Intelligent Systems (CCIS), IEEE 2nd International Conference* (•, ed.), 2012.

[17] Y. W. Zhihao Zhang, Peng Chen and G. Yu1, "A hybrid deep learning approach for urban expressway travel time prediction considering spatial-temporal features," in *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC): Workshop*, 2017.

[18] A. S. Avishai Mandelbaum, Francois Schnitzler and M. Weidlich, "Traveling time prediction in scheduled transportation with journey segments," *Information Systems. Data: Creation, Management and Utilization*, vol. Volume 64, pp. 266–280, March 2017.

[19] F. B. I Ekowicaksono and A. Aman, "Estimating origin-destination matrix of bogor city using gravity model," in *IOP Conf. Series: Earth and Environmental Science 31*, IOP, 2016.

[20] R. S. Adrien Ickowicz, "Estimation of an origin/destination matrix: application to a ferry transport data," *Public Transport, planning and operations.*, 2015.

[21] A. Kar, "Stock prediction using aritificial neural network," 2015.

[22] *Artificial Neural Network, A Practical Course*, ch. 2. Springer International Publishing, 2017.

[23] A. H. John McGonagle, George Shaikouski, "Backward propagation," 2011.

[24] Z. K. Gurmu, "A dynamic prediction of travel time for transit vehicles in brazil using gps data," Master's thesis, University of Twente, 2010.

[25] A. C. K. Wong, "Travel time prediction model for regional bus transit," Master's thesis, University of Toronto, 2009.

[26] B. Sharma and P. Venugopalan, "Comparison of neural network training functions for hematoma classification in brain ct images," 2014.

[27] E. H. John Duchi and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," 2011.

[28] S. Ruder, *2016*, An overview of gradient descent optimization algorithms.

[29] M. I. J. John C. Duchi and H. B. McMahan, "Estimation, optimization, and parallelism when data is sparse," in *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2832–2840, December 2013.

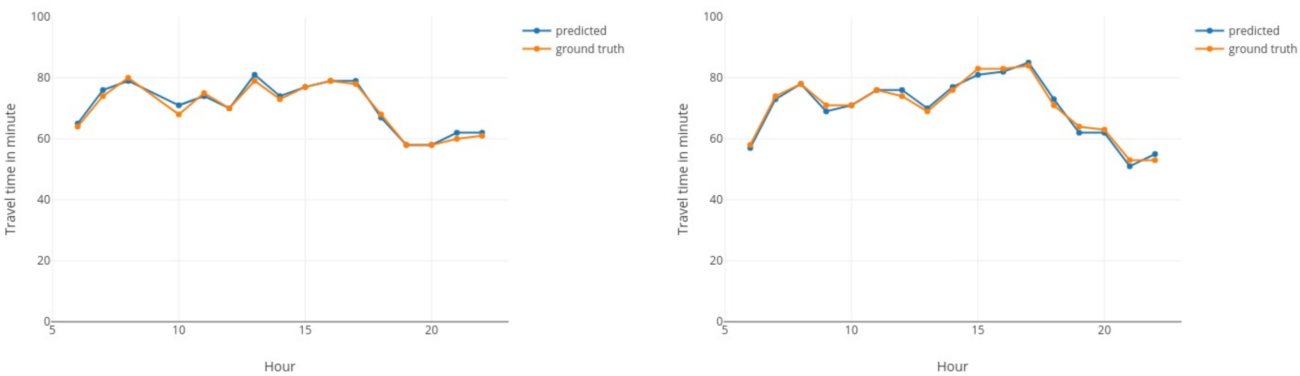
[30] D. C. Council, "Dublin bus gps sample data from dublin city council," 2012. `https://data.smartdublin.ie/dataset/dublin-bus-gps-sample-data-from-dublin-city-council-insight-project`.

[31] D. S. M. Bradford S Westgate, Dawn B Woodard and S. G. Henderson, "Travel time estimation for ambulances using bayesian data augmentation.," vol. 7, no. 2, 2013.

