Technical University of Munich
Chair of Transportation Systems Engineering

Master's Thesis

On-demand Modeling and Forecasting Individual Upcoming Trips

Author:
Abdullahi Olakunle Fatola

Matriculation No:
03694325

Supervised by:
Dr. Tao Ma

Chair leader:
Univ.-Prof. Dr. Constantinos Antoniou

August 10, 2020

Abstract

This research was motivated by the challenges of transport supply, failing to fulfill the travel demand of commuters, which often leads to traffic congestion and disruption of business for public transport and mobility operators. Travel time is an essential determinant for planning individual upcoming trips. As a result, this research identified relevant data-driven indicators affecting the travel time of individual on-demand trips, developed and compared suitable deep learning models to predict the travel time, and also identified the computational costs of developing suitable travel time prediction models. The research strategy adopted was the exploratory case study of Chengdu city, China. Large data sets of commuters' on-demand ride requests, collected from the digital platform of a mobility service provider in China, Didi Chuxing, resulted in using a convenience sampling approach and a quantitative research analysis method to achieve the research objectives. The model evaluation metrics are Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of determination (R-squared). The findings indicated that model inputs, such as the departure time, travel distance, traffic analysis zones, and most importantly, travel speed, are highly relevant data-driven indicators influencing the travel time of individual on-demand trips. Furthermore, Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) were the deep learning networks developed, evaluated, and compared. The LSTM model 5 has the highest travel time prediction accuracy of 0.997, 0.059, and 0.364 for R-squared, MAE, and RMSE, respectively. Each deep learning model development consumed an average of 6.3 gigabytes SSD and 7.88 gigabytes RAM. At the same time, the LSTM had the highest overall computational cost of about 50 hours run-time, to execute all its models. The total run-time expended, for the 15 deep learning models, was approximately 5 days and 14 hours. Therefore, on-demand modeling and forecasting individual upcoming trips recorded its highest prediction accuracy, using the travel speed and other identified data-driven indicators derived from the trip origin timestamps, origin coordinates, and destination coordinates. To develop the LSTM model, which had the highest prediction accuracy, high computational power and computer capacity are required to reduce the run-time, and increase travel time prediction accuracy.

Keywords: Transport supply, on-demand trips, data-driven indicators, travel time, deep learning models, computational costs.

Acknowledgements

This thesis's successful accomplishment is due to the guidance, support, and invaluable motivations received from very important persons. I would like first to express my gratitude to Prof. Dr. Constantinos Antoniou, the Chair of Transportation Systems Engineering leader at the Technical University of Munich (TUM), for his acceptance to conduct this master's thesis research at the Chair.

Most notably, I wish to express my sincere appreciation to my supervisor, Dr. Tao Ma of the Chair of Transportation Systems Engineering, for his unquantifiable guidance and motivation. He selflessly offered his advice, time, vast knowledge, and experience to guide me from the first day I began my compulsory internship, also as my supervisor, to the end of the internship. Afterwards, he whole-heartedly accepted my request also to supervise this thesis, and his support had been immeasurable right from the selection of the thesis topic to the submission. He was accessible and always willing to help and answer my questions whenever I encounter difficulties during the research. His persistent supervision made the goal of this thesis realizable and I must acknowledge that he is a mentor and a genius.

I wish to equally appreciate the platform that provided the large data used for this thesis, and the platform is DiDi Chuxing GAIA Open Dataset Initiative in China. The request to use the "Chengdu City Second Ring Road Regional Trajectory Data Sets" for this thesis was seamlessly authorized and accepted. This data is the foundation of this research. Therefore, I would like to express my sincere gratitude to them for sharing the data and giving research students like me the opportunity to use their datasets for academic purposes.

I wish to also acknowledge the constant supports and motivations from my friends, Ogunmuyiwa Ayomini, Ibraheem Adeniran, and Tolulope Adesuyi. They stood by me till the end of this thesis and also offered their intellectual inputs as guidance to resolve problems faced during the research work.

Conclusively, I would like to express my deepest gratitude and special regards to my parents, Mr. Adewale Fatola and Mrs. Abiodun Fatola, as well as my siblings Fatai Fatola, Latifat Fatola, and Aishat Fatola, for their advice, encouragement, and motivation since the beginning of this masters program to the successful completion of this thesis. They kept me inspired during the thesis, and I could also not have accomplished the thesis without their inputs and supports.

I am whole-heartedly grateful to everyone whose assistance and guidance was a milestone in completing this thesis. Thank you very much, and your supports will always be valued.

Contents

Abstract i

Acknowledgements ii

Table of Contents iii

1 Introduction 1

1.1 Introduction . . . 1

1.2 Research background . . . 3

1.3 Objectives and research questions . . . 3

1.3.1 Objectives . . . 3

1.3.2 Research questions . . . 4

1.4 Research framework . . . 4

2 Literature review 6

2.1 Mobility-On-Demand (MOD) . . . 6

2.2 On-demand trips and travel time effects . . . 7

2.2.1 On-demand trips and the cycle . . . 7

2.2.2 Demand-responsive ride-sharing and e-hailing transits . . . 9

2.2.3 Impacts of travel time on on-demand trips . . . 10

2.3 Travel time and its indicators for on-demand trips . . . 11
2.3.1 Travel time description . . . 11
2.3.2 Indicators or factors influencing travel time . . . 12
2.4 Deep learning models for travel time prediction . . . 14
2.4.1 Artificial Neural Network (ANN) . . . 14
2.4.2 Convolutional Neural Network (CNN) . . . 17
2.4.3 Recurrent Neural Network (RNN) . . . 21
2.5 Evaluation metrics for travel time prediction models . . . 24

3 Research methodology **28**
3.1 Introduction to methodology . . . 28
3.2 Research strategy . . . 29
3.2.1 Exploratory case study and convenience sampling . . . 29
3.3 Data collection . . . 30
3.4 Data analysis framework . . . 32
3.4.1 Quantitative analysis and analytic tools . . . 32
3.4.2 Derivation of relevant travel time indicators . . . 33
3.4.3 Working mechanisms and tools for modeling . . . 36
3.5 Research limitations . . . 40

4 Data analysis and discussions **42**
4.1 Identification of relevant travel time indicators . . . 42
4.1.1 Statistical description of relevant indicators . . . 45
4.1.2 Distribution density of relevant indicators . . . 52
4.1.3 Spatial trip distribution of relevant indicators . . . 56

4.2 Developing deep learning models for travel time prediction 65
4.2.1 Artificial Neural Network (ANN) model results 68
4.2.2 Convolutional Neural Network (CNN) model results 81
4.2.3 Long Short-Term Memory (LSTM) model results 93
4.3 Comparison and evaluation of deep learning neural networks performance . . 106
4.4 Computational costs of developing the deep learning models 113
4.4.1 Computational run-time for the deep learning models 116

5 Conclusion and recommendations 119

5.1 Summary of findings and conclusions 119
5.2 Limitations and recommendations 121

Declaration 127

List of Figures

1 Research framework flow chart 5

2 On-demand transit request flow, adapted from (Liyanage and Dia, 2020) . . 8

3 On-demand transit complementing public and private transport (Hazan et al., 2019) 9

4 Travel time illustration adapted from (Duan et al., 2016) 12

5 Framework of travel time prediction and data sources, adapted from (Wu et al., 2020) 13

6 Structure of ANN neurons, adapted from (Gurmu and Fan, 2014) 15

7 Structure of ANN layers, adapted from (Gurmu and Fan, 2014) 15

8 Convolutional Neural Network (CNN) Architecture for spatio-temporal data of travel time prediction, adapted from (Ran et al., 2019b) 18

9 A Structure of LSTM Cell (Wei et al., 2018) 22

10 An architecture of LSTM Neural Network for travel time prediction (Ran et al., 2019a) 22

11 An Open Street Map (OSM) of Chengdu City, China (The case study area) . 31

12 An Open Street Map (OSM) for about 1.35 billion coordinates of origin-destination trips and trip routes in Chengdu City, China (The case study area) 31

13 The ellipsoidal triangle NAB (Karney, 2013) 35

14 Artificial Neural Network (ANN) Model Architecture 37

15 Convolution Neural Network (CNN) Model Architecture 39

16 Long Short-Term Memory (LSTM) Model Architecture 40

17 Open Street Map (OSM) for 7 million coordinates of on-demand trip departures in Chengdu City, China (30 days model training data) 43

18 Open Street Map (OSM) for 7 million coordinates of on-demand trip arrivals in Chengdu City, China (30 days model training data) 43

19 Open Street Map (OSM) for about 315 million time-step coordinates of on-demand trip routes in Chengdu City, China (7 days model test data) 44

20 Combined 1.35 billion coordinates of on-demand trip arrivals, departures and routes in Chengdu City, China using Open Street Maps (OSM) 44

21 Box-plot of departure hours and travel time 47

22 Box-plot of departure days and travel time 47

23 Highly Significant negative indicators with travel time 50

24 Highly Significant positive indicators with travel time 51

25 Distribution density of speed (km/h) in the training data of departures and arrivals 53

26 Distribution density of distance (km) in the training data of departures and arrivals 53

27 Distribution density of travel time (min) in the training data of departures and arrivals 54

28 Departure hour distribution density across the 30 days training data 54

29 Departure minute distribution density across the 30 days training data . . . 55

30 Distribution of departure trip coordinates per travel time (min) 56

31 Distribution of arrival trip coordinates per travel time (min) 57

32 Distribution of departure trip coordinates per travel distance (km) 57

33 Distribution of arrival trip coordinates per travel distance (km) 58

34 Distribution of departure trip coordinates per travel speed (km/h) 58

35 Distribution of arrival trip coordinates per travel speed (km/h) 59

36 Traffic Analysis Zone 1 60

37 Traffic Analysis Zone 2 60

38 Traffic Analysis Zone 3 61

39 Traffic Analysis Zone 4 61

40 Traffic Analysis Zone 5 62

41 Traffic Analysis Zone 6 62

42 Traffic Analysis Zone 7 63

43 Traffic Analysis Zone 8 63

44 Traffic Analysis Zone 9 64

45 Artificial Neural Network (ANN) model 1 training and validation loss curves 69

46 Artificial Neural Network (ANN) model 2 training and validation loss curves 69

47 Artificial Neural Network (ANN) model 3 training and validation loss curves 70

48 Artificial Neural Network (ANN) model 4 training and validation loss curves 70

49 Artificial Neural Network (ANN) model 5 training and validation loss curves 71

50 Artificial Neural Network (ANN) model 1 training and validation correlation 72

51 Artificial Neural Network (ANN) model 2 training and validation correlation 72

52 Artificial Neural Network (ANN) model 3 training and validation correlation 73

53 Artificial Neural Network (ANN) model 4 training and validation correlation 73

54 Artificial Neural Network (ANN) model 5 training and validation correlation 74

55 Artificial Neural Network (ANN) model 1 test correlation 75

56 Artificial Neural Network (ANN) model 2 test correlation 76

57 Artificial Neural Network (ANN) model 3 test correlation 76

58 Artificial Neural Network (ANN) model 4 test correlation 77

59 Artificial Neural Network (ANN) model 5 test correlation 77

60 Artificial Neural Network (ANN) models 1 travel time prediction per hour . 78
61 Artificial Neural Network (ANN) models 2 travel time prediction per hour . 79
62 Artificial Neural Network (ANN) models 3 travel time prediction per hour . 79
63 Artificial Neural Network (ANN) models 4 travel time prediction per hour . 80
64 Artificial Neural Network (ANN) models 5 travel time prediction per hour . 80
65 Convolutional Neural Network (CNN) model 1 training and validation loss curves 82
66 Convolutional Neural Network (CNN) model 2 training and validation loss curves 82
67 Convolutional Neural Network (CNN) model 3 training and validation loss curves 83
68 Convolutional Neural Network (CNN) model 4 training and validation loss curves 83
69 Convolutional Neural Network (CNN) model 5 training and validation loss curves 84
70 Convolutional Neural Network (CNN) models 1 training and validation correlation 85
71 Convolutional Neural Network (CNN) models 2 training and validation correlation 85
72 Convolutional Neural Network (CNN) models 3 training and validation correlation 86
73 Convolutional Neural Network (CNN) models 4 training and validation correlation 86
74 Convolutional Neural Network (CNN) models 5 training and validation correlation 87
75 Convolutional Neural Network (CNN) model 1 test correlation 88
76 Convolutional Neural Network (CNN) model 2 test correlation 88
77 Convolutional Neural Network (CNN) models 3 test correlation 89

78 Convolutional Neural Network (CNN) model 4 test correlation 89

79 Convolutional Neural Network (CNN) model 5 test correlation 90

80 Convolutional Neural Network (CNN) model 1 travel time prediction per hour 91

81 Convolutional Neural Network (CNN) model 2 travel time prediction per hour 91

82 Convolutional Neural Network (CNN) model 3 travel time prediction per hour 92

83 Convolutional Neural Network (CNN) model 4 travel time prediction per hour 92

84 Convolutional Neural Network (CNN) model 5 travel time prediction per hour 93

85 Long Short-Term Memory (LSTM) model 1 training and validation loss curves 94

86 Long Short-Term Memory (LSTM) model 2 training and validation loss curves 95

87 Long Short-Term Memory (LSTM) model 3 training and validation loss curves 95

88 Long Short-Term Memory (LSTM) model 4 training and validation loss curves 96

89 Long Short-Term Memory (LSTM) model 5 training and validation loss curves 96

90 Long Short-Term Memory (LSTM) model 1 training and validation correlation 97

91 Long Short-Term Memory (LSTM) model 2 training and validation correlation 98

92 Long Short-Term Memory (LSTM) model 3 training and validation correlation 98

93 Long Short-Term Memory (LSTM) model 4 training and validation correlation 99

94 Long Short-Term Memory (LSTM) model 5 training and validation correlation 99

95 Long Short-Term Memory (LSTM) model 1 test correlation 100

96 Long Short-Term Memory (LSTM) model 2 test correlation 101

97 Long Short-Term Memory (LSTM) model 3 test correlation 101

98 Long Short-Term Memory (LSTM) model 4 test correlation 102

99 Long Short-Term Memory (LSTM) model 5 test correlation 102

100 Long Short-Term Memory (LSTM) model 1 travel time prediction per hour . 103

101 Long Short-Term Memory (LSTM) model 2 travel time prediction per hour . 104

102 Long Short-Term Memory (LSTM) model 3 travel time prediction per hour . 104

103 Long Short-Term Memory (LSTM) model 4 travel time prediction per hour . 105

104 Long Short-Term Memory (LSTM) model 5 travel time prediction per hour . 105

105 Artificial Neural Network (ANN) model 4 training and validation loss curves 107

106 Convolutional Neural Network (ANN) model 4 training and validation loss curves 108

107 Long Short-Term Memory (LSTM) model 4 training and validation loss curves 108

108 Long Short-Term Memory (LSTM) model 5 training and validation loss curves 109

109 Artificial Neural Network (ANN) model 4 test correlation plot 109

110 Convolutional Neural Network (ANN) model 4 test correlation plot 110

111 Long Short-Term Memory (LSTM) model 4 test correlation plot 110

112 Long Short-Term Memory (LSTM) model 5 test correlation plot 111

113 Artificial Neural Network (ANN) model 4 real travel time prediction per hour 111

114 Convolutional Neural Network (CNN) model 4 real travel time prediction/hour 112

115 Long Short-Term Memory (LSTM) model 4 real travel time prediction per hour 112

116 Long Short-Term Memory (LSTM) model 5 real travel time prediction per hour 113

List of Tables

1 Ride-share variants (Agatz et al., 2012) 10

2 Model Parameter Estimates (Ennen and Heilker, 2020) 11

3 Travel time modelling (Sathya Prabha and Mathew, 2013) 13

4 The statistical summary of the origin-destination on-demand trip data . . . 46

5 The level of significance of the travel time indicators 48

6 The labels or inputs of the deep learning model 1 65

7 The labels or inputs of the deep learning model 2 66

8 The labels or inputs of the deep learning model 3 67

9 The labels or inputs of the deep learning model 4 and 5 67

10 Artificial Neural Network (ANN) models trained for travel time prediction . 68

11 Artificial Neural Network (ANN) model test results for travel time prediction 75

12 Convolutional Neural Network Models Trained for Travel Time Prediction . 81

13 Convolutional neural network model test results for travel time prediction . . 87

14 Long Short-Term Memory (LSTM) Models Trained for Travel Time Prediction 93

15 Long Short-Term Memory (LSTM) models test results for travel Time prediction 100

16 Comparison results of deep learning neural networks performance 106

17 Computational costs from data processing up to model development and predictions 115

Chapter 1

Introduction

1.1 Introduction

In early nineteenth (19th) century, the world experienced poor connectivity in the transportation of people, information and goods from their origins to various destinations and unfortunately towards the end of the century, a drastic increment in the movement of people was also recorded (Perlman, 2015). Consequently, to date, increased urbanization has been experienced in cities or central business districts around the world. As a result, traffic congestion and longer travel time among others have led to the advent of various transportation and mobility alternatives offered by transport authorities and operators in order to ensure efficiency and effectiveness in daily mobility.

These mobility solutions require taking into account the commuters' travel behaviour to establish a sustainable equilibrium between the demand and supply sides of transportation. Travel demand modeling and forecast were discovered to be required techniques meant for transportation engineers to determine the future volume of traffic and support their decision-making (Aloc and Amar, 2013). This approach of modeling and prediction borne the demand-responsive services offered by transport operators to cater for passengers mobility requests. The recent advancements in technology have also contributed to the approach of model development with the introduction of machine learning and deep learning tools to capture patterns, and give insights to trip distribution of commuters from the real-time and historical data collected by operators.

Transport authorities and operators have also prioritized the provision of real-time and static commuters' trip data to increase the modal shift from private transport to public transport, for faster and highly efficient transportation and mobility (Nuzzolo et al., 2015). Consequently, insights on the demand-responsive or on-demand trips of individual travellers can be captured with the aid of traveller information systems under the umbrella of the

Intelligent Transportation System (ITS). As researched by Nuzzolo et al. (2015), a traveller information system known as Advanced Transit Traveller Information Systems (ATTIS) processes and displays information for individuals to plan their trips and also know their route choice options from their intended origin to their destination, taking into account travel time and transit schedules of the transport services.

The data obtained from the time of travel, origin and destination are important information relevant to on-demand trip planning for both commuters and operators. On-demand mobility involves provision and processing of real-time trip information tailored towards individual trip purposes and needs. The result of the processed trip information is meant to ensure the availability of vehicles to individuals upon request, which is very much applicable to the shared vehicle system of Mobility-On-Demand (MOD) (Wilhelm et al., 2015). In this situation, the commuter provides origin and destination while the system decodes the time, longitude and latitude of the trip and route choice via the Global Positioning System (GPS).

Furthermore, fundamental decisions, such as departure time and route choice for drivers and transport operators, are dependent on predictive travel time information provided through the Advanced Traveler Information Systems (ATISs), which helps to improve travel quality for passengers (Yildirimoglu and Geroliminis, 2013). Travel time forecast is, therefore, an efficient and effective tool for mobility or transport operators to enhance their profitability and reshape their business models towards catering to passengers' on-demand requests. This is also applicable to shared mobility, car-pooling and public transport services.

High reliability on travel time estimation and prediction are mostly achieved through historical and real-time commuters' data visualization, analysis and the development of strong machine learning and deep learning models. In essence, the combination of the right statistical algorithms and tools are prioritized in predictive modeling and estimation of travel travel time (Yildirimoglu and Geroliminis, 2013). Research has shown that travel time predictions are accomplished with two major methods classified into data-driven and model-driven (Duan et al., 2016).

According to Duan et al. (2016), the data-driven approach uses data features like speed, time of the day, day of the week among others as inputs or features for developing models such as Linear regression, Autoregressive Integrated Moving Average (ARIMA), Spectral basis Neural Network, State-space Neural Network (SSNN), Recurrent Neural Network (RNN), Long and Short-term Memory Neural Network (LSTM) etc. In this research, three deep learning models are developed and trained with huge historical data sets of ride requests. Simultaneously, the model performance and accuracy are also compared to determine which is more suitable for travel time predictions for on-demand mobility.

1.2 Research background

On-demand mobility and the sharing economy require establishing reliable matches between the supply and demand aspects of transportation to ensure efficient travel for commuters and profitable business models for operators in the mobility industry. Travel time is an essential indicator for achieving the on-demand services and its prediction can be accomplished using the data-driven approach, either through historical or real-time data acquisition.

The challenges of traffic congestion caused, on most occasions, as a result of high motorization on the roads and uncoordinated supply of public transport to fulfill the travel demand of commuters, have necessitated high requests for demand-responsive mobility alternatives to complement the conventional public transport system. These problems also influence the decision-making on the travel of individual commuters and services rendered by transport operators. Furthermore, route choice and departure time are found to be important travel decisions that are influenced by travel time information, which is an important performance measure in balancing these inadequacies in transportation supply and demand (Yildirimoglu and Geroliminis, 2013).

The high accuracy required of the travel time information prediction, which is dependent on developing suitable machine learning or deep learning models, alleviates commuters' anxiety and the uncertainty about their on-demand trip decisions that can also be influenced by the departure time, origin-destination and route choice (Khosravi et al., 2011). Therefore, attaining high predictive accuracy requires using relevant indicators affecting travel time to train models and evaluate the models' performance to ascertain the most suitable among them. These indicators serve as model inputs which play huge roles in shaping the models to reach highly accurate output of travel time prediction, meant to be as close to the real individual travel time as possible.

1.3 Objectives and research questions

1.3.1 Objectives

In this research, various deep learning models are developed and compared in order to accurately predict the travel time for individual on-demand trips. This goal is achieved with the following objectives stated below;

1. Identify relevant data-driven indicators influencing the travel time of individual on-demand trips.

2. Develop and compare suitable deep learning models to predict the travel time of individual on-demand trips.

3. Identify the computational costs of developing suitable deep learning models from huge datasets to predict the travel time of individual on-demand trips.

1.3.2 Research questions

The following research questions were answered in the course of this study.

1. What are the relevant data-driven indicators that influence the travel time of individual on-demand trips?
2. Which deep learning model is most suitable to predict the travel time of individual on-demand trips?
3. What are the computational costs of developing suitable deep learning models from humongous datasets to predict the travel time of individual on-demand trips?

1.4 Research framework

The framework of this research serves as guide during the research process and it highlights the composition of each chapter in this study. The first chapter comprises of introduction, research background, research objectives, research questions and the research framework. The second chapter reviews previous research works to determine the research gaps that can be explored in this research. The third chapter reveals the methodology adopted for this research which comprises of the research strategy, data collection approach, data analysis framework and research limitations while the fourth chapter delves into visualization, analysis and discussions of the results derived during the study. The fifth chapter summarizes this thesis and draws a conclusion, and finally gives recommendations for subsequent research. Figure 1 shows the main structure and flow chart of this research work as highlighted above.

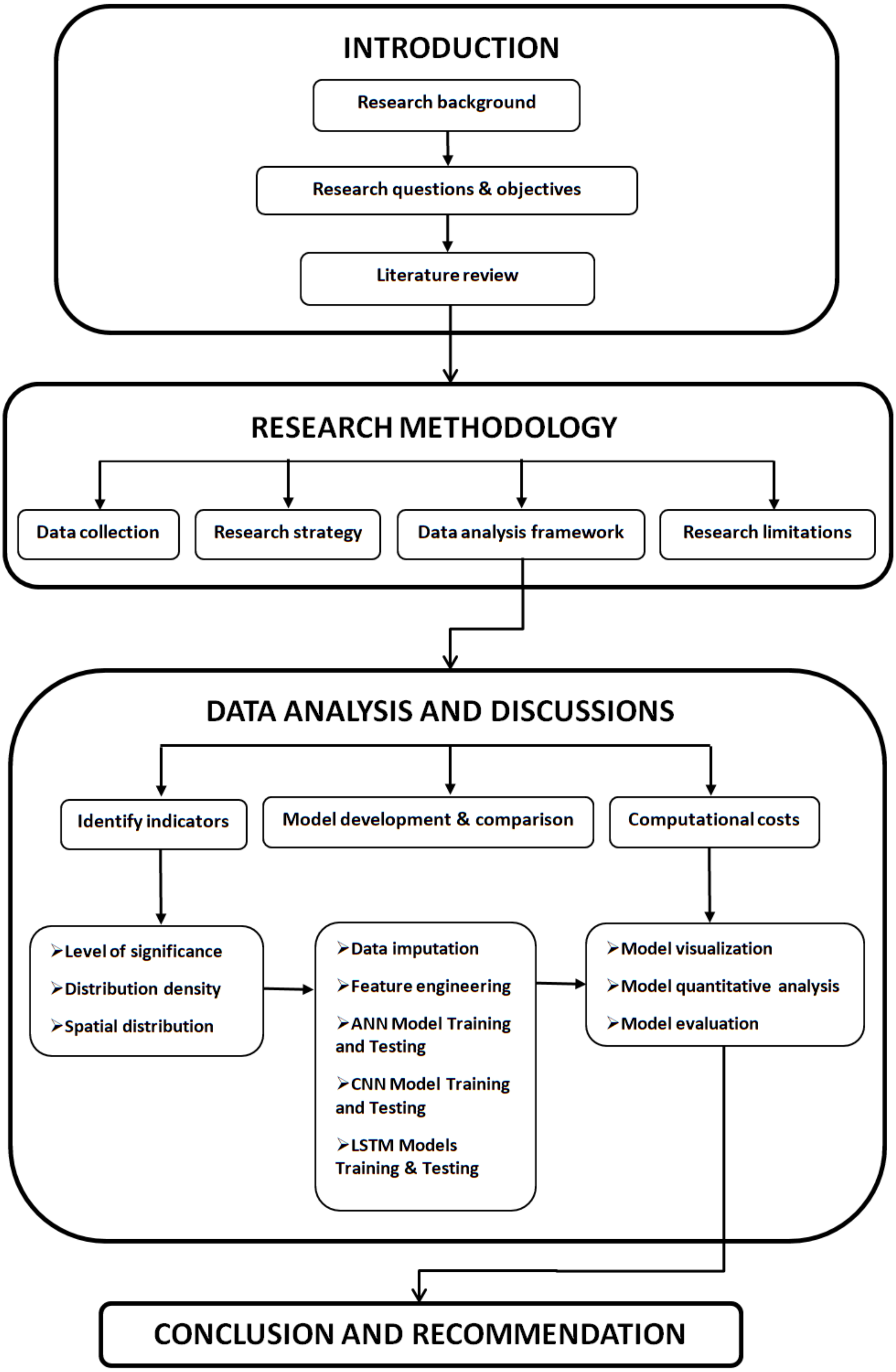

Figure 1: Research framework flow chart

Chapter 2

Literature review

2.1 Mobility-On-Demand (MOD)

Public transport have faced issues related to reliability, inconsistencies in operation schedules, delays at stops and stations and congested carriages during peak periods among others, thus leading to travelers' shift into other mobility alternatives (Liyanage and Dia, 2020). Liyanage and Dia (2020) also discovered that these alternatives, which includes ride-sharing and car-pooling, are services tailored towards complementing public transport, and catering for transport users' demand-responsive requests facilitated by technological advancement. These services are famously known as on-demand mobility and the evolution of the digital age have rapidly increased the usage of these alternative services.