Appendix

Origin	2039	2040	2041	2042	4569	4570	2056	2057	2059	2060	2064	4727	2068	2070	768	770	772	906	907	908	909	845	790	794	320	274	2	192	795	796	797	799	800	811	804	807
2039	0	440	580	764	1325	1384	1484	1624	1686	1763	2027	2124	2287	2385	2586	2746	2805	3044	3104	3147	3224	3284	3405	3624	3803	3965	4205	4342	4422	4461	4621	4703	4742	4781	5041	5450
2040	0	0	140	324	885	944	1044	1184	1246	1323	1587	1684	1847	1945	2146	2306	2365	2604	2664	2707	2784	2844	2965	3184	3363	3525	3765	3902	3982	4021	4181	4263	4302	4341	4601	5010
2041	0	0	0	184	745	804	904	1044	1106	1183	1447	1544	1707	1805	2006	2166	2225	2464	2524	2567	2644	2704	2825	3044	3223	3385	3625	3762	3842	3881	4041	4123	4162	4201	4461	4870
2042	0	0	0	0	561	620	720	860	922	999	1263	1360	1523	1621	1822	1982	2041	2280	2340	2383	2460	2520	2641	2860	3039	3201	3441	3578	3658	3697	3857	3939	3978	4017	4277	4686
4569	0	0	0	0	0	59	159	299	361	438	702	799	962	1060	1261	1421	1480	1719	1779	1822	1899	1959	2080	2299	2478	2640	2880	3017	3097	3136	3296	3378	3417	3456	3716	4125
4570	0	0	0	0	0	0	100	240	302	379	643	740	903	1001	1202	1362	1421	1660	1720	1763	1840	1900	2021	2240	2419	2581	2821	2958	3038	3077	3237	3319	3358	3397	3657	4066
2056	0	0	0	0	0	0	0	140	202	279	543	640	803	901	1102	1262	1321	1560	1620	1663	1740	1800	1921	2140	2319	2481	2721	2858	2938	2977	3137	3219	3258	3297	3557	3966
2057	0	0	0	0	0	0	0	0	62	139	403	500	663	761	962	1122	1181	1420	1480	1523	1600	1660	1781	2000	2179	2341	2581	2718	2798	2837	2997	3079	3118	3157	3417	3826
2059	0	0	0	0	0	0	0	0	0	77	341	438	601	699	900	1060	1119	1358	1418	1461	1538	1598	1719	1938	2117	2279	2519	2656	2736	2775	2935	3017	3056	3095	3355	3764
2060	0	0	0	0	0	0	0	0	0	0	264	361	524	622	823	983	1042	1281	1341	1384	1461	1521	1642	1861	2040	2202	2442	2579	2659	2698	2858	2940	2979	3018	3278	3687
2064	0	0	0	0	0	0	0	0	0	0	0	97	260	358	559	719	778	1017	1077	1120	1197	1257	1378	1597	1776	1938	2178	2315	2395	2434	2594	2676	2715	2754	3014	3423
4727	0	0	0	0	0	0	0	0	0	0	0	0	163	261	462	622	681	920	980	1023	1100	1160	1281	1500	1679	1841	2081	2218	2298	2337	2497	2579	2618	2657	2917	3326
2068	0	0	0	0	0	0	0	0	0	0	0	0	0	98	299	459	518	757	817	860	937	997	1118	1337	1516	1678	1918	2055	2135	2174	2334	2416	2455	2494	2754	3163
2070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	201	361	420	659	719	762	839	899	1020	1239	1418	1580	1820	1957	2037	2076	2236	2318	2357	2396	2656	3065
768	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	219	458	518	561	638	698	819	1038	1217	1379	1619	1756	1836	1875	2035	2117	2156	2195	2455	2864
770	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	298	358	401	478	538	659	878	1057	1219	1459	1596	1676	1715	1875	1957	1996	2035	2295	2704
772	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	239	299	342	419	479	600	819	998	1160	1400	1537	1617	1656	1816	1898	1937	1976	2236	2645
906	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	103	180	240	361	580	759	921	1161	1298	1378	1417	1577	1659	1698	1737	1997	2406
907	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43	120	180	301	520	699	861	1101	1238	1318	1357	1517	1599	1638	1677	1937	2346
908	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	137	258	477	656	818	1058	1195	1275	1314	1474	1556	1595	1634	1894	2303
909	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	181	400	579	741	981	1118	1198	1237	1397	1479	1518	1557	1817	2226
845	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121	340	519	681	921	1058	1138	1177	1337	1419	1458	1497	1757	2166
790	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	219	398	560	800	937	1017	1056	1216	1298	1337	1376	1636	2045
794	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	179	341	581	718	798	837	997	1079	1118	1157	1417	1826
320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	162	402	539	619	658	818	900	939	978	1238	1647
274	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	240	377	457	496	656	738	777	816	1076	1485
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	137	217	256	416	498	537	576	836	1245
192	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	119	279	361	400	439	699	1108
795	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	199	281	320	359	619	1028
796	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	160	242	281	320	580	989
797	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	121	160	420	829
799	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	78	338	747
800	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	299	708
811	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	260	669
804	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	409
807	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