On-demand mobility, also known as Mobility-On-Demand (MOD), is highly decisive in balancing process of the supply and demand systems of transportation to ensure an efficient performance of its ecosystem. The sustainability of the demand and supply coordination in the ecosystem is supported by stakeholders and enablers who includes the government at all levels, public transit agencies, operators and logistics providers, apps and mobile service providers, the users or commuters among others (Shaheen and Cohen, 2020).

This chapter reviews literature and previous studies on on-demand mobility modeling and prediction of individual trips. Relevant indicators affecting the travel time and the impacts of the travel time on individual on-demand trips are reviewed. Various deep learning models adopted by previous researchers to predict travel time are also reviewed with the approach used in deriving and evaluating the most suitable model for prediction.

2.2 On-demand trips and travel time effects

In recent times, the requests for on-demand or demand-responsive mobility have rapidly increased and this had been disrupting and reshaping the transportation system to be highly efficient. As against the conventional transport systems, it is discovered that adapting transport supply to demand-responsive requests by commuters would alter transport operations through flexibility in route choice, deviations from fixed stops and timetable flexibility (Viergutz and Brinkmann, 2015).

The aforementioned stakeholders in the mobility-on-demand ecosystem prioritize efficiency, influenced by various factors, to meet the requests for on-demand trips from commuters. Travel time is one of these factors capable of influencing individual mobility. According to Oh et al. (2015), it is a performance indicator which, when predicted, becomes tool for the comparison of various traffic management techniques. Furthermore, the accuracy of travel time can also improve the traffic management system's (TMS) ability to establish proactive responses to traffic situations (Oh et al., 2015).

2.2.1 On-demand trips and the cycle

In 1970, Torsten Hagerstand, a geographer at the Lund University in Sweden postulated theories to reveal that a set of social and personal constraints limit individuals' activities which can only take place at different spatial points in form of a time-space prism (Chu et al., 2012). These theories focus on trips based on activities which necessitate the on-demand trip request of individuals. Furthermore, Chu et al. (2012) noted that not much research had been conducted to create trip modeling approaches which utilize temporal and spatial features, thus, showing previous researchers only consider activity time and duration but not the spatial conditions.

Spatial factors such as space and distance are considered for modeling on-demand trips as a result of the need for door-to-door flexibility and dynamic route choice to satisfy the user's request. Trips requested on demand are characterized by the need to complement the conventional public transport and private transport, thereby ensuring it is a sustainable and economically attractive option for travel (Liyanage and Dia, 2020). It was however discovered by Liyanage and Dia (2020) that technological advancements borne by the digital age have established the mobile app-based approach for on-demand mobility services and personalized trip request procedures.

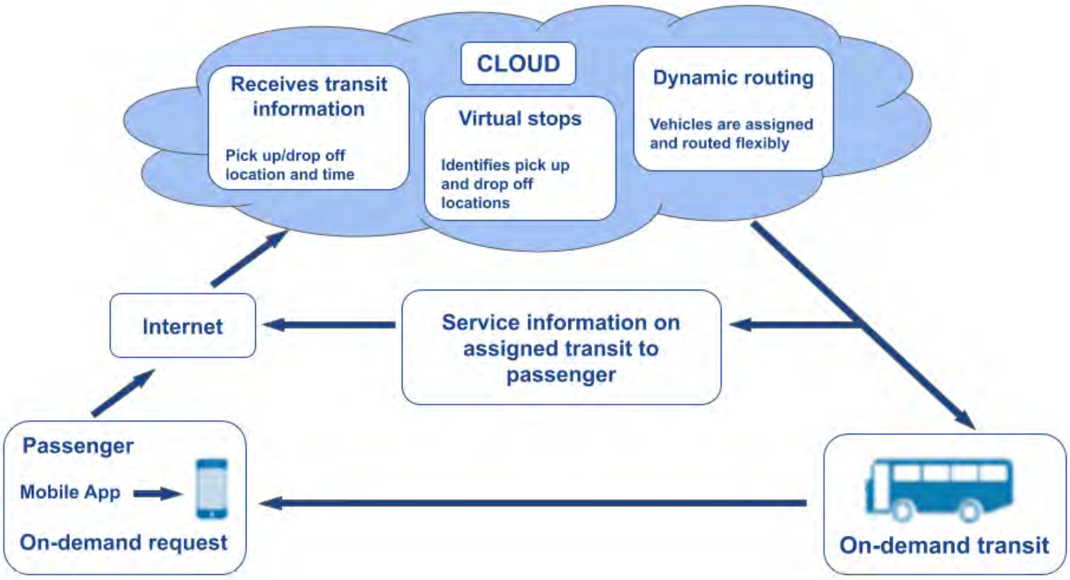

Figure 2: On-demand transit request flow, adapted from (Liyanage and Dia, 2020)

Figure 2 shows how emerging technologies like Internet of Things (IoT), mobile apps, cloud computing and Artificial Intelligence (AI) are influencing on-demand transit as against the conventional style of phone bookings and manual trip scheduling (Liyanage and Dia, 2020). The transit or trip requests originate from the passengers on demand through using mobility service providers' mobile applications, to send their orders for ride to the transport or mobility operator. These on-demand service requests are sent and received using internet connection by service providers which eventually get through to the cloud, where their transit information (such as pickup and drop off time and location etc.) are processed and sent back to passengers as notifications.

According to Hazan et al. (2019), passengers, afterwards, proceed to a nearby location referred to as virtual bus stops where they are picked up, conveyed with other passengers for pick-up and finally dropped off at locations nearest to their destinations. On-demand transits goals include, providing solutions for current gaps in public transportation, irregularities in current mass-transit maps as well as first and last mile to bus stops in the proximity (Hazan et al., 2019). Therefore, on-demand transit system bridges the gap in services and limitations of public transport and private ones as described in Figure 3 below.

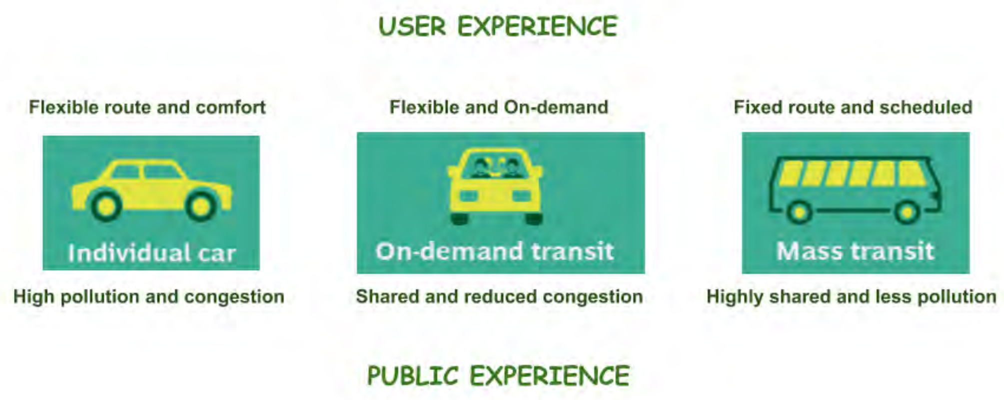

Figure 3: On-demand transit complementing public and private transport (Hazan et al., 2019)

2.2.2 Demand-responsive ride-sharing and e-hailing transits

Flexible hire services in Germany, as researched by Liyanage and Dia (2020), to accommodate short distance trips through the BMW Drive Now scheme which is a similar approach of the city driving on-demand initiative of Ford. Furthermore, these initiatives allows one-way trips as well as the pay-by-the-minute services and similar attractive service options have strengthened collaborative mobility which has seen 70% of Autolib (A French car sharing operator) users look beyond ownership of a car (Liyanage and Dia, 2020).

Electronic ride hailing services such as Didi and Uber are among many others operating on-demand services. However, their operations are not limited to sharing or car-pooling, they also offer personalized demand-responsive services for commuters. Hu (2019) further stressed that platforms for ride-hailing services match drivers in close proximity with their private vehicles to pick passengers up and most importantly ensure shorter waiting time and shorter trip distance are achieved in service deliveries to the commuters.

Ride-sharing can, however, be categorized into four different system variants according to a research by Agatz et al. (2012) in Table 1 below, which includes drivers offering a ride to a single rider or willing to take multiple riders and vice versa, riders willing to share trips with single driver or multiple drivers connected with transfers on the route to their destinations.

This ride-sharing system categories as studied by Agatz et al. (2012) recognizes the importance of time feasibility in the match of drivers and riders travel. As in the case of a single driver sharing trip with at most a single rider, vehicle miles traveled by both can be cut by maximum of 50% as a result of no transfer which saves travel time (Agatz et al., 2012). Furthermore, in the case of a necessity for round trip planning, a rider, on one hand, would

Table 1: Ride-share variants (Agatz et al., 2012)

	Single Rider	**Multiple Riders**
Single Driver	Matching of pairs of drivers and riders	Routing of drivers to pickup and deliver riders
Multiple Drivers	Routing of riders to transfer between drivers	Routing of riders and drivers

request multiple drivers and on the other hand, if the drivers also have flexible time, sharing trips with multiple riders could be possible (Agatz et al., 2012).

2.2.3 Impacts of travel time on on-demand trips

Travel time immensely contributes to demand-responsive trips, as the goal of every commuter is to move from one location to another without delay caused by congestion or waiting time at bus stations. Travel time has less negative impact on motorized transport than on non-motorized transport mode while its influence on drivers and riders in motorized transport also differ (Ennen and Heilker, 2020). Since this research focuses on motorized on-demand trips, we can deduce that travel time is an important factor to be considered in demand-responsive trips.

As shown in Table 2, Ennen and Heilker (2020) explained further that a large coefficient for drivers was discovered in absolute terms during their research and this could indicate drivers experience fatigues while in traffic congestion. A result of such large coefficient is longer travel time and virtually all transport operators and commuters envisage lower travel time before their trips start. A study in Germany by Ennen and Heilker (2020), disclosed that to ensure travel time savings, the average willingness to pay (WTP) by drivers and passengers are 5.32 Euros per hour and 3.11 Euros per hour, respectively. This denotes both drivers and riders consider travel time reduction impact on a trip to be highly important.

According to a study by Ennen and Heilker (2020), travel time influence discovered as a result of trip characteristics such as the coordinates of trip origin and destination, and the travel time are extracted from the trip's route planning service and mode options. We can infer from this research that travel time influences on-demand trips along with other features such as origin and destination.

Table 2: Model Parameter Estimates (Ennen and Heilker, 2020)

Modes: Car/Motorcycle (1), Taxi (2), Public transport (3), Cycling (4), Walking (5)

Variable (applies to modes in parentheses)	Coefficient	Standard error
Mode characteristics		
Price (1-3)	−0.392**	0.030
Population density (1)	−0.177**	0.013
Travel time driver (1)	−0.035**	0.006
Travel time passenger (1-3)	−0.020**	0.003
Travel time cycling (4)	−0.137**	0.006
Travel time walking (5)	−0.126**	0.003
Number of transfers (3)	−0.028	0.076
Headway (3)	−0.003*	0.002
Mode constants (baseline: walking)		
Car/Motorcycle (1)	−0.347**	0.075
Taxi (2)	−4.103**	0.223
Public transport (3)	−2.793**	0.102
Cycling (4)	−1.499**	0.079
Nested logit parameter (motorized) (1-3)	0.195**	0.032
Trip characteristics		
Shopping (1, 2)	0.304**	0.058
Leisure trip on weekend night (1)	−0.110	0.231
Rain/Snow (4)	−0.706**	0.120
Winter (4)	−0.803**	0.074
Traveler characteristics		
Car unavail. (1)	−1.090**	0.138
(Car unavail.) x (Work/School trip) (1)	−2.172**	0.243
(Leisure avail.) x (Travel time) (1-5)	0.017**	0.002
Driver(s) older than 64 (1)	−0.398**	0.072
Walking impairment (1, 2)	0.574**	0.148
Gender constants (female=0, male=1)		
Car/Motorcycle (1)	0.003	0.073
Public transport (3)	−0.305**	0.092
Cycling (4)	0.117	0.086

Notes: * Significant at the 5% level, ** Significant at the 1% level.

2.3 Travel time and its indicators for on-demand trips

2.3.1 Travel time description

There are three methods which Duan et al. (2016) stated in their research work for detecting or gathering data on travel time and they include the use of vehicle data collected from Global Positioning Systems (GPS), estimating base on vehicle occupancy and speed, as well as using Automatic Vehicle Identification (AVI) at end stations. As shown in Figure 4 (Xt) is the travel time for a complete journey (L) that started at time (t) from passenger's origin (A) to destination (B) which is a simple illustration of how travel time can be measured.

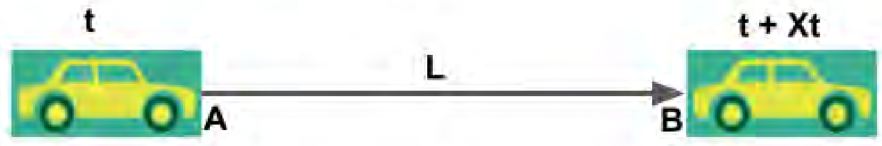

Figure 4: Travel time illustration adapted from (Duan et al., 2016)

Travel times are on most occasions considered as the timestamps at stops which differs from services of on-demand mobility known for estimating the travel time of every segment or leg on the path between two stops (Kayikci, 2018). These two stops can be deduced to be origin and destination of the passengers or the pick up timestamp and drop-off timestamps. The duration of travel between these stops have some indicators that determine or influence how much travel time is expected during the journey.

2.3.2 Indicators or factors influencing travel time

Timestamps, longitudes and latitudes are discovered to be sequence of triple collected with the aid of technologies like Global Positioning System (GPS) to discover zones or regions where geo-referenced objects or moving objects with set of trajectories, frequently visit (Masiero et al., 2011). Subsequently, the research also elaborated that trip coordinates represented with longitude and latitude are important indicators that influence travel time. It was confirmed in another research that the process of estimating travel time can independently undergo time-sliced subsets of trips having pick-up and drop-off with their respective longitudes and latitudes (Vazifeh et al., 2018).

Furthermore, Masiero et al. (2011) emphasized that variables such as longitude and latitude of vehicle's originating point and destination, distance of travel between the origin and destination, start timestamp hour of the day, start timestamp date, start timestamp day of the week (e.g. Monday), start timestamp day of the month and vehicle identifier, are indicators for travel time estimation. We can also deduce that traffic analysis zones or regions of interest are also important indicators that influence travel time but not used in the research for the model used in predicting travel time.

According to a study by Sathya Prabha and Mathew (2013), the research model estimation utilizes distance traveled by vehicle, speed during a time interval and the volume during the time interval as independent variables. These variables are also known to be indicators of travel time as they either have high negative or positive correlation with the travel time. Table 3 below shows the speed of the vehicle to have negative correlation with the negative coefficient and tStat, and it is also the most important indicator which when travel time reduces, the vehicle speed increases and viceversa (Sathya Prabha and Mathew, 2013).

Table 3: Travel time modelling (Sathya Prabha and Mathew, 2013)

Parameters	Explanation	Coefficients	Standard Error	t Stat	P-value
Constant	Intercept	50.970	9.989	5.103	0.000
Length	Distance travelled by vehicle (m)	0.116	0.141	0.820	0.000
Speed	Speed during the 15-minute time period (km/h)	-2.060	1.117	-1.845	0.000
Volume	Volume during the 15-minute time period (PCU/h)	0.022	0.102	0.218	0.035
R Square	0.997				
Observations	48				

The features or indicators of travel time are most time gathered in large amount from diverse sources, such as open data platforms, where the data had been structured and classified by experts with General Transit Feed Specification (GTFS), both in real-time and static collection (Wu et al., 2020). It was further revealed in Figure 5 below, that the data were collected from two sources which was subsequently cleaned and structured in data fusion. Consequently, exploiting logical reasoning from the data to apply into deep learning models, leads to the stage of knowledge base before finally extracting the relevant features for the model to predict the travel time.

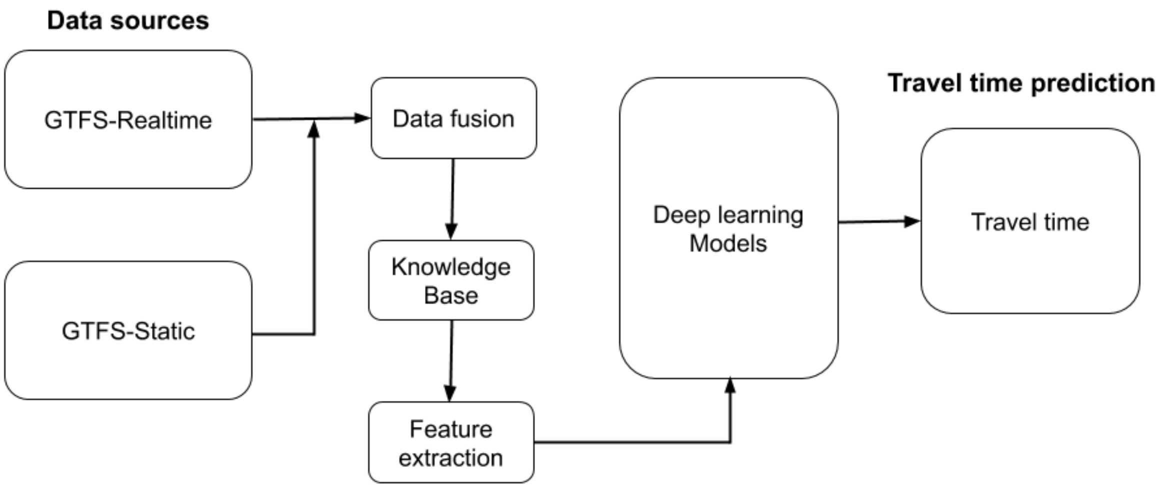

Figure 5: Framework of travel time prediction and data sources, adapted from (Wu et al., 2020)

Conclusively, most of the research works examined considered indicators or variables, including distance, speed, start time hour, start time day of the week, start time day of the month, start time date, pick-up longitude, pick-up latitude, drop-off longitude and drop-off latitude, as having influence on travel time and also used as features to develop models for travel time prediction.

2.4 Deep learning models for travel time prediction

Travel time estimation or prediction is of high importance to both passengers and operators which had made virtually all electronic maps and ride-hailing service like Didi, Google Map, Uber among others to provide estimation of travel time in their mobile applications for seamless user experience (Wang et al., 2018). According to Siripanpornchana et al. (2016) and Liu et al. (2017), time-series models such as Autoregressive Integrated Moving Average (ARIMA) is a widely utilized statistical prediction approach for travel time and relies on an assumption that predicting future values depends on random noise and historical data collected.

Subsequently, deep learning model approach became commonly used for travel time prediction. Deep learning methods is a branch of machine learning research that has solved various challenges of classification and regression in modeling and prediction (Liu et al., 2017). Neural Network models such as Artificial Neural Network (ANN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), Long Short-term Memory (LSTM) and Convolution Neural Networks (CNN) are widely and commonly used across many fields, thus, previous research were reviewed.

2.4.1 Artificial Neural Network (ANN)

Complex non-linear data challenges are resolved using the Artificial Neural Network (ANN) and ANN is implemented through two phases called the learning phase and recalling phase (Amita et al., 2015). Amita et al. (2015) further revealed that weights are assigned to models and they are trained during the learning phase while the application of weight assigned during learning phase is also applied to the recalling phase. ANN is also a type of machine learning method with the capacity to solve noise and complex data within large sets of data considering complexity of the relationships between their predictors (Gurmu and Fan, 2014). These are the basic applications and phases of Artificial Neural Network (ANN) model and it can be applied to the prediction of travel time for individual trips.

An advantage of ANN model is being able to predict travel time without full consideration of the physical traffic processes as it imitates the ability of human brains to process data in an intelligent way (Gurmu and Fan, 2014). Another important aspect of ANN for travel time prediction is its spatial-temporal input data or the independent variables used to train the model, which have very high significance in the overall outcome or accuracy of the model (Zeng and Zhang, 2013). We can deduce from this that spatial and time-dependent variables discussed as travel time indicators in previous section are very important in the model performance.

Artificial Neural Network (ANN) Architecture

Artificial Neural Network model development has similarity with the development of regression models, with a division of the whole data set into training and testing sets of data (Amita et al., 2015). The training data is what the model uses to learn patterns in the data to be able to predict while the testing set is used to check the performance or accuracy of the trained model. The structure of ANN basically contains nodes and three layers interconnected together, thus having the first layer as an input containing information like independent variables, last layer having output information and in between is a hidden layer which generates weight and bias parameter during training (Gurmu and Fan, 2014).

The quantity of nodes that are contained in the hidden layer are to be determined through experiments (Gurmu and Fan, 2014). Amita et al. (2015) further disclosed that ANN comprises of highly interconnected processing elements which are of huge size and also in resemblance with the biological nervous system like the brain which has information processing elements also known as neurons. As a result, these neurons are organized into layers and interconnected patterns within and this form of arrangement is termed architecture of net (Amita et al., 2015).

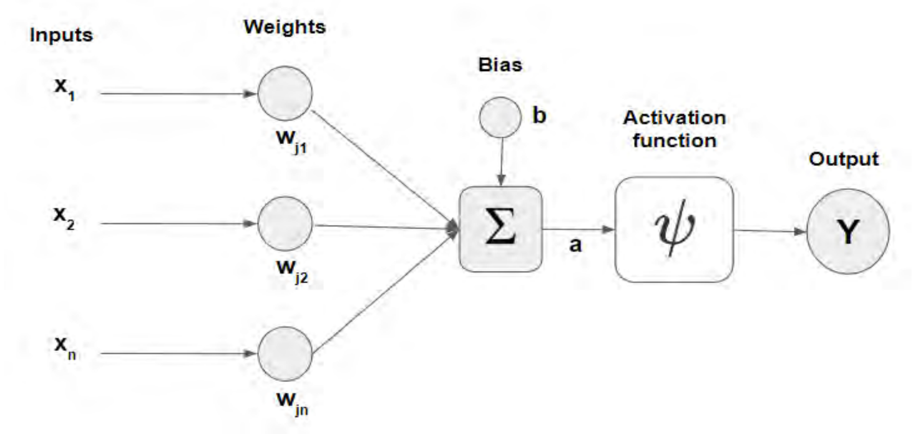

Figure 6: Structure of ANN neurons, adapted from (Gurmu and Fan, 2014)

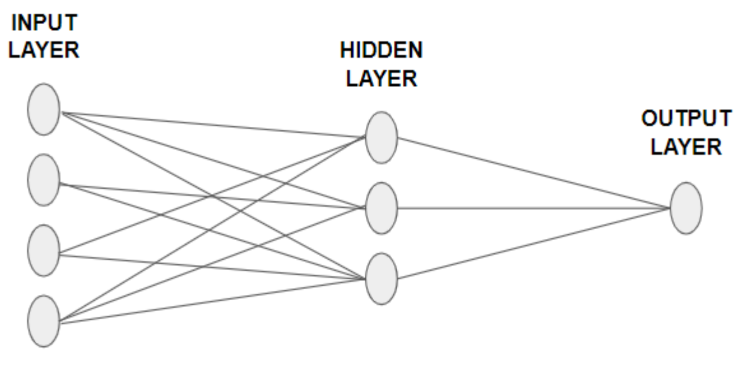

Figure 7: Structure of ANN layers, adapted from (Gurmu and Fan, 2014)

Figure 6 above shows the network architecture of the neurons as described in equation 2.1, 2.2 and 2.3, as well as Figure 7 displaying the basic network topology of the Artificial Neural Network layers for Multi-layer Perceptron (MLP) with one hidden layer. Amita et al. (2015) revealed that the number of processing elements (neurons) in the hidden layers relies on how complex the problem is and the decision is made through trial and error. We can deduce that both the hidden layers and the hidden neurons in each hidden layer use trial and error approach to determine their quantity within the network. Furthermore, non-linearity between input and output of the network and nodes is usually incorporated with the aid of the activation function, while improving weights to reduce mean squared error employs a training procedure called back propagation on many occasions, such as transport-related applications (Gurmu and Fan, 2014).

According to Equation 2.1 as described by Gurmu and Fan (2014), the Artificial Neural Network (ANN), which has a function signal at the output layer of neurons j resulted in the computation of the first equation. Equation 2.2 shows the output equation of the neural network architecture just before the activation function and Equation 2.3 is the formal or the widely accepted equation for output of the architecture.

$$Y_j = \psi(X_1, X_2, X_3, \ldots, X_n) \tag{2.1}$$

$$a = \sum_{i=1}^{m} w_{ji} X_i + b_j \tag{2.2}$$

$$Y_j = \psi_j\left(\sum_{i=1}^{m} w_{ji} X_i + b_j\right) \tag{2.3}$$

Where:
m = the number of inputs used as neuron j
X_i = the group of input variables of neuron j
Y_j = the output of the j^{th} neuron
w_{ji} = the weight linking the input (i) to the neuron (j)
b_j, ψ_j = the applied bias term and the adopted activation function, respectively.

There are four stages of back-propagation network discussed by Amita et al. (2015) and they include, Weights initialization, Feed Forward, Back-propagation of error signals and lastly, Weights and biases update. The four stages in the respective stated order have their first stage to initialize weights to random values with small sizes, the next stage receives input information and thereafter transmit through the hidden unit to the output unit. The following stage back-propagates the error signal to previous layer when a desired output is not reached while the last stage updates the weights and biases in respect to the error signal. In essence, these stages are the process undergone to train a model in order to get an accurate output using Artificial Neural Network model.

2.4.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN), vastly applied to model image classification, video processing and tracking objects, is a standard approach for modeling spatial dependencies which is highly significant for the prediction of travel time using the Global Positioning System (GPS) sequence (Wang et al., 2018). CNN with multi-layers is an excellent neural network for traffic prediction with high precision and performance, and most importantly it is a powerful technique useful for short-term travel time prediction (Wei et al., 2018). In essence, CNN is a deep learning technique for developing models capable of reading patterns and trends in spatial, and temporal data with very high prediction accuracy and model performance.

Using the longitude and latitude of corresponding vertices to sort out an adjacency matrix columns and rows respectively, Convolutional Neural Network model assumes there are relationships between the cells closed in the input tensor of the adjacency matrix (Bogaerts et al., 2020). Bogaerts et al. (2020) further explained that these closed cells represent a real road network with close road links and with this approach it is easy to arrive at spatial patterns that are valuable for the model development. Those are important steps to undergo during the pre-processing work flow before training the model for prediction.

Convolutional Neural Network (CNN) Network Architecture

A research conducted by Ran et al. (2019b) described the architecture of a Convolutional Neural Network (CNN) used in training models for spatio-temporal large data sets to predict travel time comprise of input sequence representation, local receptive fields, convolutional max-pooling layer, fully connected layer using a nonlinear activation function and finally an output predictor. Figure 8, however, shows the architecture of CNN which was extracted from the research of Ran et al. (2019b) and it contains the input, convolution layer having rows of matrix identified as C_k, max-pooling layer represented with m, the flattening layer, global-level feature at the fully connected layer and finally the output.

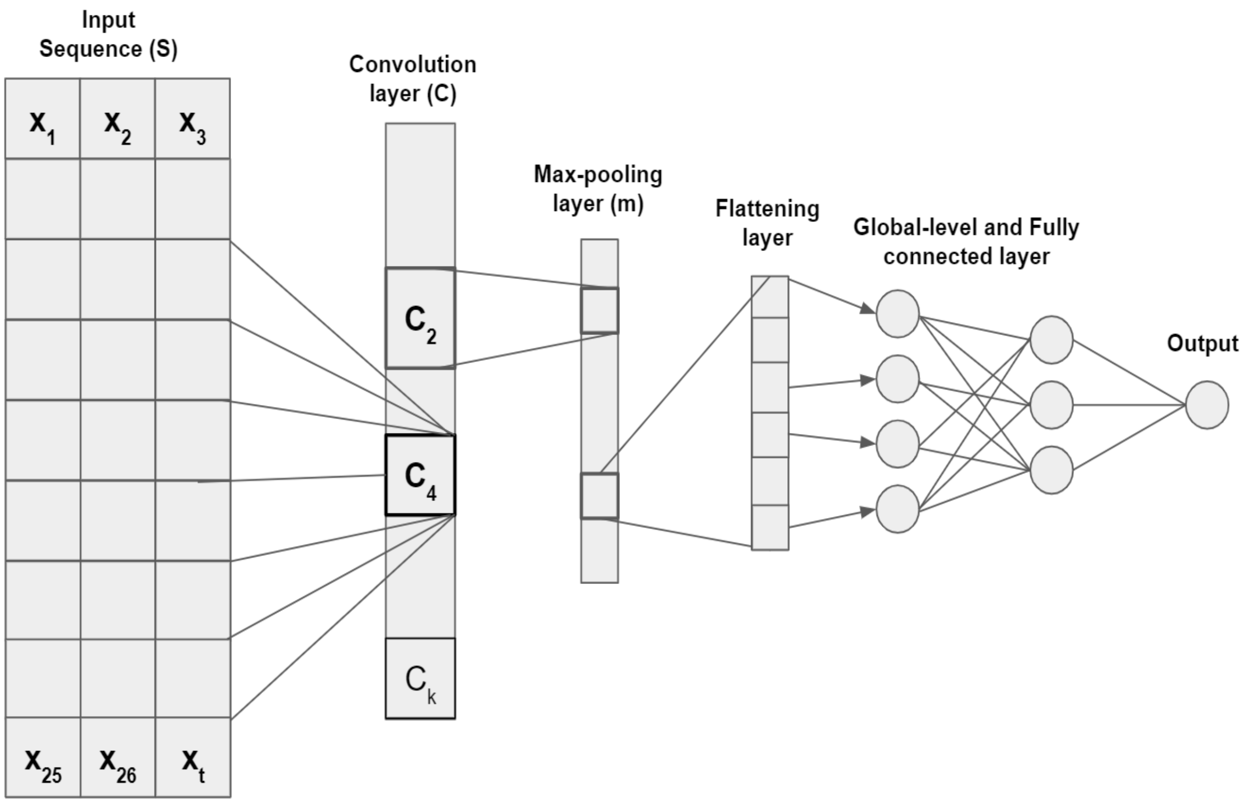

Figure 8: Convolutional Neural Network (CNN) Architecture for spatio-temporal data of travel time prediction, adapted from (Ran et al., 2019b)

Input sequence layer, as shown in Figure 8 comprise of several spatio-temporal inputs represented with (S) and they were generated from traffic data related to the values predicted while the max-pooling convolution identified the input sequence's relationships with the predicted values (Ran et al., 2019b). Ran et al. (2019b) further explained that before global-level features were provided to a predictor to generate predictions, the fully connected layer was used together with activation function to capture the nonlinear relationship and to extract global-level features. The equation formula adopted in processing these Convolutional Neural Network (CNN) architecture are further shown as follows for each layer.

$$S = x_1, x_2, x_3 \dots x_t \tag{2.4}$$

$$C(k) = W_k X(j), \quad 0 < j \leq t, \quad k \in (1, \dots, n_1) \tag{2.5}$$

$$m_i = \max C(i;.), \quad 0 \leq i \leq n_1 \tag{2.6}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{2.7}$$

$$z = \tanh(W_f\, m) \tag{2.8}$$

$$\hat{y} = W_f\, z \tag{2.9}$$

Where:
S = the sets of input sequence from the traffic data
X_t = the spatio-temporal input values
C(k) = k^{th} row of matrix C, a convolution result of convolution kernel W_k for one input sequence
W_k = one convolution kernel
X(j) = j^{th} item of vector X
n_1 = an hyper-parameter representing the number of convolution kernels
m_i = the max function of max-pooling layer m
C(i;.) = the i^{th} row of matrix C
tanh = the hyperbolic tangent function
z = the final global-level feature
W_f = the parameter matrix for the fully connected layer
ŷ = the travel time through the given link within a time interval

Equation 2.4 shows the sets of input sequence extracted from the data sets and it has a length of t used in subtraction. Equation 2.5 represents the convolution outcome of the convolution layer, Equation 2.6 addresses the selection of better features using max function to get results from the max-pooling layer while Equation 2.7 is the activation function formula for the model to capture the non-linear relationship. Equation 2.8 is the formula for the global-level feature after max-pooling process and this equation is sent to the predictor for the prediction outcome using the equation 2.9. All these equation descriptions are according to Ran et al. (2019b) research.

Another research conducted by Wei et al. (2018), adopted a different formula, but with similar prediction output to Equation 2.9, which is used for time shifting features and 2D spatial-temporal extraction.It has two layers of convolutional network and one layer for pooling. It was further noted that the convolutional layers' input and output are feature maps whose k^{th} feature map at a particular layer is represented as H^k, the weights W^k and bias b_k determining the filters and the f as ReLU function whose role is to solve the problem of gradient vanishing, as in Equation 2.10 below (Wei et al., 2018). This research also combined the network of both CNN and the Long Short-term Memory (LSTM) to arrive at an output.

$$h^k = f(W^k \cdot x + b_k) \tag{2.10}$$

Where:
h^k = k^{th} feature map at a particular layer
f = Rectified Linear Unit (ReLU) function
W^k = Weights of k^{th} feature map at a particular layer
b_k = bias of k^{th} feature map at a particular layer
x = inputs

Wang et al. (2018), however, had different argument on the 2-dimensional CNN that the coordinates from Global Positioning System (GPS) mapped into grid cells does not have the accuracy to represent the real spatial information in the data. Therefore, a different approach called Geo-Conv layer was adopted to capture spatial dependency in the sequence of geo-location and the information is retained in finer granularity (Wang et al., 2018).

Geo-convolutional neural network is a spatio-temporal component that does the transformation process of turning the sequence from raw GPS into feature map series for the purpose of capturing local spatial correlation between continuous points from the GPS (Wang et al., 2018). We can deduce so far that CNN models, on many occasions, were used in previous research works for the prediction of space or location related transport model development. It has also been discovered that CNN is sometimes in previous research works merged with a Recurrent Neural Network (RNN) called Long Short-term Memory (LSTM) to develop models for predicting temporal or time-related information.

2.4.3 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is another method of deep neural network that flexibly deals with sequence data and automatically captures from large amounts of data the time series features comprising of semantic information such as complex temporal dependency (Sun et al., 2020). Since traffic management and mobility data also have time-series related information like travel time, RNN is usually applied to build the models for accurate predictions.

Initially, the conventional prediction models encounter challenges of extracting time mode information of travel time which made it also problematic to train longer travel time series (Wei et al., 2018). In order to resolve these drawbacks and also reduce the exploding and gradient vanishing problem, Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) network were introduced (Sun et al., 2020). These two RNN methods have been used by researchers for travel time estimations and other time-series related problems. Therefore, previous research works on the LSTM neural network will also be reviewed subsequently.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a powerful RNN method for time series estimation which uses historical data as a result of its ability to capture relevant information from historical input sequence (Wei et al., 2018). Just like LSTM is a strong tool for short-term time series prediction, Agafonov and Yumaganov (2019) discovered that it can also handle long-term dependencies with its ability to utilize three gates for information control in its cell for every time steps. These gates are input, forget and output contained in the structure of LSTM cell disclosed in the research of Agafonov and Yumaganov (2019).

Long Short-Term Memory (LSTM) Network Architecture

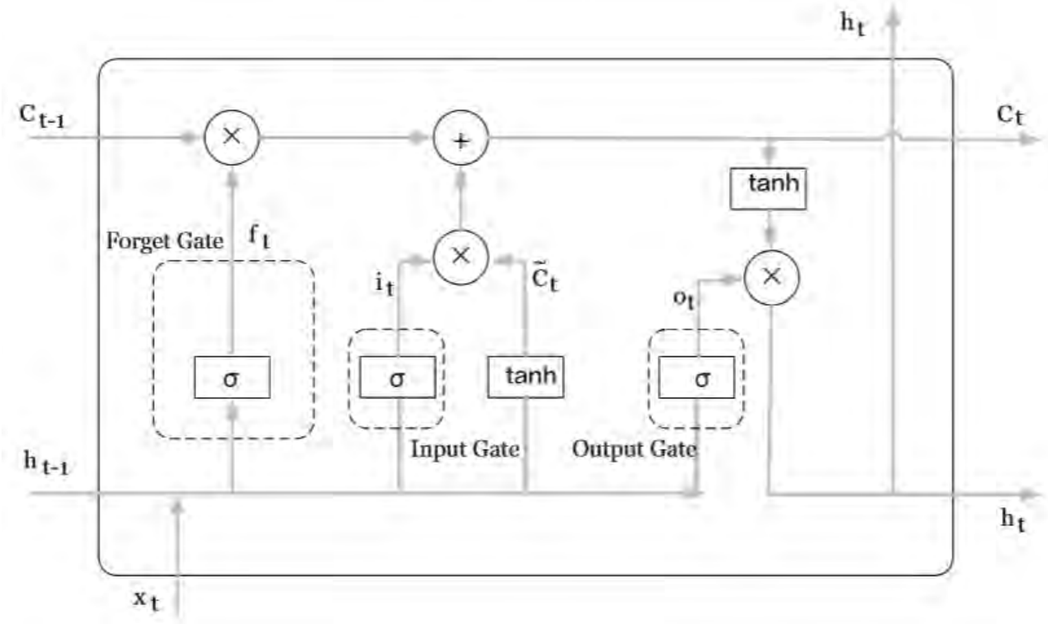

Figure 9: A Structure of LSTM Cell (Wei et al., 2018)

According to a research by Wei et al. (2018), Figure 9 above shows the structure of Long Short-Term Memory (LSTM) cell state which contains input gate, output gate and forget gate that regulate adding or removing information from the cell state. We can also deduce from this structure that the LSTM cell has input X_t with hidden layer output h_t, the three gates i_t, f_t and o_t, the former hidden layer output h_{t-1}, cell input and output state (Ct and C_t respectively) and the former output state C_{t-1}. These are the compositions of a cell structure for LSTM and they are applicable to developing LSTM models for travel time estimations.

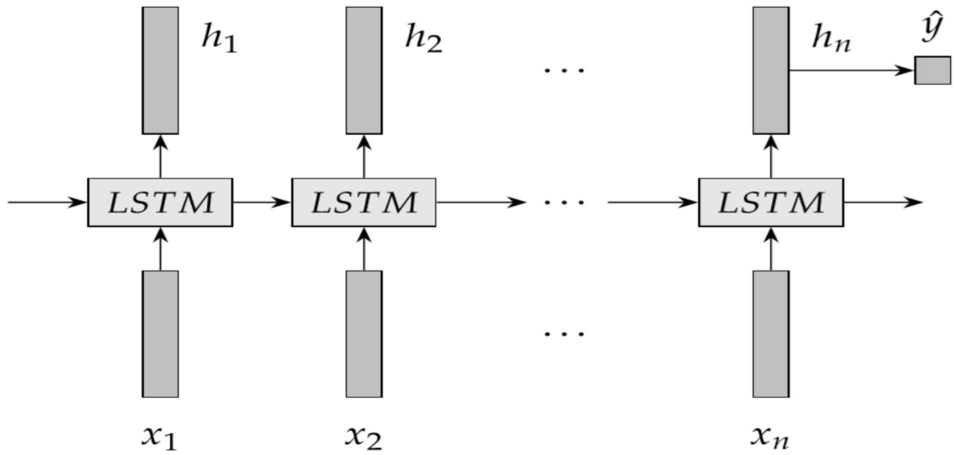

Figure 10: An architecture of LSTM Neural Network for travel time prediction (Ran et al., 2019a)

$$i_t = \sigma(W_1^i \cdot x_t + W_h^i \cdot h_{t-1} + b_i) \tag{2.11}$$

$$f_t = \sigma(W_1^f \cdot x_t + W_h^f \cdot h_{t-1} + b_f) \tag{2.12}$$

$$o_t = \sigma(W_1^o \cdot x_t + W_h^o \cdot h_{t-1} + b_o) \tag{2.13}$$

$$\tilde{C}_t = \tanh(W_1^C \cdot x_t + W_h^C \cdot h_{t-1} + b_C) \tag{2.14}$$

$$\hat{y} = W_s \cdot h_n + b_s \tag{2.15}$$

Where:
i_t = the input gate state at time t
f_t = the forget gate state at time t
o_t = output gate state at time t
C_t = the cell input state at time t
b_i, b_f, b_o, b_c = bias terms
W_1^i, W_1^f, W_1^o, W_1^c = weight matrices for the 3 gates and cell input
W_h^i, W_h^f, W_h^o, W_h^c = weight matrices connecting h_{t-1} to the 3 gates and cell input
h_{t-1} = the former output at time t
C_{t-1} = the former cell output state at time t
x_t = input at time t
h_t = hidden layer output at time t
ŷ = output predictor
h_n = output vector
b_s = bias term
W_s = Weight matrices

Equation 2.11 represents the input gate formular, Equation 2.12 is the forget gate formular, Equation 2.13 shows the output gate formular and the Equation 2.14 is the formular for cell input, all as shown in Figure 9. These gates and equations are components of a LSTM cell architecture as research by Wei et al. (2018).

As illustrated in Figure 10 and Equation 2.15, the architecture of LSTM NN for travel time prediction and output predictor equation respectively, contain input sequence X_n as travel time values which produces output vectors H_n using transition functions in order to arrive at an output predictor ŷ. LSTM network structure also have an advantage of considering, in long term prediction, the effects of transport situation on remote route links which eventually produce an output of travel time between each remaining stops (Agafonov and Yumaganov, 2019).

Evaluating the performance of the neural network models is however important to detect the accuracy of our prediction and Agafonov and Yumaganov (2019) revealed in their research that there are two standard metrics called Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Some other researchers also adopted Mean Squared Error (MSE), Root Mean Squared Error (RMSE) among many others and these model evaluation methods will be discussed in following section.

2.5 Evaluation metrics for travel time prediction models

Model evaluation is an essential part of machine learning and deep learning techniques as it informs the experts the accuracy and performance of the model for prediction. In a research conducted by Nyaki et al. (2020), the evaluation method used to measure the models developed and compared for travel time estimation was Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). It was further stated by the research that these measurements determine how much deviation the model has from the predicted travel time (Nyaki et al., 2020).

$$PMAPE = \frac{1}{N}\sum_{n=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \tag{2.16}$$

Where:
y_i = observed values for i^{th} travel time
ŷi = predicted values for i^{th} travel time

N = number of times for observed values

The evaluation metrics adopted by (Sun et al., 2020) to measure FusionRNN-ETA, a type of Recurrent Neural Network model for travel time prediction, were similar to the ones mentioned above but with one other method known as Mean Absolute Error (MAE). Previous research reviewed have shown that these four aforementioned methods are widely used in measuring and evaluating models developed to predict travel time. Therefore, we will be examining each one of them as discussed in previous works.

Prediction Mean Absolute Error (PMAE) is a performance metrics utilized to measure the outcome of travel time prediction by measuring the difference between two continuous variables (Wei et al., 2018). The research further revealed that the differences between actual values and predicted values have its sample standard deviation to be represented by Prediction Root Mean Squared Error (PRMSE) while measuring the accuracy of prediction for algorithms written to develop models for the travel time estimation is known as Prediction Mean Absolute Percentage Error (PMAPE) (Wei et al., 2018). These performance metrics are used to compare different models developed for travel time prediction to know the most suitable one, and the lower the values of PMAE and PRMSE the better.

$$PMAE = \frac{1}{N}\sum_{i=1}^{N}(|y_i - \hat{y}_i|) \tag{2.17}$$

Where:
y_i = observed values for i^{th} travel time
ŷi = predicted values for i^{th} travel time
N = number of times for observed values

Prediction Mean Squared Error (PMSE) is the foundation of Prediction Root Mean Squared Error (PRMSE), the only difference is finding the squared root of the PMSE values to get PRMSE. According to Tang et al. (2018), PRMSE is a validation metric commonly adopted to quantitatively measure the accuracy of predictions. PMSE measures the deviation between the predicted values and actual values while its root results into PRMSE.

$$PRMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_i - \hat{y}_i)^2} \tag{2.18}$$

Where:
y_i = predicted values for i^{th} travel time
ŷi = observed values for i^{th} travel time
N = number of times for observed values

Another evaluation metric for travel time is coefficient of determination (R^2) score and its values usually range between 0 to 1, the farther the value to 1 the higher the relative error and the closer to 1 the lower the relative error's degree (Hiroi et al., 2019). R-squared (R^2) is also a validation metrics and it was used in the research work of Tang et al. (2018). The Equations, 2.16, 2.17, 2.18 and 2.19 are used for measuring the performance metrics of prediction accuracy of models.

$$R^2 = 1 - \frac{1}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sum_{i=1}^{n}(x_i - \widehat{x_i})^2 \tag{2.19}$$

Where:
R^2 = coefficent of determination or correlation
x_i = the observed value or ground truth
x̂i = the predicted value
x = the average of travel time

In conclusion, this research work will be filling the gaps of previous literature by identifying more relevant indicators different from the aforementioned and commonly used data-driven indicators in machine learning and deep learning models for travel time prediction. Identifying more relevant indicators from the data to be collected will further improve model performance and the accuracy for planning, and decision-making for on-demand trips from the commuters and transport operators' end. Another gap to be filled include comparing models with set of relevant indicators as model inputs under each deep learning network (e.g. Artificial Neural Network (ANN)). And finally, comparing the models with the highest performance on each of the adopted deep learning networks. The deep learning networks to be used are Artificial Neural Network (ANN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

In addition, this research also identified the computational cost of developing the deep learning models and identifying relevant indicators, which are mostly not addressed in previous literature. These costs include identifying the computer specifications used for the model development, run-time taken for data processing and preparing each model, and the total run-time for each neural network to train and test their models for prediction.

Chapter 3

Research methodology

3.1 Introduction to methodology

This chapter discusses the research methods and approaches adopted in fulfilling the research objectives, including methods used in identifying relevant data-driven indicators affecting the travel time of individual on-demand trips, developing and comparing suitable deep learning models for the prediction of travel time and also identifying the computational costs of developing the models.

The identification of relevant indicators used some approaches adopted by the reviewed literature on identifying travel time indicators. The most important is the second objective, which involves modeling to predict travel time and compare these models, and thus requires a huge amount of data collection and analysis. After examining previous research works, some gaps discovered led to the need for this research.

Some of these gaps include using only basic features or indicators, not considering other possible combinations of indicators influencing travel time. The new features utilized in this research will be elaborated further in this chapter. Another research gap discovered in previous works is the model evaluation metrics adopted, which are the common methods used by most researchers. Also, in previous research, many utilize two deep learning models for their predictions. However, in this research work, individual deep learning technique was stand-alone evaluated and compared to know their performance.

The importance of this research work is to examine different deep learning models that can accurately predict travel time, which is useful for transport operators and individual commuters during their demand-responsive trips. The prediction also allows both parties to have ample information on their trips before embarking, and it saves costs, time, and helps in trip decision-making. Therefore, this chapter discusses this study's research methodology, which includes research strategy, data collection approach, data analysis framework, and discovered limitations of research.

3.2 Research strategy

To accomplish this work, a case study research approach was adopted using an exploratory case study method based on the research objectives. Biggam (2015) explained in a book on Master's Dissertation that case study research examines an instance of a precise type of a thing, and he proceeded with a deduction from Cohen and Manion research in 1995, which revealed case study to involve in-depth research and analysis on an individual unit's feature.

The first research objective for this thesis uses an exploratory case study to identify the data-driven indicators influencing the travel time of individual on-demand trips using the geographical location stated above. Subsequently, the second objective of developing and comparing suitable deep learning models for travel time prediction also adopts the exploratory case study method.

3.2.1 Exploratory case study and convenience sampling

Exploratory case study strategy, as its name implies, explores a large scale study of the future using research questions, which results in fulfilling the objectives (Biggam, 2015). This strategy was adopted due to the research objectives, which demands identifying data-driven indicators that influence travel time of individual on-demand trips and developing and comparing suitable models for the travel time prediction. The approach requires studying large historical data sets of on-demand trips peculiar to a particular geographic location and giving insight into the future by identifying indicators, modeling, and predictions.

In the case of this research, the exploration is on the travel time of individual on-demand trips through identifying the travel time indicators, train the data using deep learning techniques, and compare the models used for forecasting the travel time. The literature reviewed in the second chapter of this study disclosed some fundamental indicators that influence travel time, and these indicators also served as modeling features for travel time prediction.

A convenience sampling approach used for the case study area resulted from collecting demand-responsive trip datasets owned by car-pooling, ride-sharing, or mobility companies, through the internet from their electronic or online platforms. Edgar and Manz (2017) defined convenience sampling as a non-probabilistic sampling technique focusing on samples on the internet, online, or around a particular location that is more convenient for the researcher. This type of sampling method is mostly used for exploratory research to gain insights and ideas capable of resulting in in-depth analysis for a future study or representative research (Biggam, 2015).

To achieve this sampling technique, the data of on-demand ride requests, retrieved from the database of a mobility company is used in sampling commuters' on-demand trips. It also requires large open data sets with large customers or commuters, using its online platform to place orders for on-demand trips. This also demands the use of a particular geographic location with a dense urban area and a central business district.

3.3 Data collection

The data used for this study was collected from a secondary source and the online platform of mobility company called DiDi Chuxing GAIA Open Dataset Initiative. Authorization was granted by the company to use the second ring road regional trajectory data set of Chengdu City, which contains large data sets of ride requests and ride routes in the city in China. Didi Chuxing offers a wide range of mobility services to over 450 million users, and since December 2017, it had over 25 million orders for their services per day (Didi-Chuxing, 2020). The mobility services include taxi, premier, express, minibus, car rental, bike-sharing, among others, and the majority are on-demand services (Didi-Chuxing, 2020). The company also ensures collaboration through the GAIA initiative with academic communities by providing large, anonymized data sets of real-world scenarios to enhance in-depth and prospective research in transportation (Didi-Chuxing, 2020). They are the primary reasons for choosing the platform as a suitable source of data collection for this study, meant to fulfill this research's objectives.

The two large data sets collected from the DiDi Chuxing database were the ride request data (comprising origin-destination coordinates, ride start and stop time) and the Chengdu city routes' coordinate data having an accuracy of 2 - 4 seconds (Didi-Chuxing, 2016). The humongous data has a 50-gigabyte memory capacity with geographical coordinates range of latitude [30.727818], longitude [104.043333], latitude [30.726490] and longitude [104.129076], and date range from 1st November 2016 to 30th November 2016 (Didi-Chuxing, 2016). The first data set of origin-destination trips has slightly over 7 million (7,058,950) on-demand trips of DiDi commuters. The second data set comprises the trip routes for requested rides within November. It also has about 45 million (45,000,000) time-steps of 2-4 seconds accuracy, per day and summing up to 1.35 billion time-steps in the whole 30-day November in 2016.