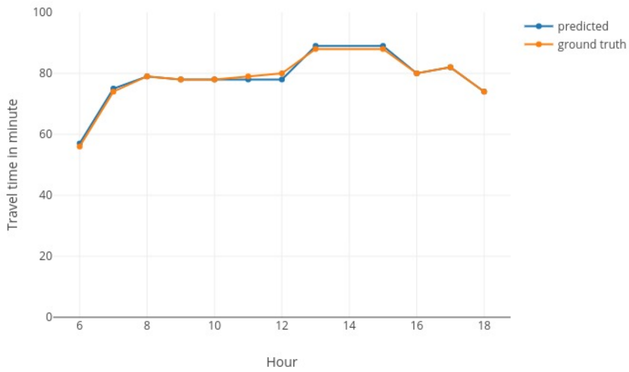
Figure 20. Od matrix showing travel time in Seconds



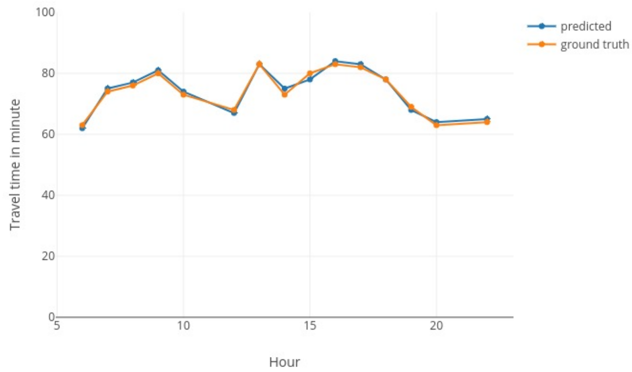
(a) Monday.

(b) Tuesday.

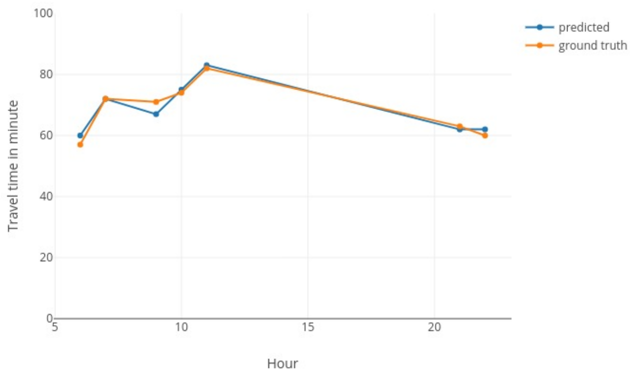
Figure 21. Average Travel Time vs Hour of the day.



(c) Wednesday.



(d) Thursday.



(e) Friday

Figure 21. Average Travel Time vs Hour of the day.

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Ayobami Ephraim Adewale**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis

   Link Travel Time Prediction Based on O-D Matrix and Neural Network Model

   supervised by Amnir Hadachi, PhD

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 21.05.2018