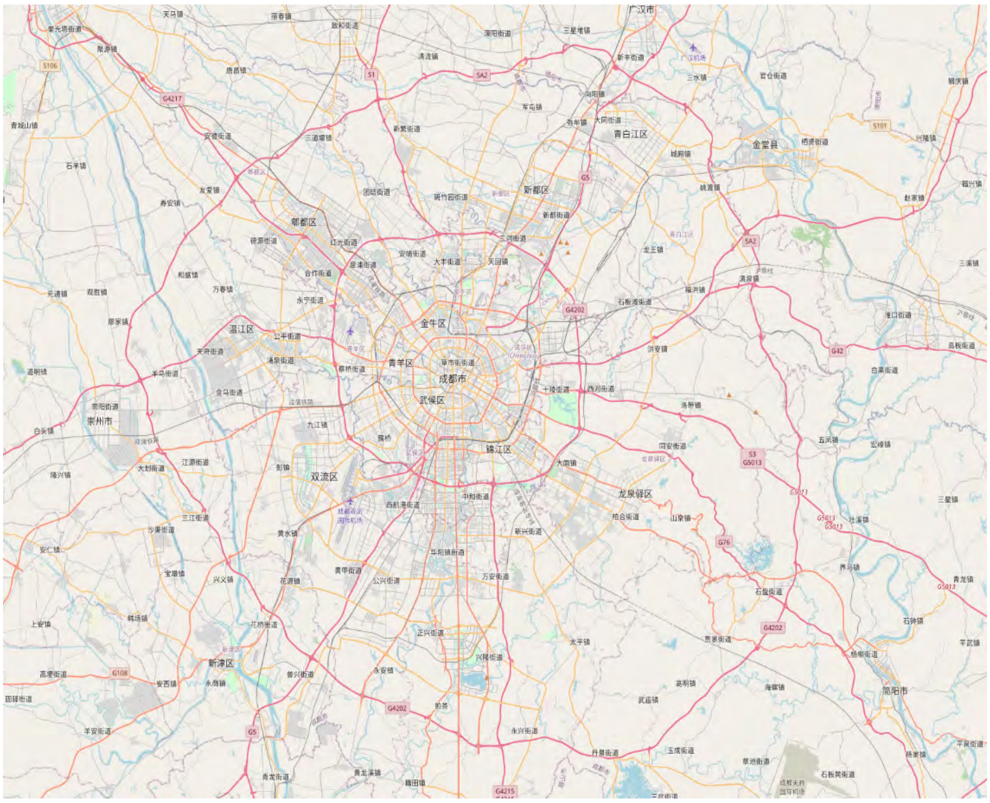

Figure 11: An Open Street Map (OSM) of Chengdu City, China (The case study area)

Figure 12: An Open Street Map (OSM) for about 1.35 billion coordinates of origin-destination trips and trip routes in Chengdu City, China (The case study area)

These data sets are essential for this study because they are suitable for fulfilling this research's objectives. The first objective focuses on identifying data-driven indicators that influence individual on-demand trips, and on identifying them, a large set of data is required. These indicators are data-driven due to being captured via the internet, thereby missing some more relevant indicators with a strong influence on the travel time, which the data could not capture. The use of the potential indicators, obtained in the raw data, to derive or calculate additional relevant ones also makes the indicators data-driven. In essence, since the second objective deals with developing and comparing models for predicting travel time, then data-driven indicators will be more suitable as inputs or features for the deep learning models.

3.4 Data analysis framework

The raw data collected comprises few features or indicators not enough to identify the primary indicators found in previous literature. The raw data used for training data contains features or indicators such as ride start time-stamps, ride stop time-stamps, pick-up longitudes, pick-up latitudes, drop-off longitudes, drop-off latitudes, and the order Id. Simultaneously, the routes' data collected every 2-4 seconds is was the test data, and it had features such as order Id, driver Id, longitudes, latitudes, and time-stamps.

Sequel to these short descriptions of the data set features and having contained numerical values that require an advanced statistical analysis approach, a quantitative research analysis method is adopted to gain insight into the datasets and accomplish the research objectives. These need an in-depth exploration, visualization, and analysis of the datasets to identify and fulfill the first objective and the second objective of this research. The origin-destination ride request data set was used to train the deep learning models, while the ride route data was utilized as the test data to measure the models' accuracy. Also, only the first seven (7) days of the test data or the ride route data were processed and used to test the models due to its large capacity and complexity to label, in consonance with the training data.

3.4.1 Quantitative analysis and analytic tools

The 'how' question mostly results in a quantitative research approach, and this kind of approach is possible when research questions jointly consider methods used in data collection and the research strategy adopted for the study (Biggam, 2015).In respect to this elucidation, fulfilling the research objectives and questions on how to identify data-driven indicators influencing travel time and how we can develop and compare deep learning models for travel time prediction led to the collection of large data sets from a secondary data source or an online database. After collecting the data for a particular geographic location, exploratory case study research was adopted, and it requires in-depth analysis and advanced statistical methods because of its quantifiable values. These are justifications for selecting quantitative research analysis to satisfy the objectives of this research.

Furthermore, to accomplish the second objective, the researcher derived more relevant indicators such as departure hours, highly significant departure hours, departure minutes, highly significant minutes of departure, departure day of the month, departure day of the week, departure month, departure traffic analysis zones, arrival traffic analysis zones, travel time in hours, travel time in minutes, distance traveled and vehicle speed. This approach satisfied the first objective of identifying indicators using an advanced statistical method that measures and determines highly significant features with their correlation values of 0.01 or 1% minimum and a maximum of 1 or 100%. The correlation values are further discussed in the data analysis chapter of this research work.

The advanced statistical tools used in the data sets analysis are Python programming language version 3.7.7 and Anaconda Jupyter notebook version 1.7.2. These tools have characteristics such as non-complexity, flexibility, and an interactive user interface. Jupyter notebook is an interactive computing environment, with two components (Web application and Notebook documentation) as an extension of a console-based platform used to develop, document and execute codes to fulfill a computational process (Jupyter-Team, 2015). There are enormous libraries and clients in Python used to explore, visualize, analyze, and develop models. These libraries, tools, and methods adopted in the analysis aided the derivation of these highly essential indicators, and the working mechanisms for the deep learning models, which are discussed subsequently in this section.

Identifying the computational costs of running the python scripts for data processing, model training, and model testing also used a quantitative analysis method. The computer specification values, model parameters, and model computation time or run-time were listed to know how much time and computational power consumed during the modeling and prediction with the large data sets.

3.4.2 Derivation of relevant travel time indicators

Departure time

To derive various departure time used in modeling, the timestamp was split into various categories, resulting in more departure time indicators. The first category was departure hour, extracted from the departure timestamp using the 'DateTime' library for python hours. The 'DateTime' library was also used to derive the week's departure day, departure month, and departure day of the month. Furthermore, a whole 24-hour day was divided into 6 periods which include; 1 AM - 4 AM (Mid-night), 4 AM - 8 AM (Morning), 8 AM - 12 PM (Mid-day), 12 PM - 4 PM (Afternoon), 4 PM - 8 PM (Evening), 8 PM - 12 AM (Night). The highly significant of these periods with correlation values greater or equivalent to 0.01 were identified as important indicators or modeling features. This approach was to identify which part of the day influences the travel time of on-demand trips. In essence, this strategy examines which departure period of the day increases or decreases the time taken to travel to the destination from the origin.

A similar strategy was adopted for minutes of travel, which involves dividing clock time of sixty (60) minutes into four (4) categories. An example of this division is a commuter who departs between 12 PM and 1 PM by identifying his or her minute of departure from 12.00 PM - 12.15 PM, 12.15 PM - 12.30 PM, 12.30 PM - 12.45 PM and 12.45 PM - 01.00 PM. For this research, the Greenwich Meridian Time (GMT) (e.g., 1 PM = 13.00) format was used. Afterward, the highly relevant of these 15-minute categories were used in modeling for travel time prediction. Departure hours of the day with highly significant values also served as features for the models. This division's essence is to know if departing during peak hours or non-peak hours have an impact on travel time or departing at a particular 15-minute period of each hour affects the travel time in the geographical location. The highly significant of these new indicators were selected base on correlation values greater than or equal to 0.01 using the correlation class from the Pandas library in Python programming language.

Distance traveled and speed

The raw data have no information on distance and speed. These indicators were derived using pick-up longitude and latitude, drop-off longitude, and latitude for distance calculation in kilometers. At the same time, travel speed was calculated using distance in kilometers and travel time in hours. To calculate the distance traveled, a 'Geopy' Python client or library was utilized, and this client uses classes from geocoders for Norminatim open-source geocoding service with Open Street Map (OSM) as well as Google Map's Application Programming Interface (API) for geocoding (Python-Community, 2020). Geodesic distance is the class that the Geopy python library uses to calculate the distance traveled by each commuter from originating point to destination point.

Geodesic distance by default adopts Karney (2013) method and uses the earth's model, which is an ellipse, to calculate the shortest distance on the surface of this WGS-84 ellipsoidal default. (Geopy-Contributors, 2018). One of the problems geodesic tends to solve is the inverse problem that involves calculating the shortest path between two points (Karney, 2013). Figure 13 describes the ellipsoidal model of the earth where the geodesic distance is S12 between AB, 12 is the longitude of B relative to A, 1 and 2 are latitudes A an B respectively, there are also two meridians NAF and NBH and the equator EFH (Karney, 2013).

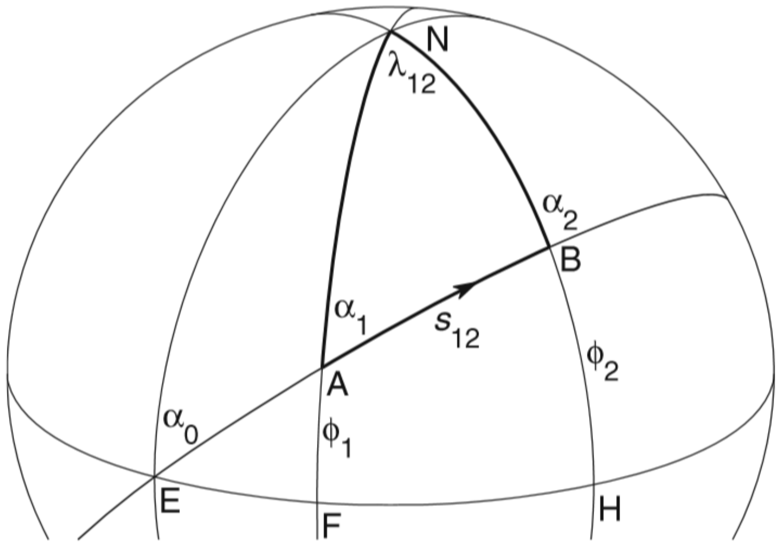

Figure 13: The ellipsoidal triangle NAB (Karney, 2013)

Therefore, in calculating the commuters' distance traveled, pick-up longitudes and latitudes with drop-off longitudes and latitudes were features used, and the results were converted into kilometers. Kilometer ranges were also created as additional indicators to examine their influence on the travel time. These ranges include 0 - 2 KM, 2 - 4 KM, 4 - 6 KM, 6 - 8 KM, 8 - 10 KM, 10 - 12 KM, 12 - 14 KM, 14 - 16 KM and 16 - 20 Km. These were established to check if commuters traveling within those 2-km intervals have high significance on travel time or can be relevant indicators for developing travel time prediction models. Eventually, the highly significant of these new indicators were selected base on correlation values greater than or equal to 0.01 using the correlation class from the Pandas library in Python programming language.

Traffic analysis zones

Categorizing the coordinates into segments gives insight about the number of commuters departing a region or location and arriving at the same or other zones. There is a relationship between commuters' movement and the vehicle stops, the zones, and the driver, which results in discovering the departure time and location and arrival time and location (Masiero et al., 2011). That shows the importance of creating traffic analysis zones to divide the whole case study area, Chengdu City, into smaller regions for in-depth analysis.

Taking into consideration the geographical range of coordinates given from DiDi Chuxing Gaia initiative platform, which include; latitude [30.727818], longitude [104.043333], latitude [30.726490] and longitude [104.129076], the city was segmented into nine (9) zones. These zones consider departing and arriving commuters at each zones and the coordinate ranges used to create these zones are, minimum latitude [30.5] and maximum latitude [30.9] (for both pick-up and drop-off latitudes) and also minimum longitude [103.9] and maximum longitude [104.5] (for both pick-up and drop-off longitudes). The total length of the whole study zone between the maximum and minimum latitude coordinates is 44.3457km, while the total distance between maximum and minimum longitude coordinates is 57.5993km, calculated using the 'geopy' class of python.

The nine traffic analysis zones (TAZ) have 15km by 15km length and width each and they are similar to having North Zone, South Zone, East Zone, West Zone, North-East Zone, South-East Zone, North-West Zone, South-West Zone and the Central Zone. Trip origin and destination coordinates that fall within these zones are stored, and these nine zones were identified as additional indicators influencing travel time. This denotes nine origin zones and nine destination zones created as indicators or inputs for modeling, considering their correlation values. Highly significant of these new indicators were selected base on correlation values greater than or equal to 0.01 using the correlation class from the Pandas library in Python programming language.

3.4.3 Working mechanisms and tools for modeling

Artificial Neural Network (ANN)

The method of the approach adopted to develop the Artificial Neural Network Model for this research work was, at first, creating a simple ANN architecture with an input layer, one hidden layer, and output layer. The first model has an input layer containing six (9) neurons (or Units), and the inputs include Pick-up Longitude, Drop-off Longitude, Ride Start Hour, Pick-up Latitude, Drop-off Latitude, Ride Start Minute, Ride Start Month, Ride Start Day and Trip Distance. The hidden layer also has nine (neurons), and the output layer contains one neuron, known as the travel time. The model training ran for 50 iterations (or Epochs), and afterward, the training loss and validation loss plotted, have converging curves. Subsequently, for the first model, the activation function used was the Rectified Linear Units (ReLU), the optimizer was Adaptive Moment Estimation (Adam), and the loss function was Mean Absolute Error (MAE).

The activation function is always applied to Neural Networks to avoid the output signal to become simple linear function or linear regression model, and linear equations were found not to have the ability to capture intricate patterns from big data sets (Sharma et al., 2020). Sharma et al. (2020) further noted that neural networks should have the ability to map patterns from arbitrary complex input data to output data and non-linear process functions, which thus necessitate the application of activation function in the architecture. ReLU was used as an activation function due to its wide usage for the prediction of continuous data and non-linear models. It is efficient because not all neurons are activated simultaneously, and the gradient is zero on some occasions during the training of the network (Sharma et al., 2020).

Adam optimizer was applied to extract parameters to reduce cost function through iteration, and this optimizer is widely used compared to Root Mean Square prop (RMSprop) and Stochastic Gradient Descent (SGD), and it combines the advantages of Root Mean Square prop (RMSprop) and Adaptive Gradient (AdaGrad) (Yang et al., 2019). Base on these reasons, Adam optimizer was used, and MAE adopted as the loss function. Measuring the prediction accuracy and knowing the deviation between prediction and actual values, make loss function to be important in neural network training (Yang et al., 2019).

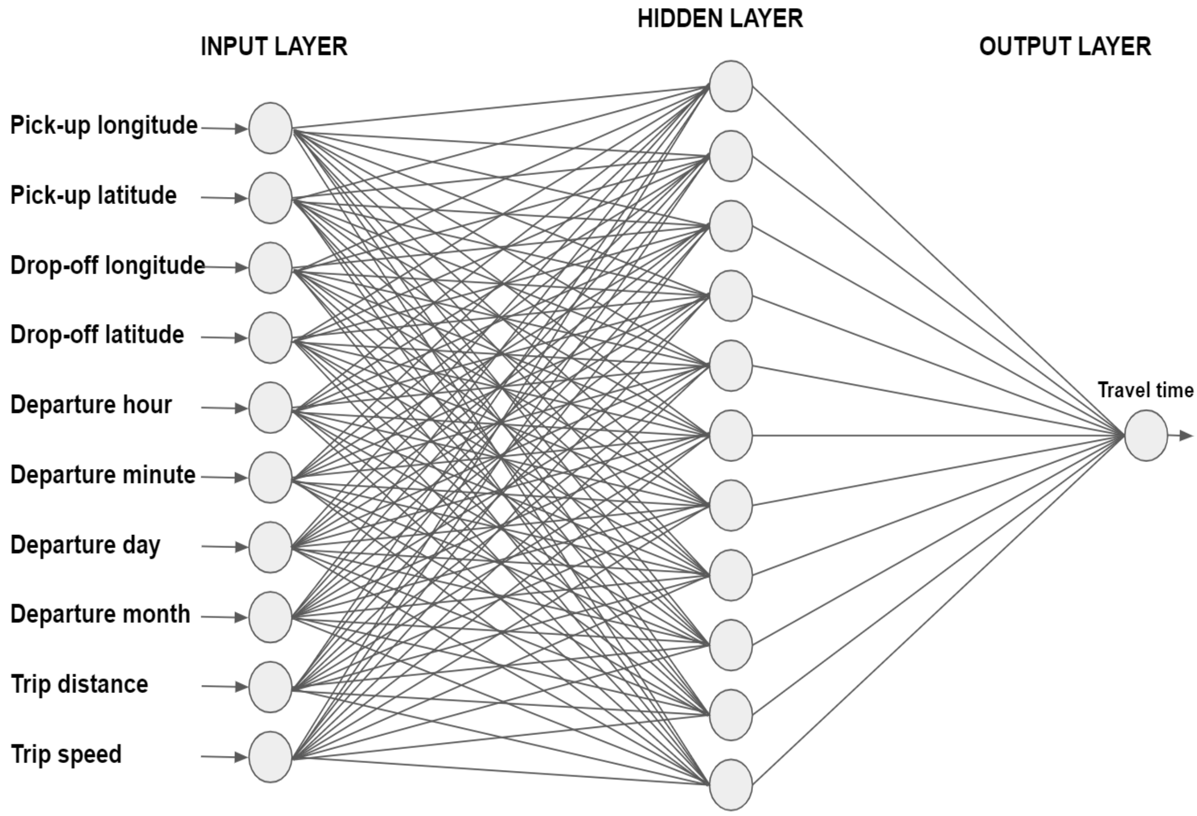

Figure 14: Artificial Neural Network (ANN) Model Architecture

Five ANN models were developed and compared, and these models have similar activation functions, loss function, number of units, number of epochs, number of hidden layers, and optimizer as the first model. The difference between these models is the number of inputs or travel time indicators, as derived above. The five models were developed with four (4) hidden layers based on their respective number of features used as input, and they were further elucidated in the data analysis chapter 4 of this research. Before the inclusion of travel speed, which has very high significance with travel time, three models were already developed and compared. The first model without speed included the indicators shown in Figure 14, distance traveled range was added to the second model without speed while the third model included the Traffic Analysis Zones for origin and destination, as explained in the section above on the methodology adopted in deriving relevant indicators. The four hidden layers that were eventually used in the five models arrived at after several trials and errors with different hidden layer numbers, starting with one hidden layer up to four hidden layers.

The Tools used for model development were the TensorFlow and Keras Python libraries and clients. TensorFlow is a product of Google and Alphabet, developed for large scale models trained with machine learning techniques as a second-generation system capable of running with thousands of Graphics Processing Units (GPU) cards on hundreds of machines for special purposes (Abadi et al., 2015). The 'Dense', 'Activation', 'Adam,' and 'Sequential' classes of TensorFlow- Keras library were used during the modeling using the Python programming language and Jupyter Notebook. The models were optimized and compared using values of coefficient of determination (R-squared), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) derived from the models. These techniques are equivalent to TensorFlow-Keras classes of 'R2_score', 'mean_absolute_error', and 'mean_squared_error' whose square-root was calculated.

Convolution Neural Network (CNN)

Some approaches adopted in the Artificial Neural Network modeling were also applied to the Convolution Neural Network (CNN) models. Parameters such as Adam optimizer, Mean Absolute Error (MAE) loss function, Rectified Linear Unit (ReLU) activation function, and fifty (50) iterations were used across all models. Five models were also developed with a similar number of inputs, a number of hidden layers and output layer, and the same number of neurons, respectively, as in ANN architecture with trials and errors inclusive. It also consists of a Convolution 1D layer with 64 units or neurons, one flatten layer and a layer with max-pooling 1D and other fully connected 4-layer models. Convolution 1D, Flatten, and Max-pooling 1D are commonly used classes of TensorFlow-Keras libraries to predict outputs that are not images or classifications but those with continuous values. Similar tools were also used for modeling with CNN.

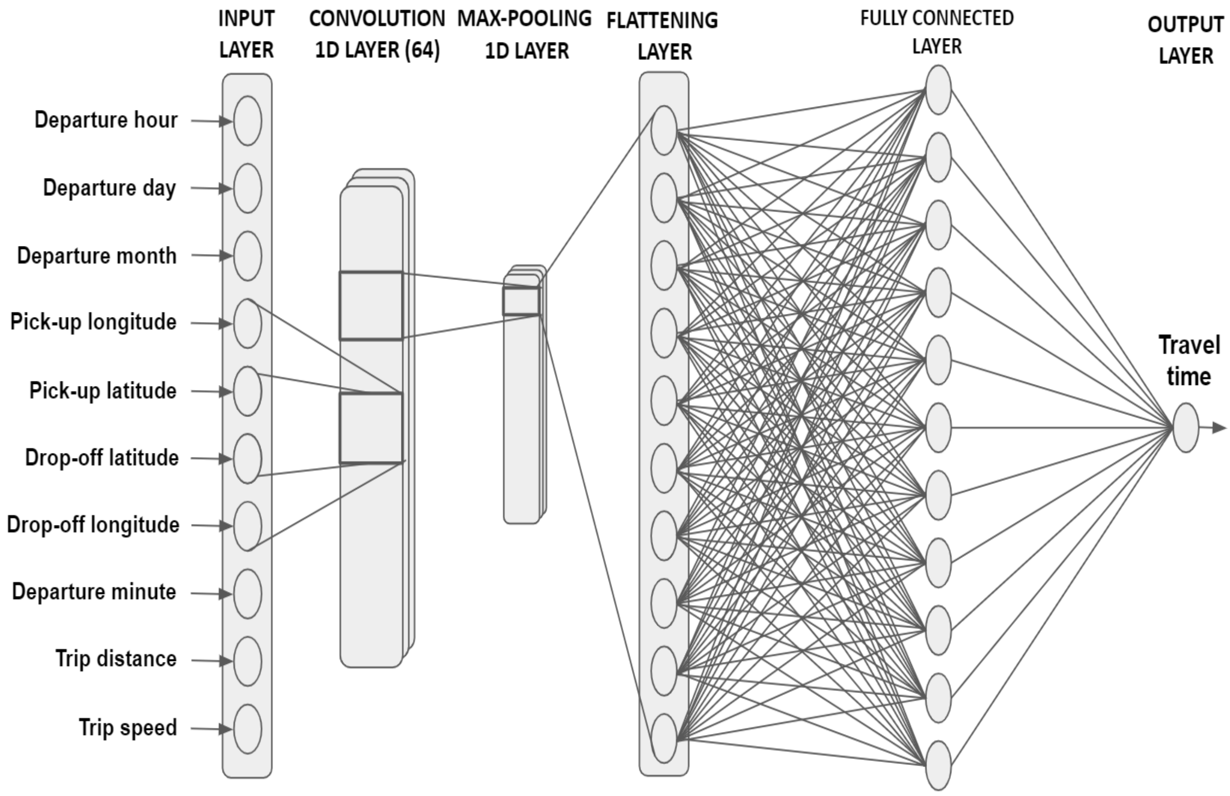

Figure 15: Convolution Neural Network (CNN) Model Architecture

Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) model was developed with a similar approach to ANN and CNN, using similar tools, Python and TensorFlow-Keras libraries. The 'LSTM' class was also utilized from the TensorFlow library to develop LSTM layers for modeling. A similar number of iterations, optimizers, loss functions, and activation functions were also adopted. The model comparison and evaluation were made with the RMSE, as mentioned earlier, MAE, as well as the coefficient of determination values.

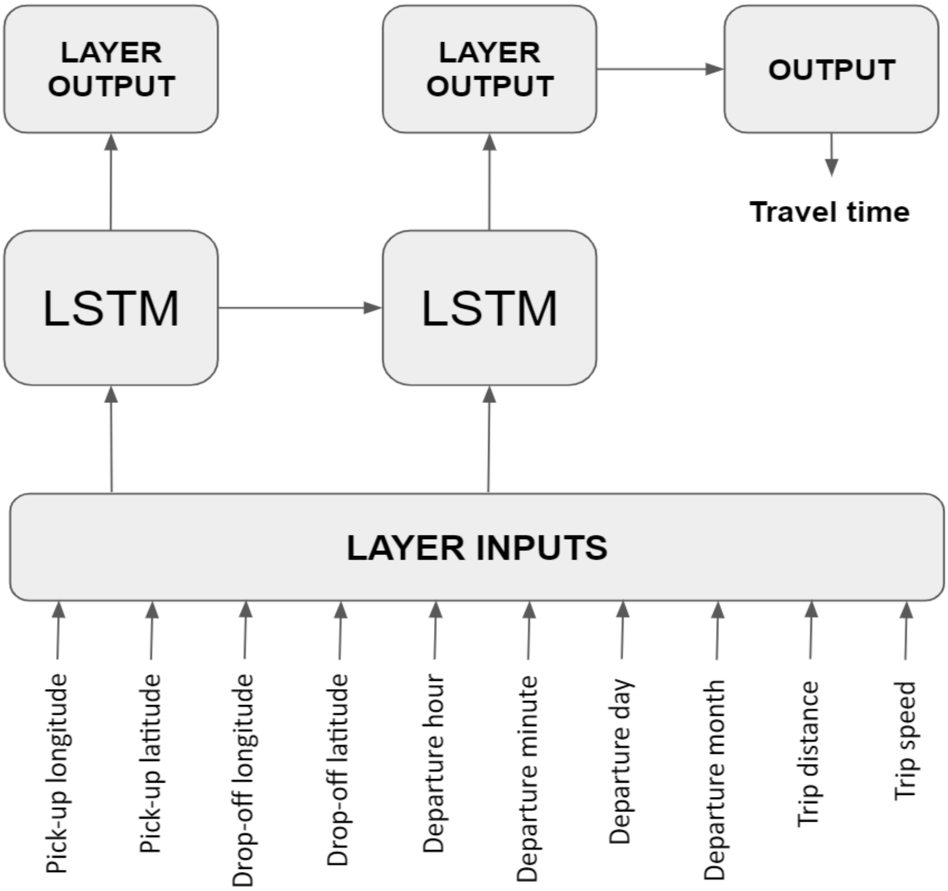

Figure 16: Long Short-Term Memory (LSTM) Model Architecture

3.5 Research limitations

In this section, a few potential challenges and limitations encountered during this research work are discussed. One of these limitations is the case study approach adopted. Chengdu city in China is the location of the study area of this research. The data sets were collected from DiDi, a mobility company in China, via the Didi Chuxing Gaia initiative platform. These ride request data sets, however, do not, in its entirety, represent the real world road conditions or travel factors that could influence the travel time of individual on-demand trips. Real-world factors such as trip purposes, activities, or events in the study area, seasonal changes or weather conditions, home-based or work-based trips, and road traffic conditions, were not captured in the data sets.

These factors are good indicators for travel time prediction, and these limitations led to the first objective, focused on identifying only relevant data-driven indicators. Since this research requires large data sets to develop deep learning models and forecast travel time, the data-driven factors identified in this study fit well to achieve the second objective on deep learning models' development for prediction. The researcher also has no access to the study area, which requires funds and travel process. Having access to the study area will be a plus to this research to identify real-world indicators, instead of only the data-driven indicators.

Another limitation experienced during this research is the researcher's computation device, which has a low capacity of Random Access Memory (RAM) and solid-state drive (SSD) disk memory, among other technicalities. Deep learning models require high-capacity computers with high computational power to train models into having convergence and stability for validation and training loss curves. More data are needed to overcome over-fitting, and the larger the data, the higher the required memory capacity to compute and train the models. These eventually resulted in using only the first (7) days of the 30-day test data containing real-world travel time, collected every 2-4 seconds (time-steps) on the commuter's trip routes. The seven-day data have about 315 million time-steps equivalent to the number of rows in the data set, out of about 1.35 billion in the time-step total for 30 days.

Therefore, in-depth and more accurate research in travel time prediction demand resolving the aforementioned technical challenges in data computation and study area accessibility to examine real-world indicators capable of influencing travel time of individual demand-responsive trips. Finally, to overcome these research study limitations, more time is required for the research study, as the researcher has limited months as a master's student to reach a considerable milestone.

Chapter 4

Data analysis and discussions

4.1 Identification of relevant travel time indicators

This section discusses identifying various relevant data-driven indicators for travel time, used as inputs for modeling. As determined in previous literature in the second chapter of this thesis, ascertaining the primary indicators' level of significance to travel time is highly essential. The method used in identifying them is in line with the research approached discussed in the methodology chapter, which involves knowing their correlation values using a class in the panda's library for the Python programming language called 'corr().' Furthermore, the researcher elucidates the statistical description of these indicators, the distribution density of highly significant indicators such as travel time, distance, speed, departure hours, and departure minutes. Also, the spatial trip distribution plotting the longitudes and latitudes are discussed in relation to travel time, travel distance, travel speed, and inclusive of the newly created nine traffic analysis origins and destination zones.

Discussing the indicators' distribution densities is vital to identify the relevance and volume patterns of the indicators within the training data set. Consequently, these indicators are data-driven due to their derivation from the primary indicators in the raw data and using the origin-destination data set to identify their level of significance with travel time. The origin- destination dataset serves as training data for the models, and it captures individual commuter's trip start location and end location in the study area. The second dataset collected is the Global Positioning System (GPS) coordinates of individual commuter's trip routes from the origins to their destinations, thereby using it as test data. It represents the real travel time, recorded every 2-4 seconds of the trips. Figures 17 and 18 illustrate the first data set (train data) split into origin and destinations while Figure 19 has the second data set (test data) of routes taken by each trip, which captures smaller region within the city center while Figure 20 shows the combined map of coordinates for origin-destination (departures and arrivals respectively) and the trip routes. The maps are on a scale of 45 km by 58 km distance.

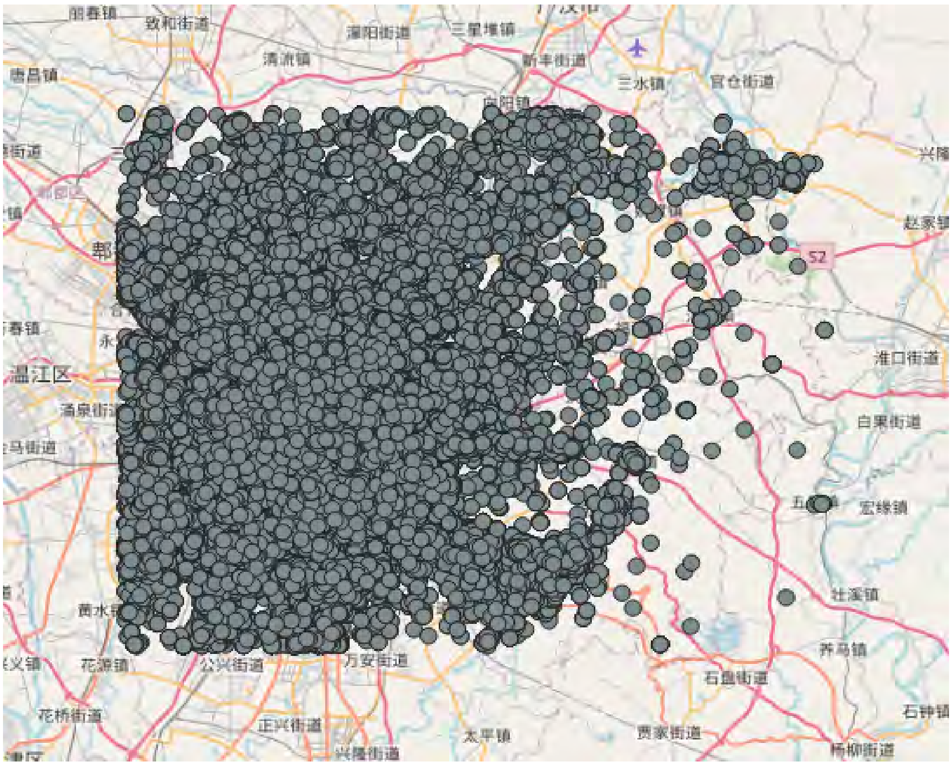

Figure 17: Open Street Map (OSM) for 7 million coordinates of on-demand trip departures in Chengdu City, China (30 days model training data)

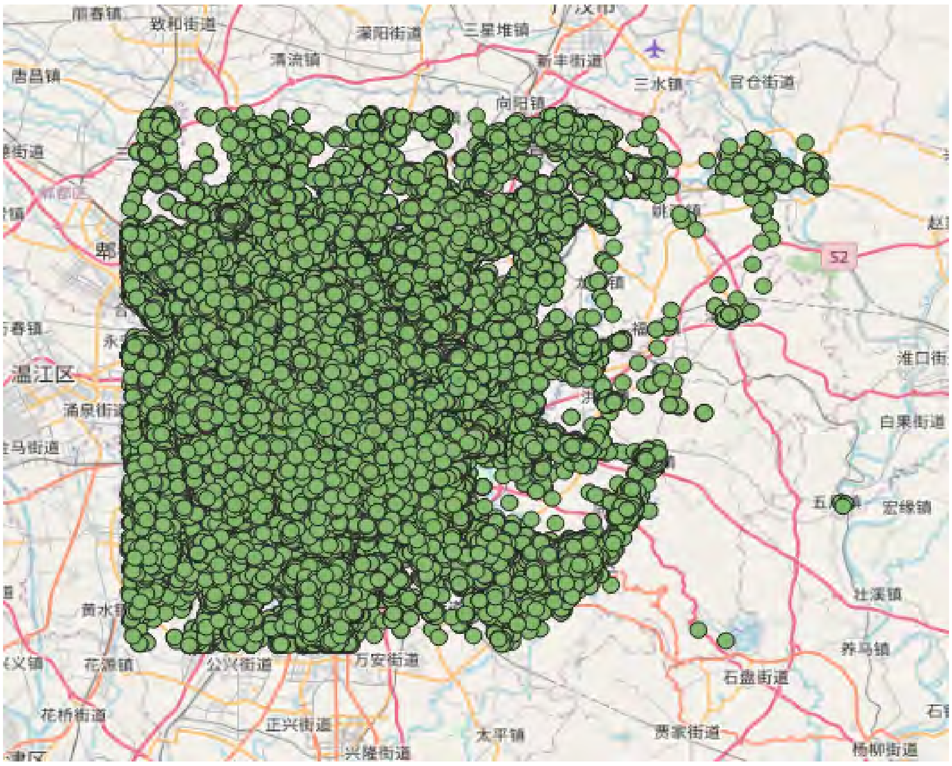

Figure 18: Open Street Map (OSM) for 7 million coordinates of on-demand trip arrivals in Chengdu City, China (30 days model training data)

Figure 19: Open Street Map (OSM) for about 315 million time-step coordinates of on-demand trip routes in Chengdu City, China (7 days model test data)

Figure 20: Combined 1.35 billion coordinates of on-demand trip arrivals, departures and routes in Chengdu City, China using Open Street Maps (OSM)

The four Open Street Maps (OSM) indicate the massiveness of data, in which billions of the coordinate points overlap with each other. The overall geographical scale is approximately 45 km by 58 km height and width as in the map, respectively. These scales are also used in further graphical plots in subsequent sections, where the 45-km height represents latitude (y-axis), and the 58-km width indicates the longitude x-axis. The grey points represent the pick-up or departure coordinates, while the green points indicate the drop-off or arrival coordinates of each commuter. The dark brown points represent the routes taken during the trip from respective origins to destinations. They are on a smaller portion of the pick-up and drop-off coordinates because of the volume of the data collected every 2-4 seconds of the commuter's movement. The route's points also represent the real travel time used to test the model to check the model prediction accuracy.

4.1.1 Statistical description of relevant indicators

After cleaning the origin-destination data and preparing the visualization of each indicator's distribution, seven million six hundred and forty-five (7,000,645) ordered trips are found fit for the model training. The total requested trips from the raw data set collected are seven million fifty-eight thousand nine hundred and fifty (7,058,950). In essence, an approximate percentage of 99.2 % of the raw data represents training data after data cleaning. The remaining 0.8 % are extreme values or outliers of longitude-latitude coordinates, speed, distance, and travel time. The geographical ranges stated on the data collection platform are considered before processing the data and removing all outliers. This step was explained, in-depth, and discussed in the research methodology chapter 3. The summarized statistics of the training data is as described in Table 4.

According to the figure on the statistical summary for the training data and trips across 30 days, the average travel time (mean) is 22 minutes, and the maximum travel time (max) is approximately 160 minutes and including a standard deviation (std) of 13 minutes approximately. The figure shows most of the trips are short and perhaps within the origin. There is also the possibility of traffic congestion or trips to suburbs as an impact on the maximum travel time of over 2 hours (hr). The distance traveled across all trips have a maximum estimation of 50 kilometers (km) at an average of 5 km with a standard deviation of 5.3 km.

These values confirm the short commuter's trips and the possibility of long trips to suburbs as described above with travel time. These also apply to the average estimated speed is 17 km/h for all trips with 22 km/h as standard deviation, which indicates short trips with low travel speed, usually within the city's Central Business Districts (CBD) with lots of activities. All these indicators are primary and commonly used by previous researchers. The statistical summary tells how highly significant these indicators are, for the travel time prediction modeling.

Table 4: The statistical summary of the origin-destination on-demand trip data

Indicators	count	mean	std	min	25%	50%	75%	max
Drop-off Longitude	7000645.0	1.040730e+02	0.040396	1.039000e+02	1.040503e+02	1.040735e+02	1.040980e+02	1.044867e+02
Drop-off Latitude	7000645.0	3.067336e+01	0.037974	3.050003e+01	3.065466e+01	3.067070e+01	3.069245e+01	3.089988e+01
Pick-up Longitude	7000645.0	1.040730e+02	0.041372	1.039000e+02	1.040507e+02	1.040737e+02	1.040973e+02	1.044877e+02
Pick-up Latitude	7000645.0	3.067269e+01	0.038850	3.050001e+01	3.065399e+01	3.067014e+01	3.069227e+01	3.089940e+01
Trip order Id	7000645.0	1.020721e+05	59284.752401	1.000000e+00	5.088100e+04	1.017640e+05	1.526400e+05	2.237820e+05
Travel Time Minutes	7000645.0	2.237453e+01	13.023396	5.000000e-02	1.321667e+01	1.965000e+01	2.835000e+01	1.596333e+02
Travel Time Hours	7000645.0	3.729088e-01	0.217057	8.333333e-04	2.202778e-01	3.275000e-01	4.725000e-01	2.660556e+00
Ride Start Hour	7000645.0	8.363290e+00	5.697881	0.000000e+00	4.000000e+00	8.000000e+00	1.200000e+01	2.300000e+01
Ride Start Minute	7000645.0	3.177254e+01	17.540097	0.000000e+00	1.700000e+01	3.300000e+01	4.700000e+01	5.900000e+01
Ride Stop Hour	7000645.0	8.361410e+00	5.421327	0.000000e+00	4.000000e+00	8.000000e+00	1.200000e+01	2.300000e+01
Ride Stop Minute	7000645.0	2.734502e+01	17.628318	0.000000e+00	1.200000e+01	2.600000e+01	4.300000e+01	5.900000e+01
Ride Start Day	7000645.0	2.932641e+00	1.966725	0.000000e+00	1.000000e+00	3.000000e+00	5.000000e+00	6.000000e+00
Ride Stop Day	7000645.0	2.931644e+00	1.963421	0.000000e+00	1.000000e+00	3.000000e+00	5.000000e+00	6.000000e+00
Trip Distance (Km)	7000645.0	5.109024e+00	5.327912	9.573874e-05	7.699472e-01	3.445873e+00	7.888696e+00	4.998109e+01
Trip Speed (Km/h)	7000645.0	1.648542e+01	22.016907	6.591815e-05	2.758716e+00	1.010169e+01	2.234923e+01	6.492875e+02

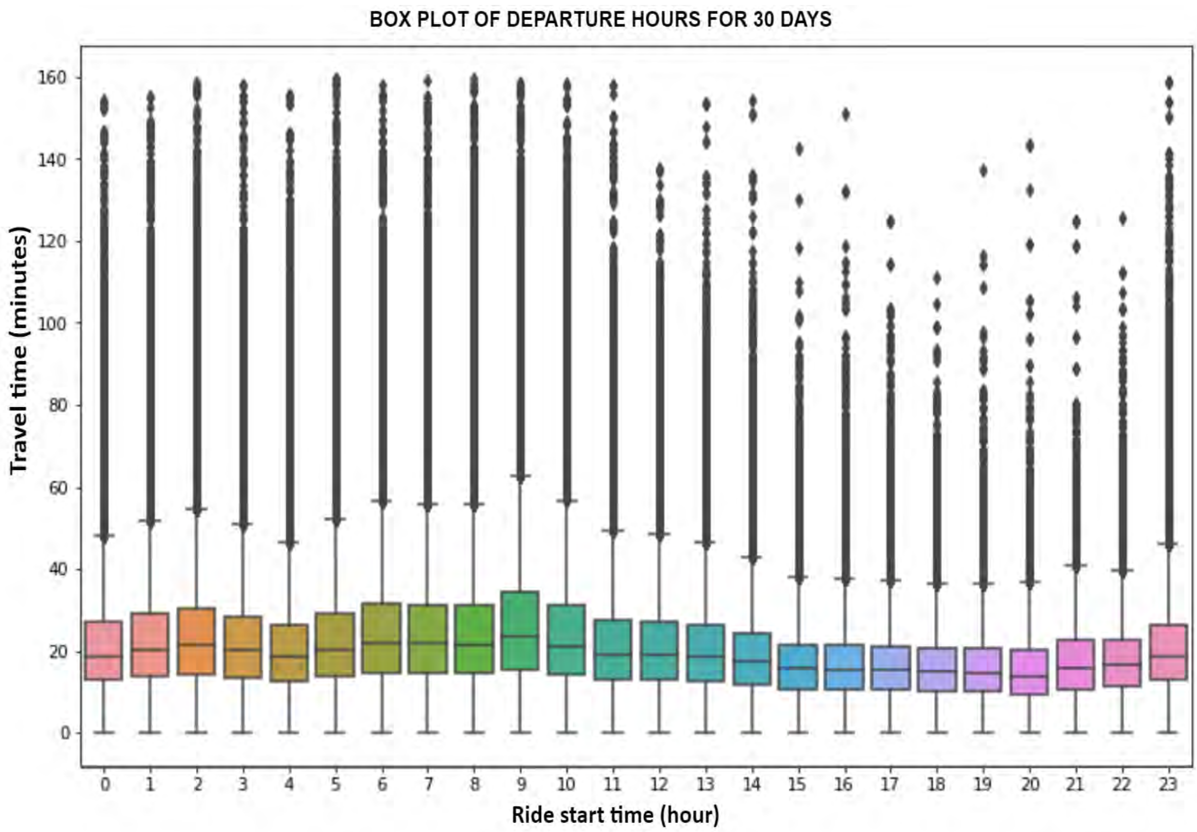

Figure 21: Box-plot of departure hours and travel time

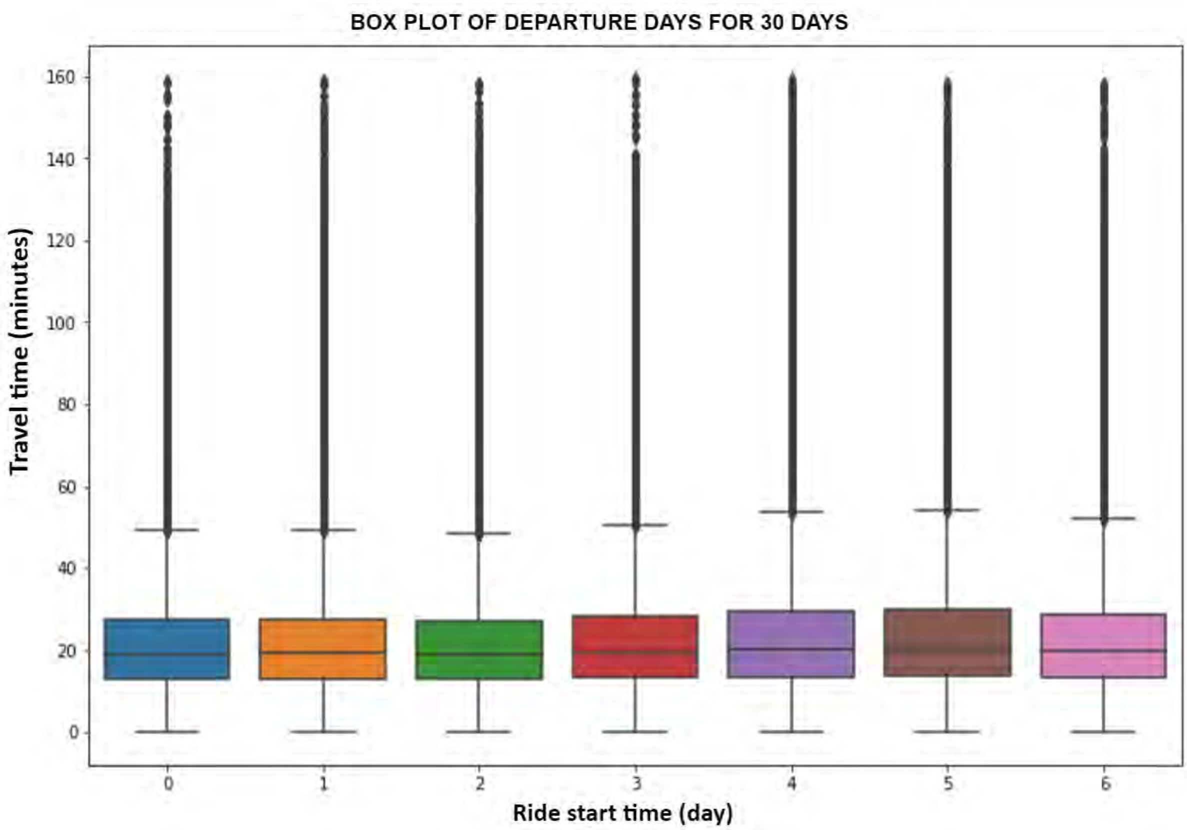

Figure 22: Box-plot of departure days and travel time

Departure time is an important feature in developing predictive models for travel time as it is presumed the arrival time is unknown during model training. This resulted in examining the statistical relationship between travel time and departure time, as indicated in Figures 21 and 22, which are departure hours and departure days respectively. The box-plots describe the travel time median, minimum, maximum, first quartile, third quartile, interquartile range and outliers, for every trip hour and trip day. These are similar to the statistical summary described previously but with no median and outliers. The median describes the mid value of the time taken to travel from every departure hour and minute. At the same time, the outliers are the extreme travel time values from the departure time to the trip destination.

The box-plot description for departure hour shows, overall, that daily departure hour across the 30 days training data have a large chunk of trips with a travel time ranging from 0 minutes to 70 minutes of maximum and minimum values on the box-plot, not taking outliers into account. In essence, most ride requests by commuters are within the city and mostly on short distance between the origin and destination, confirming most trips are either demand-responsive or on-demand. Similarly, most departures per day of the week are within range 0 - 60 minutes (min) travel time, according to the box-plots maximum and minimum points. It also has similar interpretations with the departure days, which the numbers represent Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday for 0, 1, 2, 3, 4, 5, 6, respectively.

Highly significant travel time indicators

Table 5: The level of significance of the travel time indicators

Travel Time Indicators	Level of Significance	Travel Time Indicators	Level of Significance
Trip speed (Km/h)	-0.217316	Ride start Minute	-0.019047
Pick-up Longitude	-0.110340	Ride Stop Minute	0.012096
Drop-off Longitude	-0.106567	Ride Stop Day	0.043164
Ride start Hour	-0.090763	Ride Start Day	0.045336
Ride stop Hour	-0.077261	Trip Distance	0.271076
Pick-up Latitude	-0.057920	Travel Time Hours	1.000000
Drop-off Latitude	-0.042396	Travel Time Minutes	1.000000

The reviewed literature have similar common indicators influencing travel time, as found in the training data and Table 5. The level of significance table reveals relevant indicators with correlation values higher or equivalent to 0.01 or 1%. The maximum correlation is +/- 1 or 100%, and the only indicator with such correlation is travel time in hours and minutes, which is a correlation on itself. The highest positive relationship is trip distance with a value of 0.271076, and the highest negative correlation is the travel or trip speed with an amount of

-0.217316. The positive distance correlation denotes the higher distance traveled, the more prolonged the travel time and equally, the shorter the distance, the smaller the travel time to destination, all at a regular travel speed. The speed's high negative correlation shows the higher the travel speed, the shorter the time taken to travel to the destination, and when the speed rate gets lower, the travel time rises. These interpretations of the positive and negative correlation with travel time also apply to other significant levels of identified relevant indicators.

To build highly performing models, creating and identifying more relevant indicators with a high correlation to travel time are essential. The approach adopted in deriving each of these new relevant indicators was discussed in the methodology chapter. A bar chart of the correlation or level of significance is in Figures 23 and 24 showing the relationship between the travel time and all derived features. We can deduce that the travel time in minutes and hours have the highest correlation of 1 or 100%, with itself, which is simultaneously the output of the model prediction.

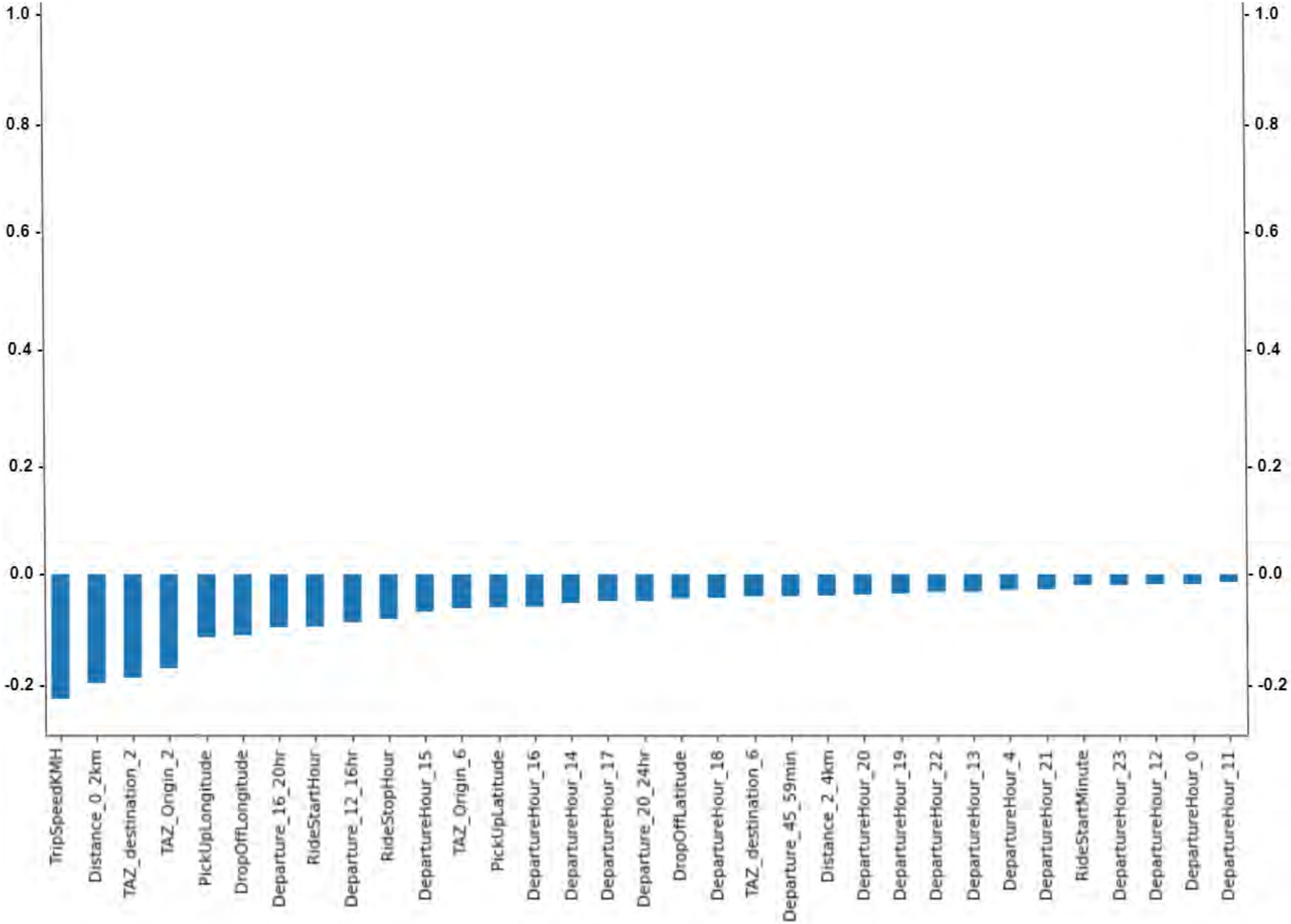

Figure 23: Highly Significant negative indicators with travel time

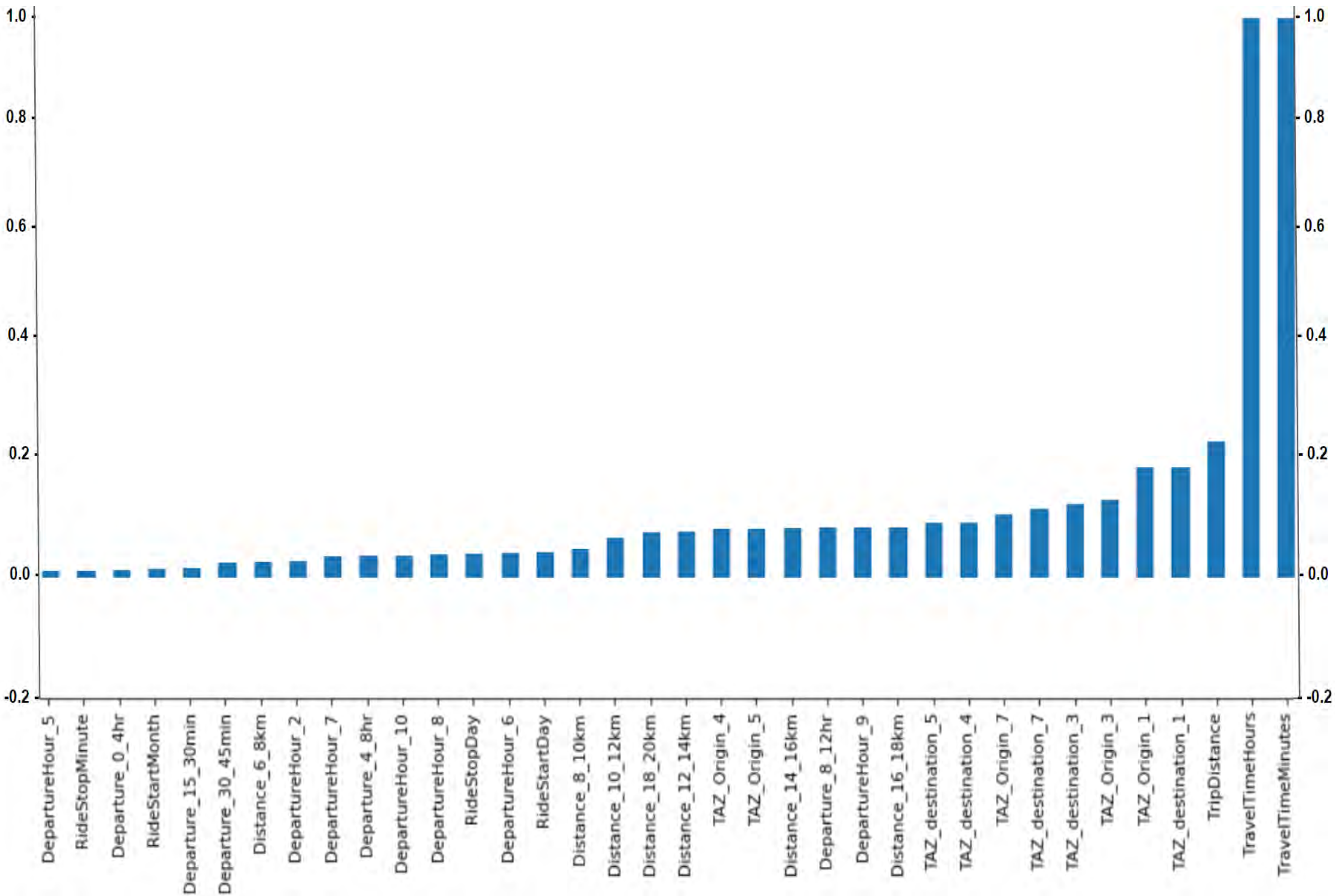

Figure 24: Highly Significant positive indicators with travel time

The bar chart figure of highly significant indicators for travel time reveals that aside from the travel time, there are three other indicators with the strongest positive influence on travel time, which are trip distance, traffic analysis origin zone (TAZ) 1, and traffic analysis destination zone 1. These results mean that the higher the number of individuals departing and arriving at zone 1 within a particular period of the day, the higher the travel time. As the departures and arrivals reduce in zone 1, the travel time also reduces. It is an indication of traffic congestion during peak hours or high population density within zone 1.

These descriptions also apply to the negative correlations. There are four top negative correlation indicators, which are trip speed, trips within a distance of 0-2 km, trip origin, and destination zones 2. The speed and distance have opposite interpretations, as discussed in the level of significance table, which is the higher the travel speed, the lower the travel time, and vice-versa. Furthermore, the two traffic analysis zone 2 negative relationships show that as most trips' travel time reduces, the number of departure and arrival detected in zone 2 increases. In contrast, as the departures and arrivals in zone 2 decreases the travel time increases. The interpretation means long distant trips are arriving and departing zone 2.

4.1.2 Distribution density of relevant indicators

Trip speed, distance and travel distributions

Figures 25, 26 and 27 show the distribution density of the trip speed, travel distance and travel time. Most trips have short travel time and short distance, ranging from 0-60 minutes and 0-30 kilometers, respectively. To ascertain most trips to be within the city or around the central business district (CBD), the distribution of the travel speeds are high between 0 and 50 km/h. The distance and speed indicators have a very highly correlated with travel time, just as discussed in the level of indicators' significance above. These distributions also show most trips are on-demand, and demand-responsive trips should fulfill commuters' immediate need to travel, efficiently, and without delay. These distribution densities are for the 30 days training data containing origin and destination.

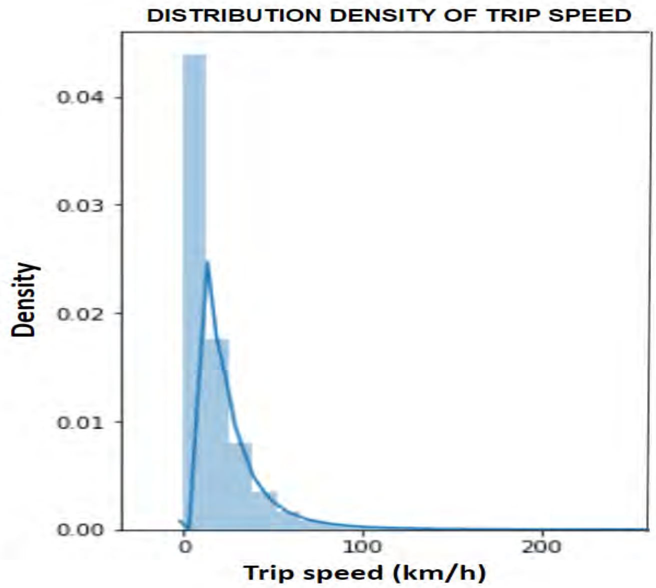

Figure 25: Distribution density of speed (km/h) in the training data of departures and arrivals

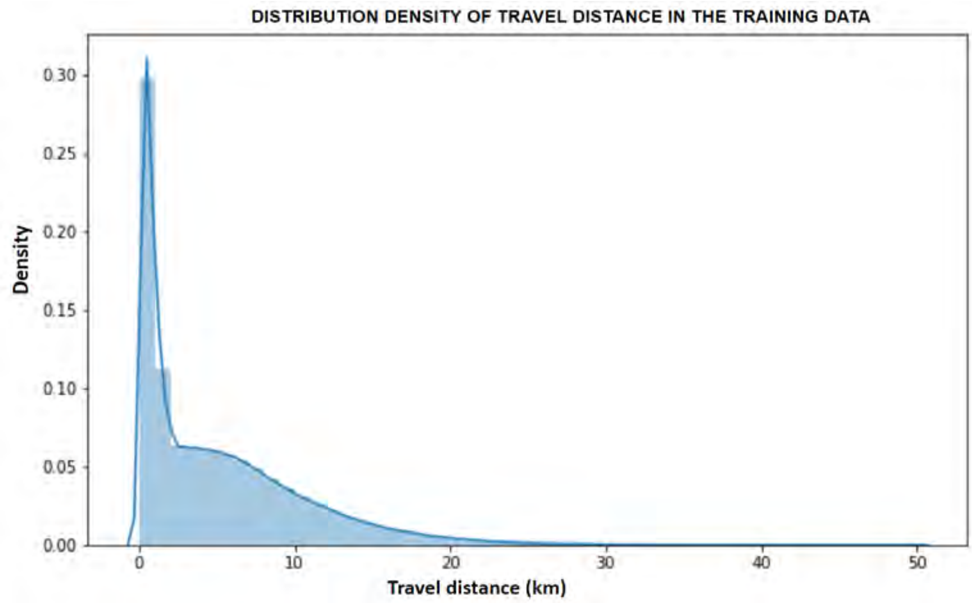

Figure 26: Distribution density of distance (km) in the training data of departures and arrivals

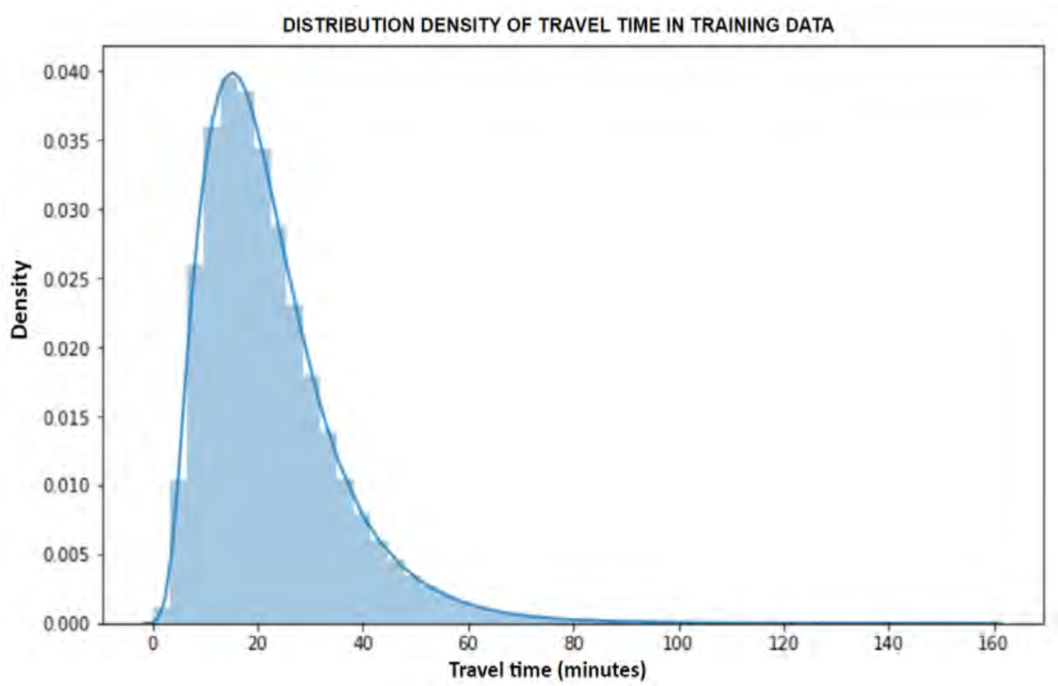

Figure 27: Distribution density of travel time (min) in the training data of departures and arrivals

Departure hours distribution

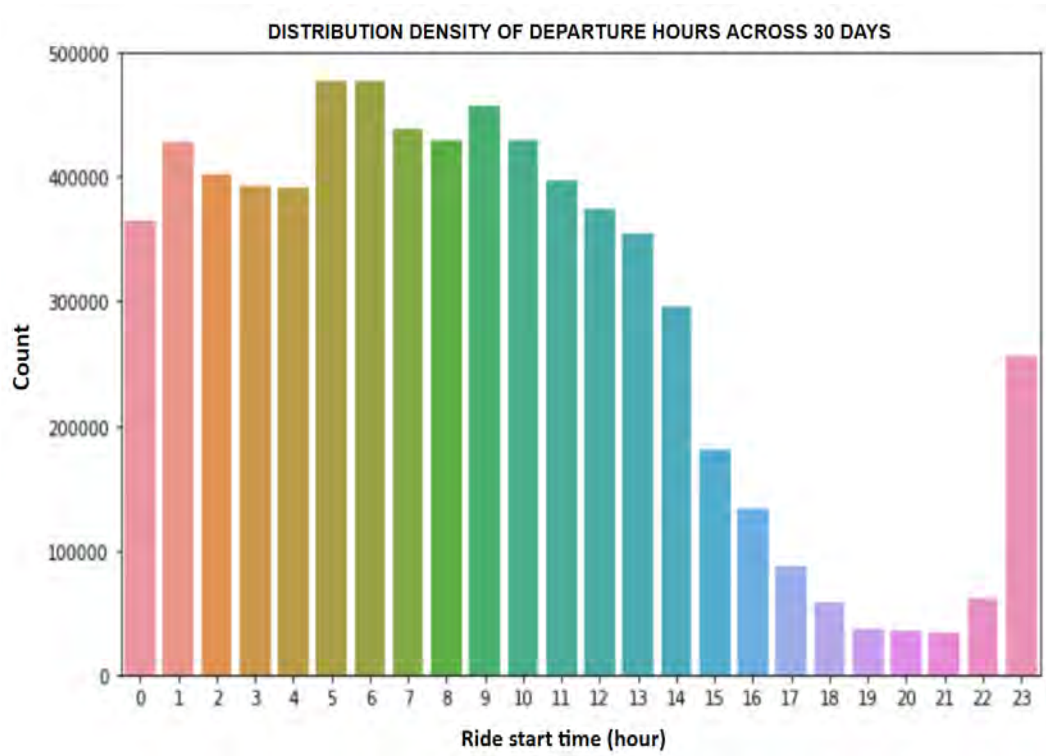

Figure 28: Departure hour distribution density across the 30 days training data

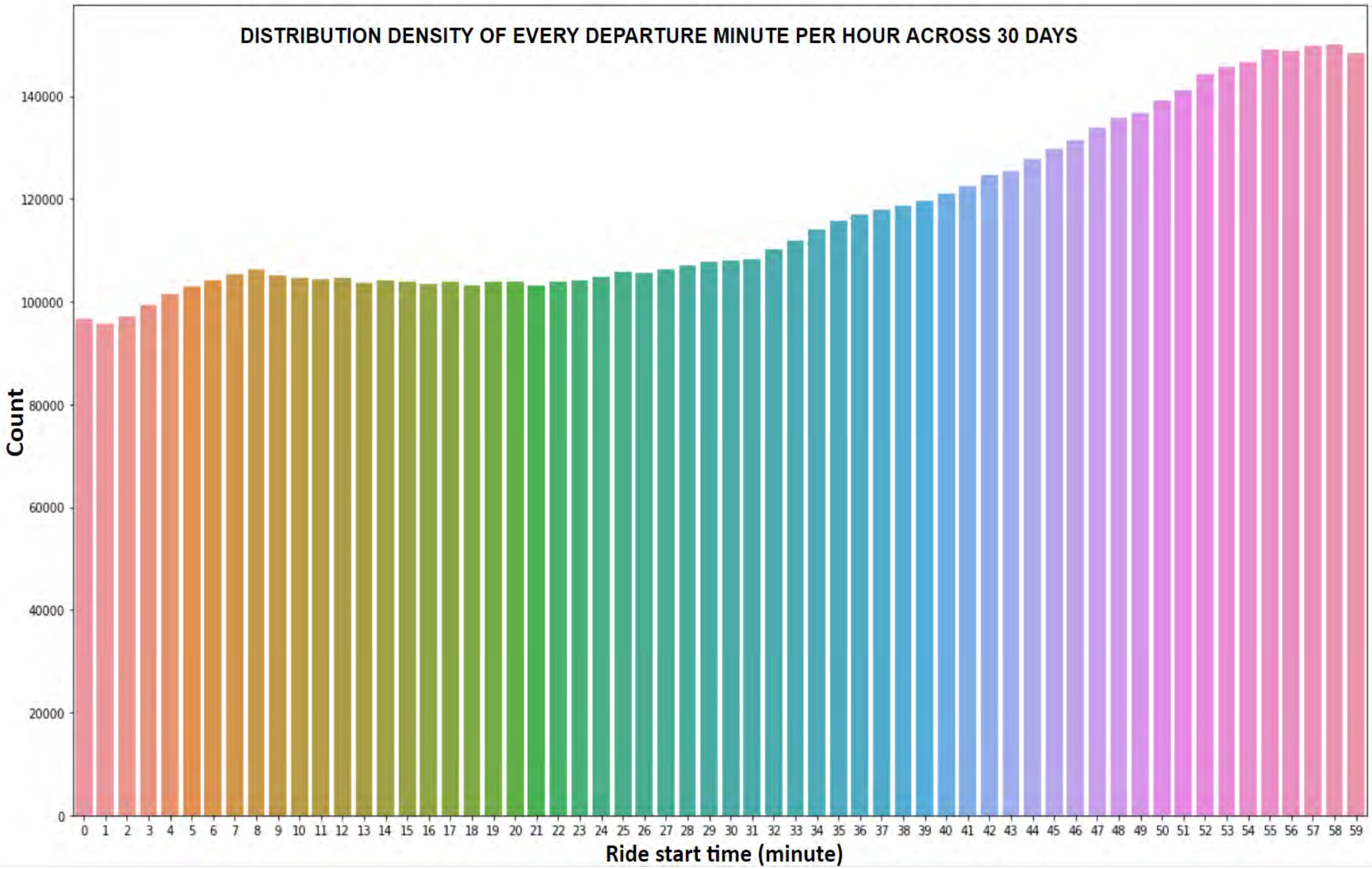

Figure 29: Departure minute distribution density across the 30 days training data

The departure time is also a highly significant or relevant indicator, and the distribution densities are as shown in Figure 28 for departure hours and minutes. There is an increased number of trips departing between 4.00 AM to 10.00 AM, confirming this period as peak hours. The off-peak hours range from 5.00 PM and 11.00 PM during the 30 days. Therefore, highly correlated departure peak hours also influence the travel time.

An in-depth study of the departure minutes, shows a lot of trips take off from 45th minute to 59th minute, for instance, 12.45 PM - 12.59 PM. Sequel to this discovery, the 60-minute clock is divided into four segments, which includes: 1st - 15th minute, 15th- 30th minute, 30th - 45th minute and 45th - 59th minute (e.g., 12.00 PM -12.15 PM, 12.15 PM -12.30 PM, 12.30 PM -12.45 PM and 12.45 PM -12.59 PM, respectively). The highly relevant of these minute categories in relation to travel time is also used as features or labels to train the models. These clock categories help identify the departure minutes with a strong influence on travel time at every departure hour of the day by giving the commuters and the mobility operators a forecast of the travel time for upcoming on-demand trips every 15 minutes of departure requests.

4.1.3 Spatial trip distribution of relevant indicators

Spatial distribution of travel time, distance and speed

Figure 30: Distribution of departure trip coordinates per travel time (min)

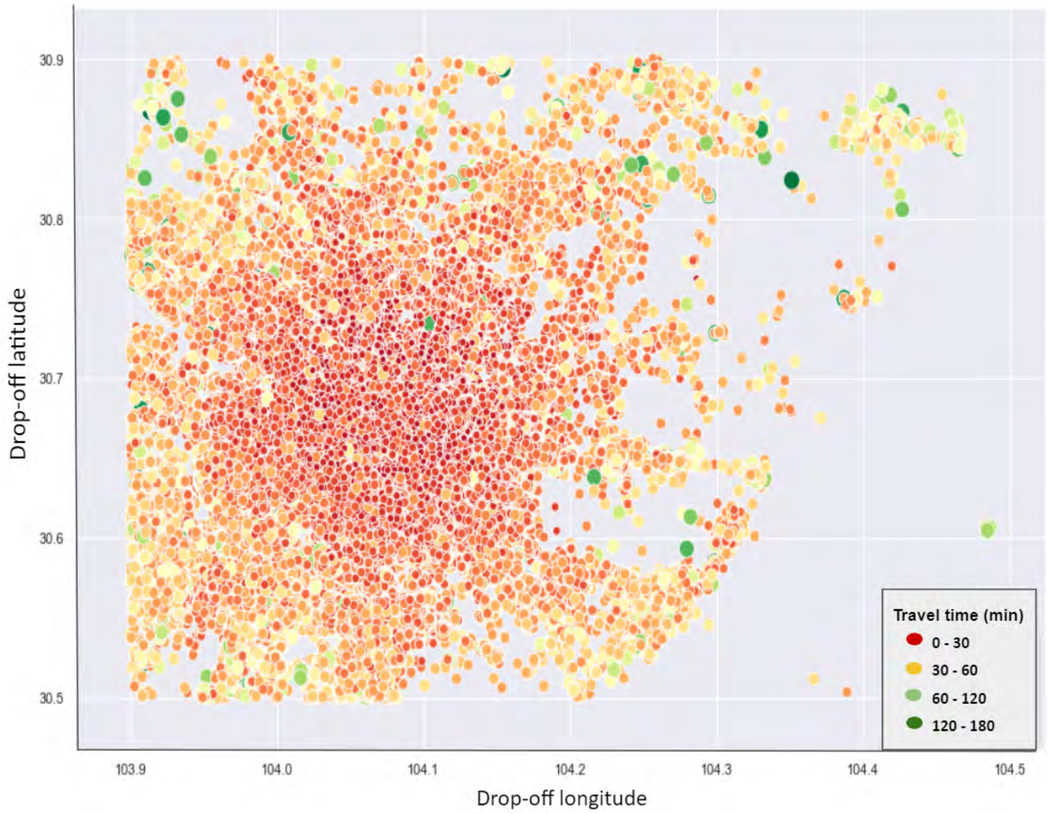

Figure 31: Distribution of arrival trip coordinates per travel time (min)

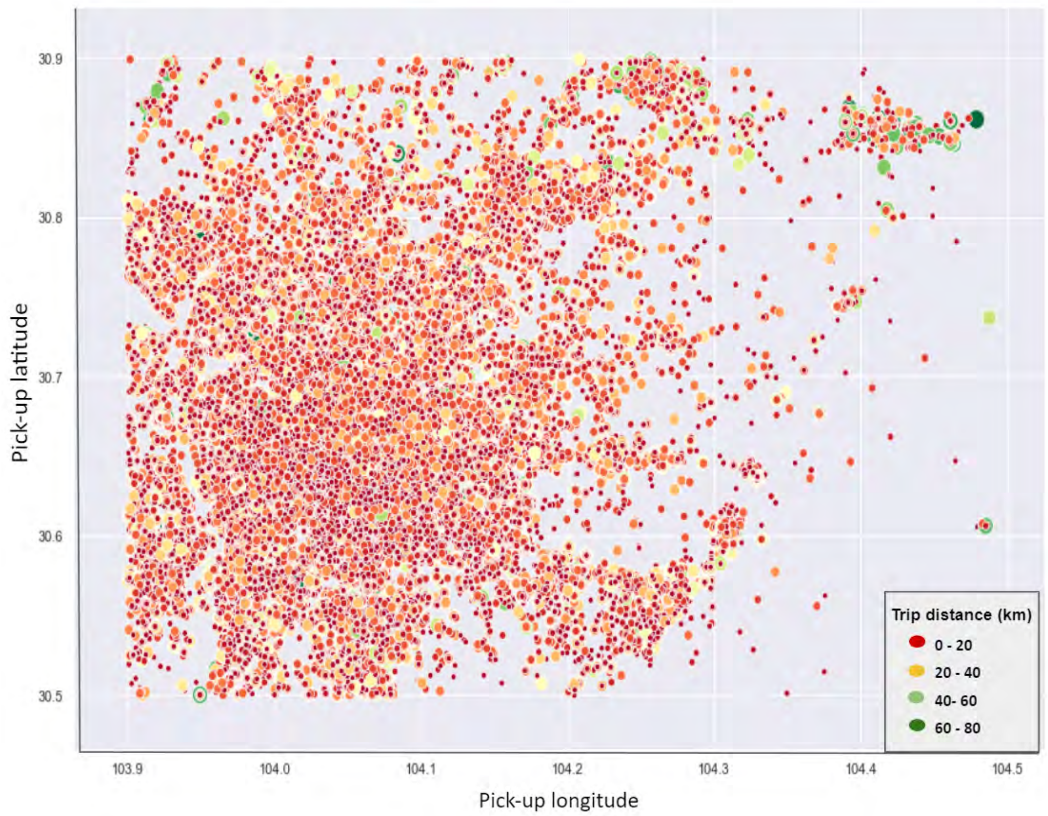

Figure 32: Distribution of departure trip coordinates per travel distance (km)

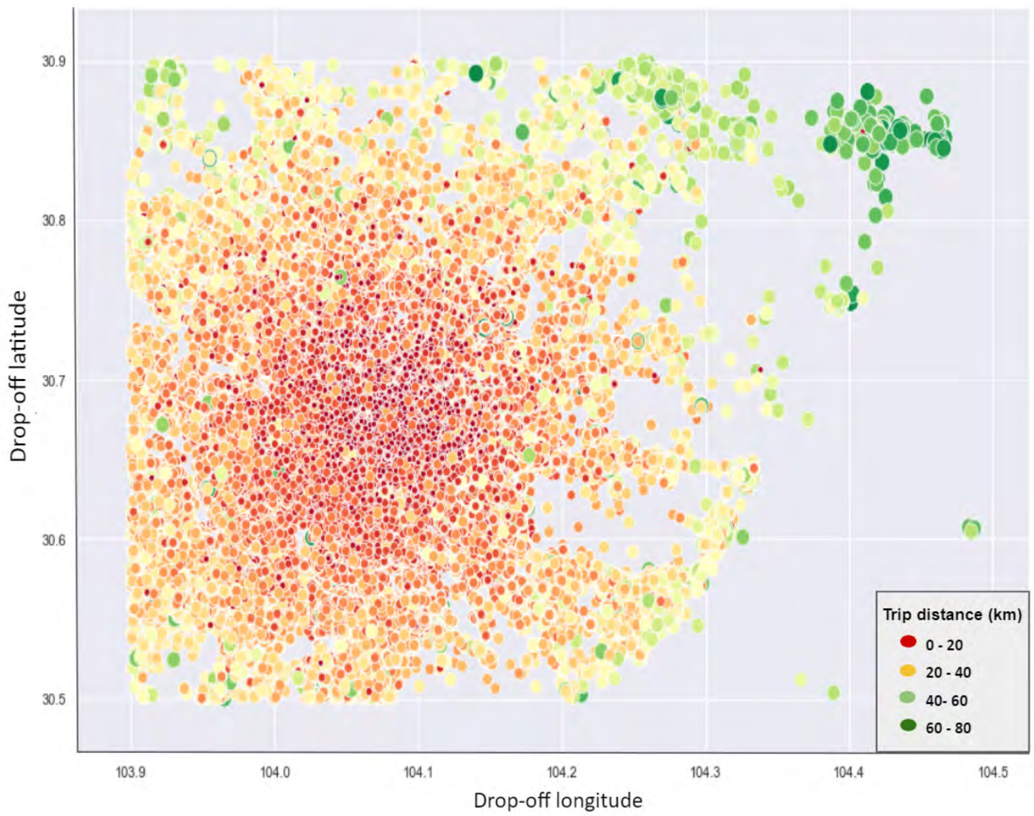

Figure 33: Distribution of arrival trip coordinates per travel distance (km)

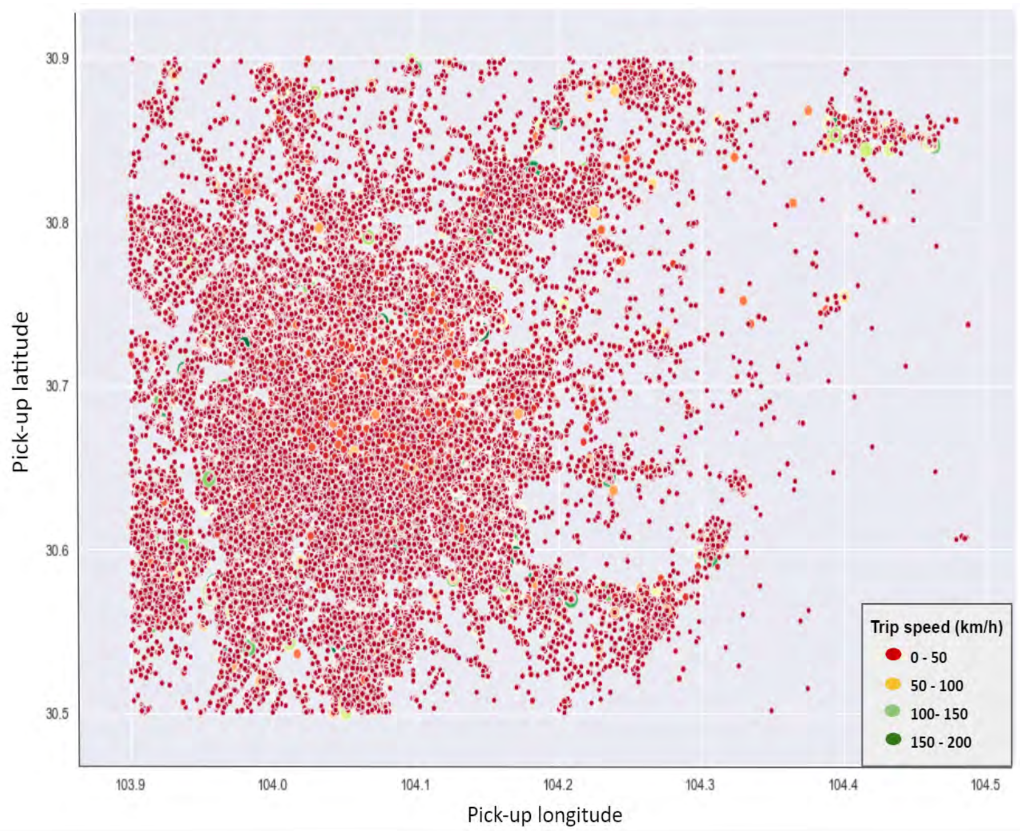

Figure 34: Distribution of departure trip coordinates per travel speed (km/h)

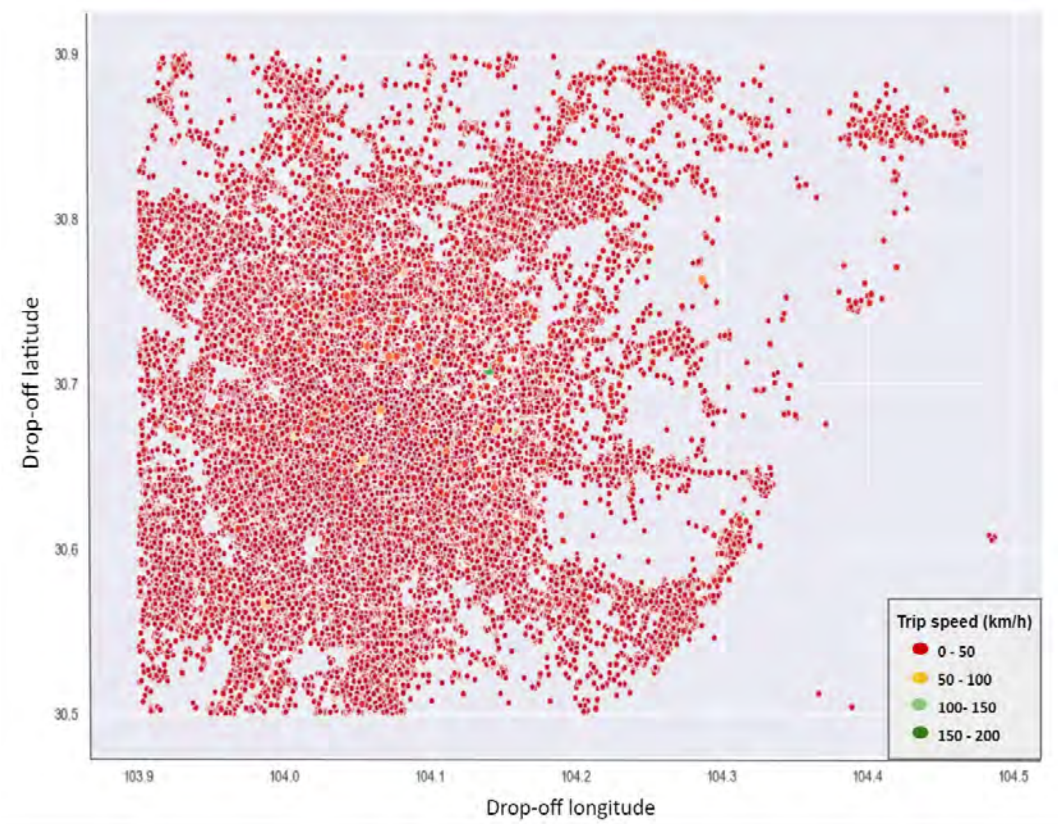

Figure 35: Distribution of arrival trip coordinates per travel speed (km/h)

The origin-destination trip distributions in relations with travel time, speed and distance, as displayed in Figures 30, 31, 32, 33, 34 and 35, are plots of pick-up longitude, pick-up latitude, drop-off longitude and drop-off latitude. The origin and destination plots per 60 minutes travel time disclose that large chunk of trip departures and arrivals are within 2 hours, concentrated at the city center or business districts. Few trips from 120 minutes, start or end at the city center while most of these longer trips are towards the city's outskirts for both departures and arrivals.

A different discovery is observed in Figures 32 and 33 between the origin and destination trip distributions. The origin or departure distribution shows many departed trips have a short distance range from 0-40 km while few departed trips took distance longer than 40 km. The destination or arrival plot discloses that trips arriving at the outskirt of the city drove longer from 40 to 80 km while arrival trip distance between 0 and 20 km are most at the center of the city. The destination coordinate plot shows the farther from the center, the longer the length, meaning, as trips approach the suburban area, the distance tends to increase, resulting from longer routes taken. In the case of speed, Figures 34 and 35 disclose that both the travel speed of departed and arrived trips range from 0 - 200 km/h. Furthermore, most trips accelerate between 0 and 50 km/h, which denotes trips within the central business district or a more significant portion of the city center's study area.

Trip distribution and traffic analysis zones

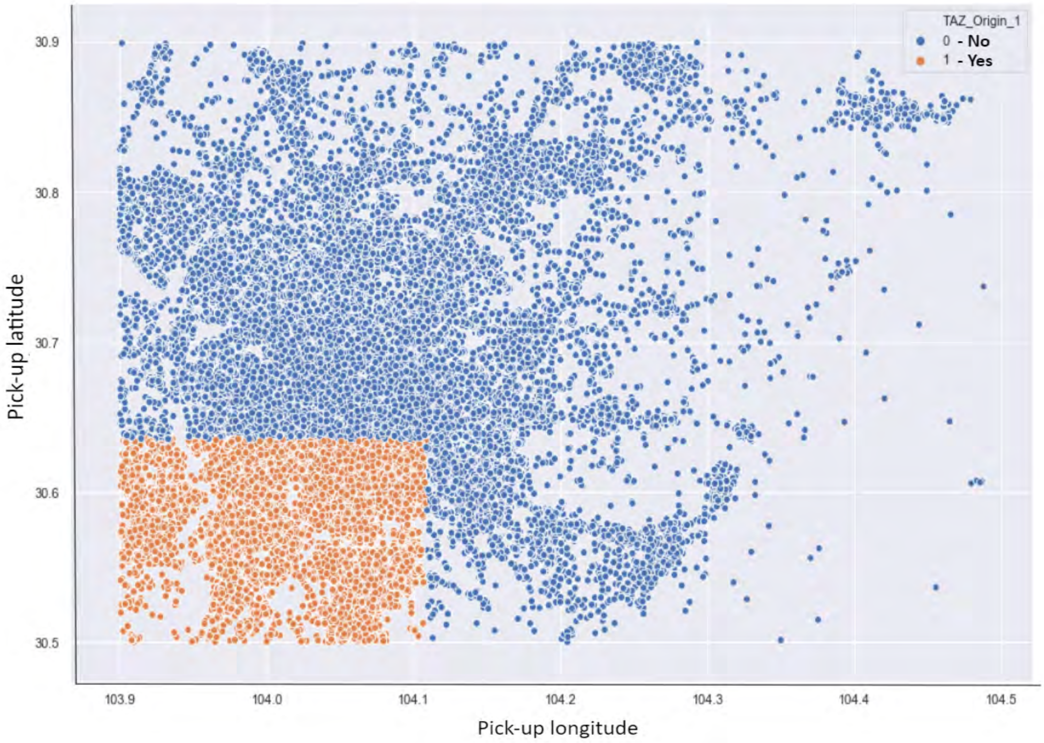

Figure 36: Traffic Analysis Zone 1

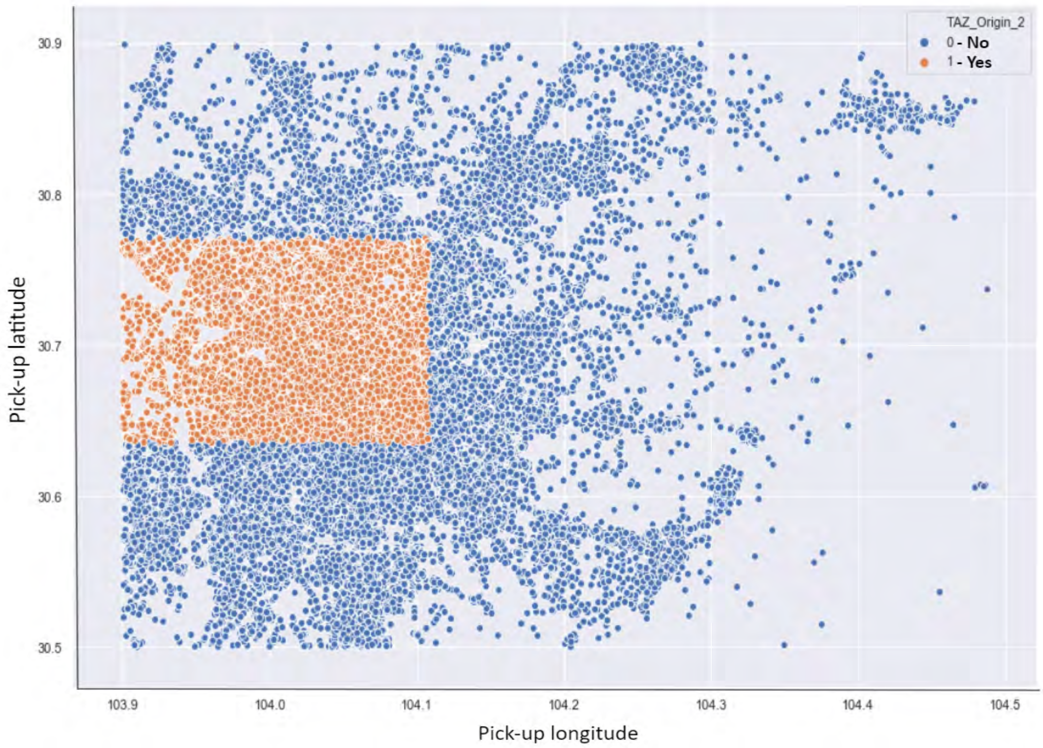

Figure 37: Traffic Analysis Zone 2

Figure 38: Traffic Analysis Zone 3

Figure 39: Traffic Analysis Zone 4

Figure 40: Traffic Analysis Zone 5

Figure 41: Traffic Analysis Zone 6

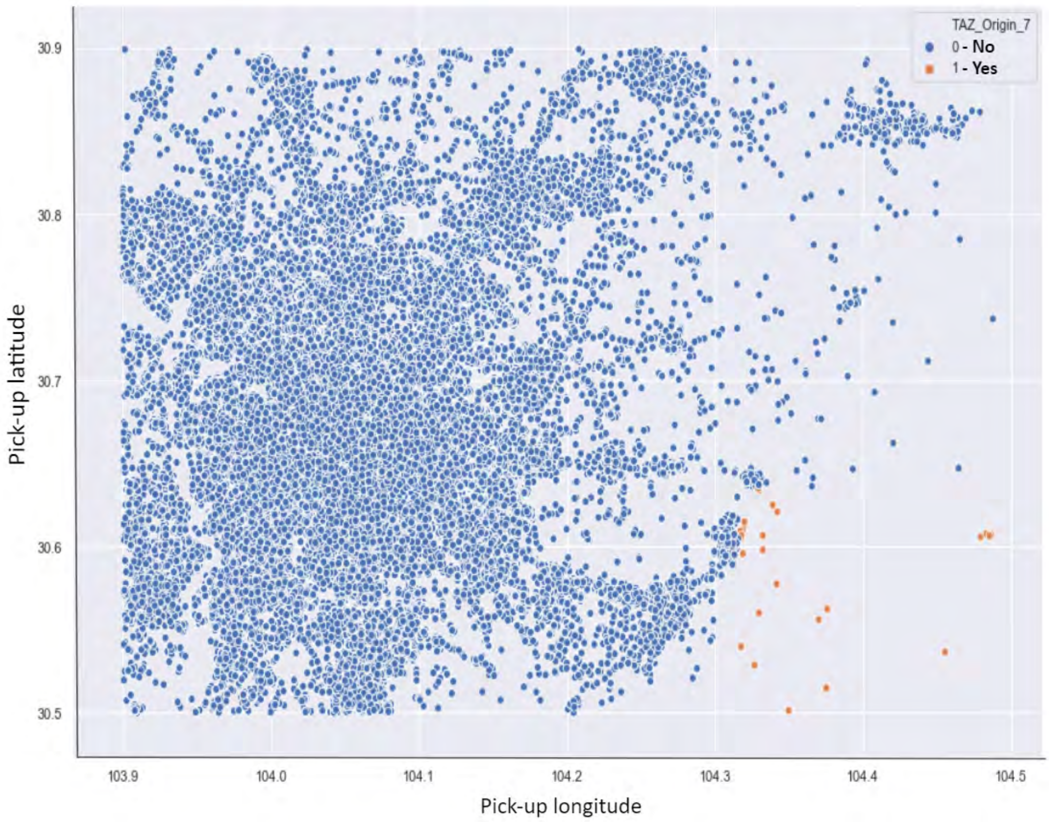

Figure 42: Traffic Analysis Zone 7

Figure 43: Traffic Analysis Zone 8

Figure 44: Traffic Analysis Zone 9

The Traffic Analysis Zones (TAZ) study the relevance of regional segments in influencing the travel time. The study area has 9 zones containing trip departures and arrivals. These zones have the idea or the concept of cardinal points, including North, East, West, South, Centre, North-East, North-West, South-East, and South-West zones. Their correlation values identify which zone is relevant or highly significant to travel time and essential for modeling. These zone plots also reveal that large parts of the trips fall in zone 1-6, and these zones have a scale of 15km by 15km on the x-axis (length) and y-axis (width). There are 18 indicators derived from the nine traffic analysis zones, containing nine origin and nine destination zones, separately.

Conclusively, a maximum of sixty (64) indicators or features are relevant and strongly influence travel time. The indicators are inclusive of the ten (10) primary data-driven indicators discovered in previous research works. The statistical descriptions and distribution densities ascertain their significance level, which serves as confirmation for the procedures used in identifying them as indicators capable of influencing travel time. These discussions also answer and justify the first research question for this thesis to fulfill the objective of identifying relevant indicators. Moreover, highly accurate prediction models are dependent on these relevant indicators to be labeled as inputs for the deep learning models.

4.2 Developing deep learning models for travel time prediction

In this section, the deep learning models developed to predict travel time are discussed, and each model performance is compared, both with validation and test data sets. As earlier discussed in previous sections, the training data is the origin-destination of on-demand ride request data collected while the test data is the trip routes data from the commuter's origin to their destination. The training data set is divided into two sets, the training and validation data set with percentages of 70% and 30%, respectively. The three (3) different neural networks are Artificial Neural Network (ANN), Convolutional Neutral Network (CNN), and a Recurrent Neural Network (RNN) called Long Short-Term Memory (LSTM). Each neural network technique has five models developed with different labels or indicators but with similar network parameters.

The parameters and measuring tools for the different neural network types are standard across, one of which are epochs denoting the number of iterations the models undergo to learn and train itself. The optimizer, loss function, activation function, number of hidden layers, the neurons (units), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared score or residual error are all model parameters and evaluation metrics. The majority of these parameters and performance measurement tools were discussed in the methodology chapter.

There are five models developed for each of the three neural networks to examine the importance of the various set of indicators or inputs already identified. The collection of inputs for the model categories are Model 1, Model 2, Model 3, Model 4 and Model 5, under ANN, CNN and LSTM while the output feature is travel time in minutes.

Model 1 inputs

Table 6: The labels or inputs of the deep learning model 1

Deep learning model	Model inputs	Description
Model 1	Pick-up Longitude	Longitudinal coordinate of trip origin
	Drop-off Longitude	Longitudinal coordinate of trip destination
	Ride Start Hour	Commuter's hour of departure
	Pick-up Latitude	Latitudinal coordinate of trip origin
	Drop-off Latitude	Latitudinal coordinate of trip destination
	Ride Start Minute	Commuter's minute of departure
	Ride Start Month	Commuter's month of departure
	Ride Start Day	Commuter's day of departure
	Trip Distance	Distance travelled during the trip

Model 2 inputs

Table 7: The labels or inputs of the deep learning model 2

Deep learning model	Model inputs	Description
Model 2	All model 1	All model 1 inputs are included in model 2
	Distance 0-2km	All commuters' trip distance from 0 to 2 km
	Departure 16-20hr	All commuters' departure time from 4 PM to 8 PM
	Departure 12-16hr	All commuters' departure time from 12 PM to 4 PM
	Departure Hour 15hr	All commuters' departure time at 3 PM
	Departure Hour 16hr	All commuters' departure time at 4 PM
	Departure Hour 14hr	All commuters' departure time at 2 PM
	Departure Hour 17hr	All commuters' departure time at 5 PM
	Departure 20-24hr	All commuters' departure time from 8 PM to 12 AM
	Departure Hour 18hr	All commuters' departure time at 6 PM
	Departure 45-59min	All commuters' departure time from 45th to 59th minute of the clock per hour
	Distance 2-4km	All commuters' trip distance from 2 to 4 km
	Departure Hour 20hr	All commuters' departure time at 8 PM
	Departure Hour 19hr	All commuters' departure time at 7 PM
	Departure Hour 22hr	All commuters' departure time at 10 PM
	Departure Hour 13hr	All commuters' departure time at 1 PM
	Departure Hour 04hr	All commuters' departure time at 4 AM
	Departure Hour 21hr	All commuters' departure time at 9 PM
	Departure Hour 23hr	All commuters' departure time at 11 PM
	Departure Hour 12hr	All commuters' departure time at 12 PM
	Departure Hour 00hr	All commuters' departure time at 1 AM
	Departure Hour 11hr	All commuters' departure time at 11 AM
	Departure Hour 05hr	All commuters' departure time at 5 AM
	Departure 00-04hr	All commuters' departure time from 1 AM to 4 AM
	Departure 15-30min	All commuters' departure time from 15th to 30th minute of the clock per hour
	Departure 30-45min	All commuters' departure time from 30th to 45th minute of the clock per hour
	Distance 6-8km	All commuters' trip distance from 6 to 8 km
	Departure Hour 02hr	All commuters' departure time at 2 AM
	Departure Hour 07hr	All commuters' departure time at 7 AM
	Departure Hour 10hr	All commuters' departure time at 10 AM
	Departure 04-08hr	All commuters' departure time from 4 AM to 8 AM
	Departure Hour 08hr	All commuters' departure time at 8 AM
	Departure Hour 06hr	All commuters' departure time at 6 AM
	Distance 8-10km	All commuters' trip distance from 8 to 10 km
	Distance 10-12km	All commuters' trip distance from 10 to 12 km
	Distance 18-20km	All commuters' trip distance from 18 to 20 km
	Distance 12-14km	All commuters' trip distance from 12 to 14 km
	Distance 14-16km	All commuters' trip distance from 14 to 16 km
	Departure 08-12hr	All commuters' departure time from 8 AM to 12 PM
	Departure Hour 09hr	All commuters' departure time at 9 AM
	Distance 16-18km	All commuters' trip distance from 16 to 18 km

Model 3 inputs

Table 8: The labels or inputs of the deep learning model 3

Deep learning model	Model inputs	Description
Model 3	All model 2	All model 2 inputs are included in model 3
	TAZ destination-1	Commuters' trip arrival in Traffic Analysis Zone (TAZ) 1
	TAZ destination-2	Commuters' trip arrival in Traffic Analysis Zone (TAZ) 2
	TAZ destination-3	Commuters' trip arrival in Traffic Analysis Zone (TAZ) 3
	TAZ destination-4	Commuters' trip arrival in Traffic Analysis Zone (TAZ) 4
	TAZ destination-5	Commuters' trip arrival in Traffic Analysis Zone (TAZ) 5
	TAZ destination-6	Commuters' trip arrival in Traffic Analysis Zone (TAZ) 6
	TAZ destination-7	Commuters' trip arrival in Traffic Analysis Zone (TAZ) 7
	TAZ Origin-1	Commuters' trip departure from Traffic Analysis Zone (TAZ) 1
	TAZ Origin-2	Commuters' trip departure from Traffic Analysis Zone (TAZ) 2
	TAZ Origin-3	Commuters' trip departure from Traffic Analysis Zone (TAZ) 3
	TAZ Origin-4	Commuters' trip departure from Traffic Analysis Zone (TAZ) 4
	TAZ Origin-5	Commuters' trip departure from Traffic Analysis Zone (TAZ) 5
	TAZ Origin-6	Commuters' trip departure from Traffic Analysis Zone (TAZ) 6
	TAZ Origin-7	Commuters' trip departure from Traffic Analysis Zone (TAZ) 7

Model 4 and 5 inputs

Table 9: The labels or inputs of the deep learning model 4 and 5

Deep learning model	Model inputs	Description
Model 4	All model 3	All model 3 inputs are included in model 4
	Trip SpeedKMH	Speed travel during commuters' trips in km/h
Model 5	All model 1	All model 1 inputs are included in model 5
	Trip SpeedKMH	Speed travel during commuters' trips in km/h

Tables 6, 7, 8 and 9 contain the inputs of the five models, with their respective descriptions. These models have distinct inputs and varying quantities. All Model 1 has nine inputs each, and model 2 has 49 inputs each. The 3rd models have 63 inputs, 4th models comprise 64 inputs, and the 5th models have 10 inputs. The essence of the diverse set of inputs in the models is to compare and identify the essential inputs and indicators significant in training models for high travel time prediction accuracy.

Each model has an output of travel time, and the results of these models are discussed in detail including the training, validation, and testing outcomes of the models for ANN, CNN, and LSTM. As a result of the high computer memory storage required to compile and process the whole 30 days trip routes of data collected, the models' test use only the first seven days. The data processing complexity results from the massive raw test data set of about 45 million rows per day, which requires a high capacity computer with high Random Access Memory (RAM) and other computation needs.

4.2.1 Artificial Neural Network (ANN) model results

Model training and validation

Table 10: Artificial Neural Network (ANN) models trained for travel time prediction

ANN Models	Epochs/ Iterations	Optimizer	Loss Function	Activation Function	Hidden Layers	Neurons/ Units	MAE (min)	RMSE (min)	Coefficient of determination (R^2)	Train/ Validation (%)
Model 1	50	Adam	MAE	ReLU	4	9	5.123	7.935	0.637	70/30
Model 2	50	Adam	MAE	ReLU	4	49	4.649	7.285	0.694	70/30
Model 3	50	Adam	MAE	ReLU	4	63	4.630	7.629	0.664	70/30
Model 4	50	Adam	MAE	ReLU	4	64	0.212	0.769	0.997	70/30
Model 5	50	Adam	MAE	ReLU	4	10	0.319	0.913	0.995	70/30

The outcome of model 1, as shown in Table 10, undergoes 50 iterations with four hidden layers and nine neurons per hidden layer using Adam optimizer, ReLU activation function and Mean Absolute Error (MAE) loss function. On average, Model 1 has MAE of 5.123 minutes, RMSE of 7.935 minutes, and R-squared(R^2) score of 0.637, which are the values for inputs or features used in previous research works but excluding trip speed. Even after adding other features like distance range, departure minute, and departure period of the day, the model improvement was a little up to Model 3. For the MAE and RMSE, the closer the values to zero, the better, which also mean lower values denote increased accuracy of the model performance and reduced error difference between predicted values and actual values. The R-squared score needs to get closer to 1 to determine models with high performance.

Among the models without travel speed as an input, only model 2 is highly performing with R-squared and RMSE values of 0.694 and 7.285 minutes, while the model's MAE is slightly higher compared to the MAE of model 3, with 4.649 minutes and 4.630 minutes, all respectively. However, the addition of speed in model 4 and 5 change the models' performance drastically. The model R-squared score increased to a maximum of 0.997, the RMSE reduced to a minimum of 0.769 minutes, and MAE leveled to 0.212 minutes for Model 4, and this makes the 4th model the highest performing model.

Figure 45: Artificial Neural Network (ANN) model 1 training and validation loss curves

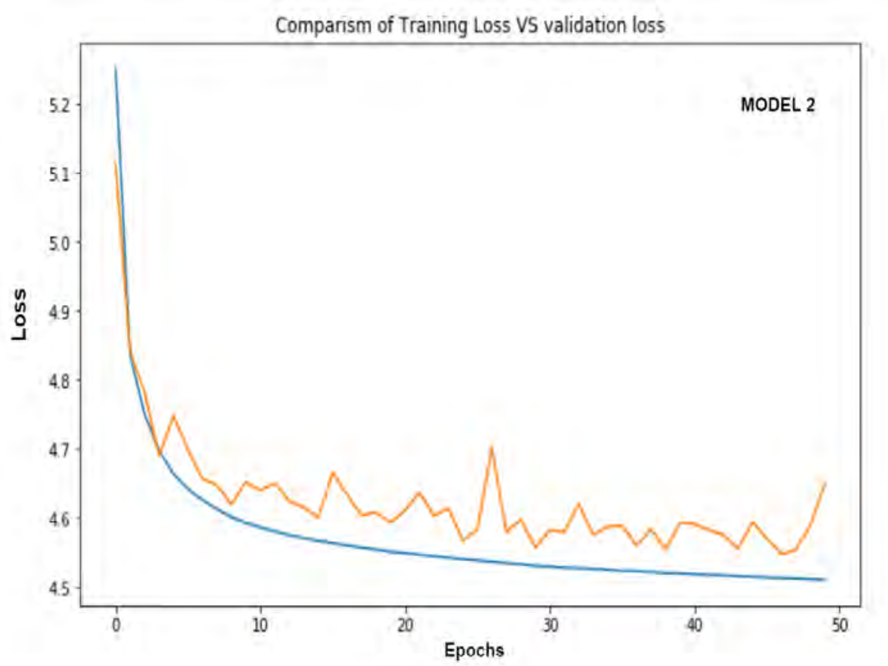

Figure 46: Artificial Neural Network (ANN) model 2 training and validation loss curves

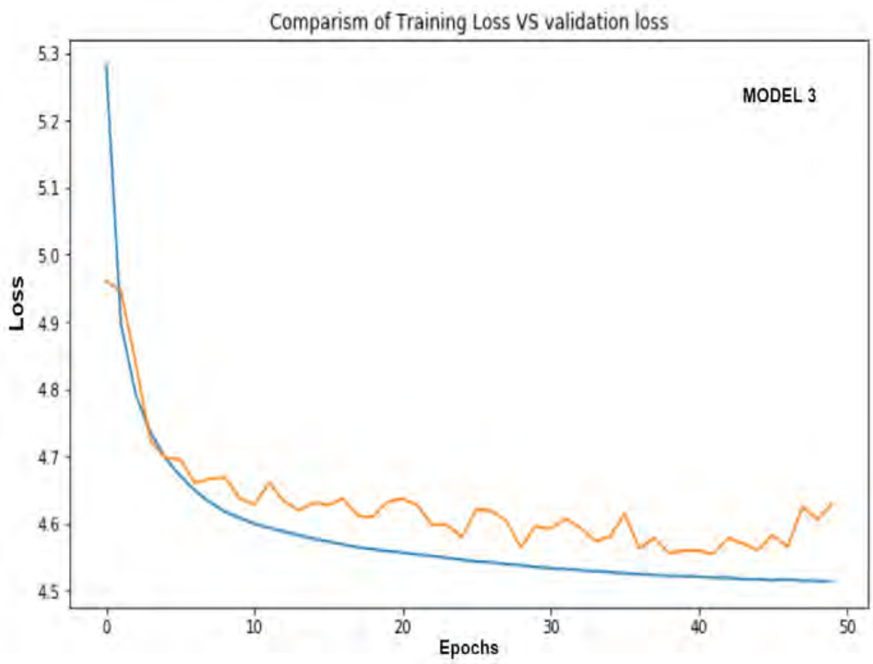

Figure 47: Artificial Neural Network (ANN) model 3 training and validation loss curves

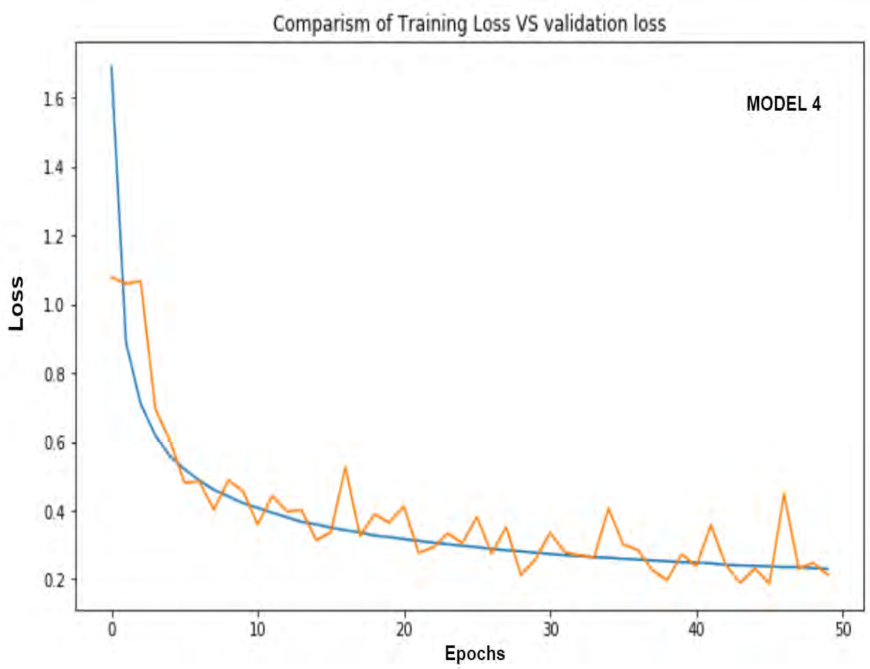

Figure 48: Artificial Neural Network (ANN) model 4 training and validation loss curves

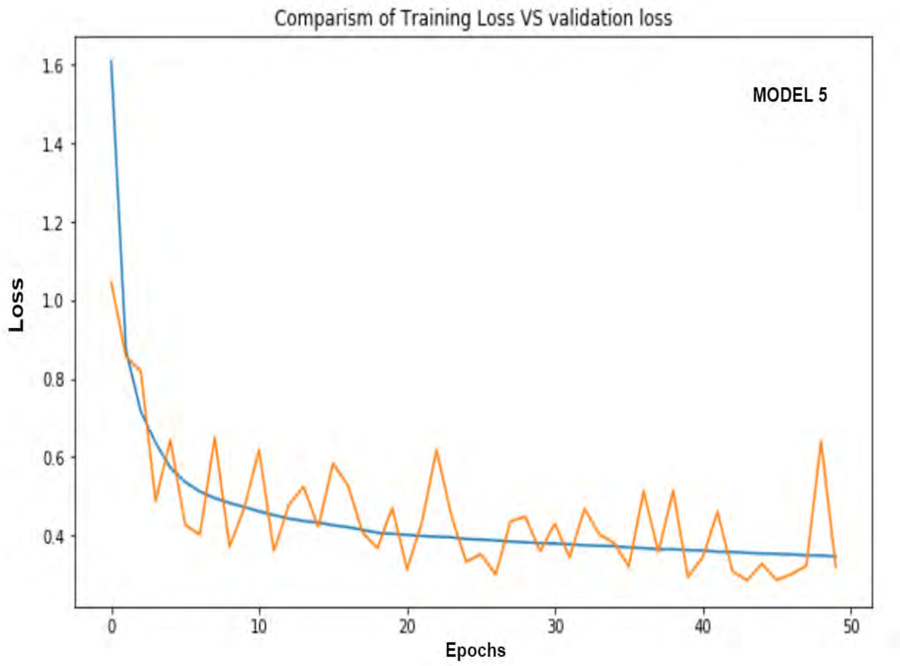

Figure 49: Artificial Neural Network (ANN) model 5 training and validation loss curves

The high performance of model 4 during the training and validation results from the importance of the labeled features. The features are sixty-four (64) in total across all models. The variation in the performance of models 4 and 5 associates with the identification of relevant indicators such as traffic analysis zones, minutes of departure, hours of departure (peak or off-peak), the period of departure during the day (Morning, Afternoon, Evening, Night, mid-night, etc.) and distance traveled ranges of 2 km intervals. In a similar description, the travel speed tends to be the most influential indicator of travel time for demand-responsive trips due to increasing the performance of the models when added to Model 4 and 5 in comparison to Model 1, 2, and 3.

The MAE loss functions are for both training data and validation data, as shown in Figures 45, 46, 47, 48 and 49. These curves aim to check the model's accuracy during the training and validation, confirmed with the stable reduction and convergence of the training loss curve. In contrast, the validation loss curves are found unstable. The instability in the validation loss curves most time means model over-fitting and applying early stop loss at the point of convergence of each curve before spiking is a solution to correct the instability. Also, for models 1-3, the Mean Absolute Error (MAE) losses reduce at every epoch or iteration from 5.4 minutes to less than 4.5 minutes. They mean high accuracy in training the models. However, models 4 and 5 start their MAE loss curves at 1.6 minutes and further reduced up to 0.2 minutes equivalent to 12 seconds. They also show very high model training accuracy.

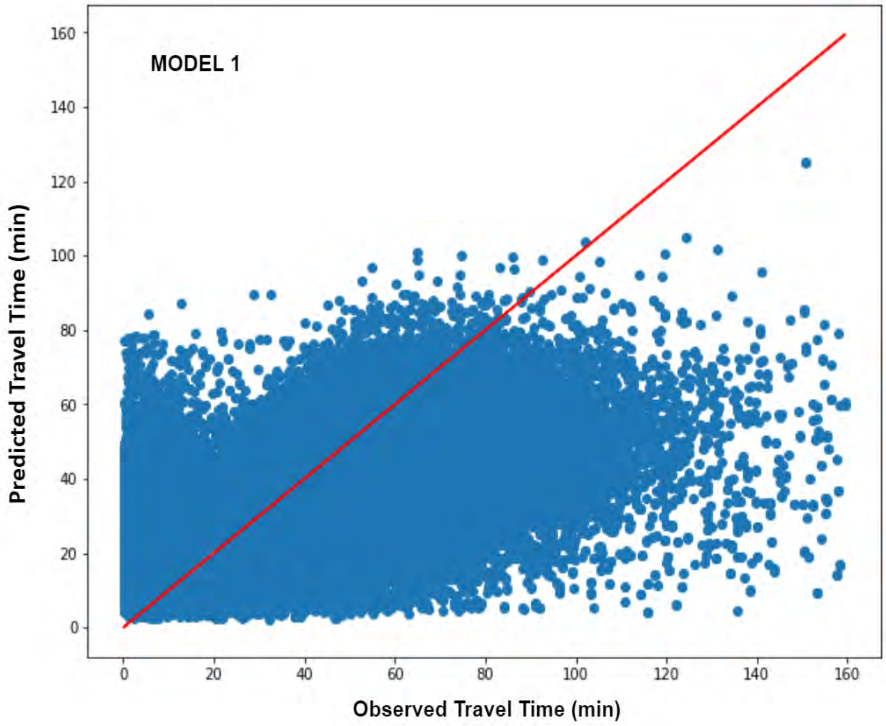

Figure 50: Artificial Neural Network (ANN) model 1 training and validation correlation

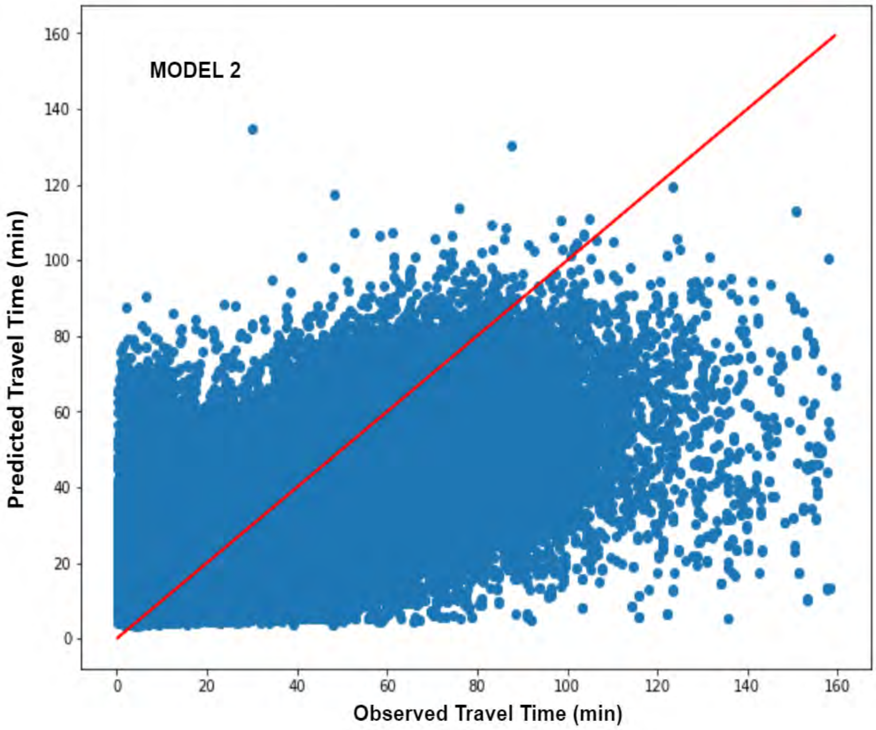

Figure 51: Artificial Neural Network (ANN) model 2 training and validation correlation

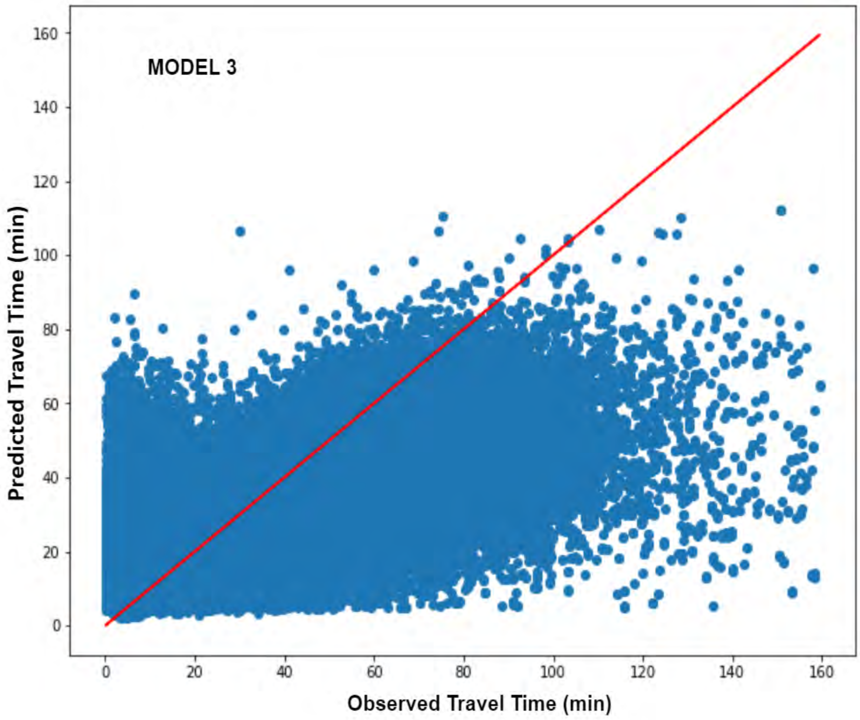

Figure 52: Artificial Neural Network (ANN) model 3 training and validation correlation

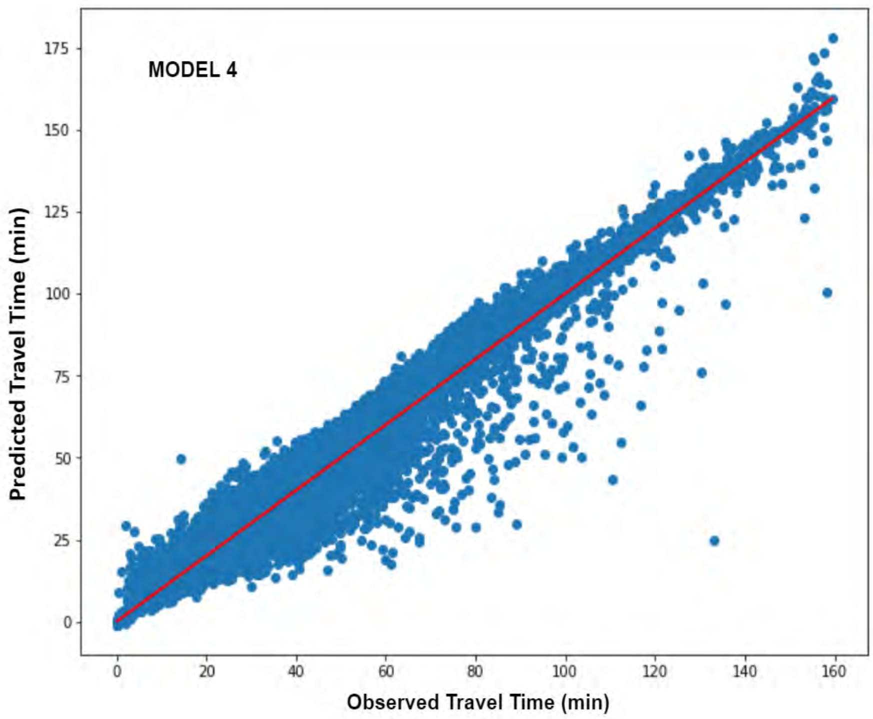

Figure 53: Artificial Neural Network (ANN) model 4 training and validation correlation

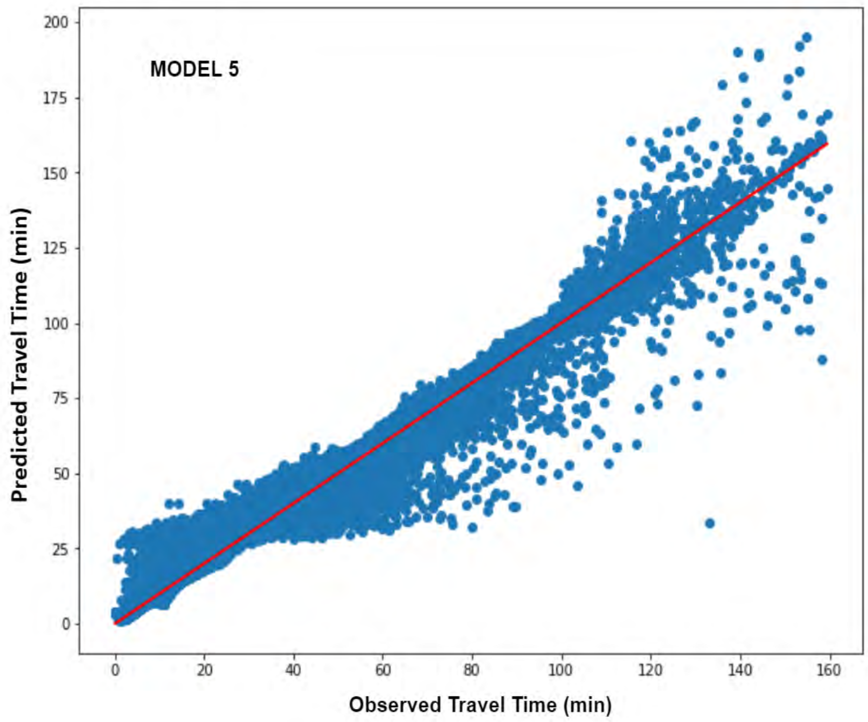

Figure 54: Artificial Neural Network (ANN) model 5 training and validation correlation

Figures 50, 51, 52, 53 and 54 indicate the correlation plot between the validated data set and the trained data set, which have an output of travel time each. The correlation lines for models 1-3 are at 45 degrees amidst the scattered plots, while models 4 and 5 have a majority of their output in high correlation with the red line plots. In essence, this means that the models 4 and 5 have a higher relationship between trained travel time and validated travel time than models 1-3, which also mean higher accuracy in travel time prediction for models 4 and 5.

Furthermore, the error differences between predictions from the validation data and the actual training data show models 1-3 display a more massive error difference from +50 minutes to -50 minutes than models 4 and 5 showing errors between -10 minutes and +10 minutes with the majority of the errors close to zero. It again confirms the very high accuracy of travel time prediction, and model 4 has the highest accuracy in modeling to predict travel time using Artificial Neural Network (ANN).

Model prediction test results

The trained models undergo tests to simulate the accuracy and performance with real-world travel time. The real travel time was collected every 2-4 seconds' coordinate of the commuter's position during the trip, from the origin up to the destination. The result of the test, as shown in Table 11, confirms model 4 as the highest performing Artificial Neural Network model with high prediction accuracy. This outcome results from its R-squared score of 0.997 equivalent to the model training validation R-squared score, the MAE value of 0.093 minutes, and the RMSE value of 0.373 minutes, which are both the lowest with the highest performance in both model validation and model testing.

Table 11: Artificial Neural Network (ANN) model test results for travel time prediction

ANN Models	MAE (min)	RMSE (min)	Coefficient of determination (R^2)
Model 1	3.916	5.422	0.439
Model 2	3.919	5.247	0.491
Model 3	3.448	4.828	0.569
Model 4	0.093	0.373	0.997
Model 5	0.392	0.965	0.982

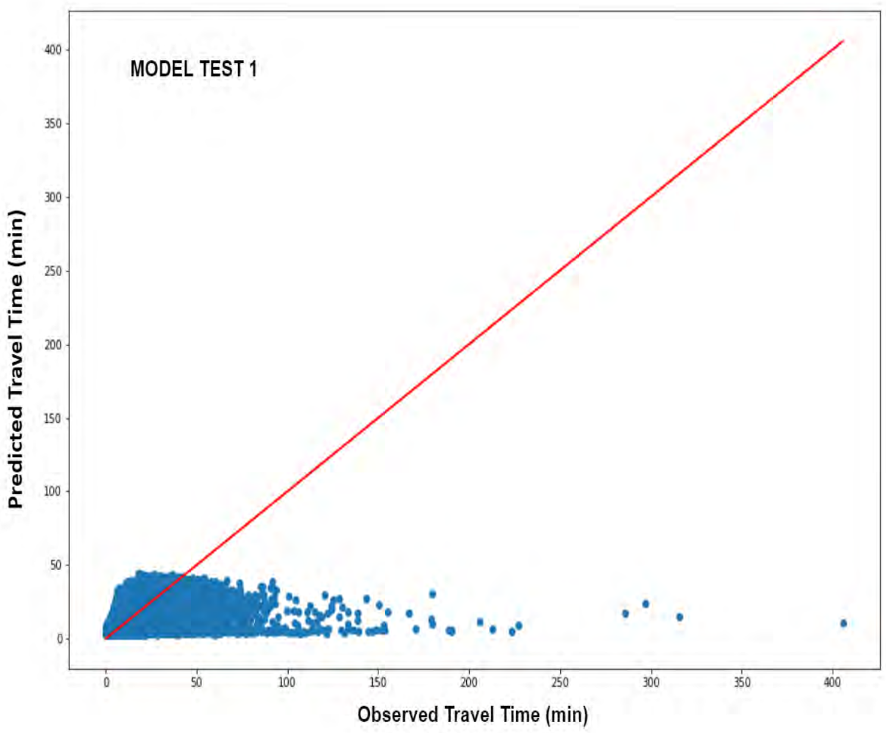

Figure 55: Artificial Neural Network (ANN) model 1 test correlation

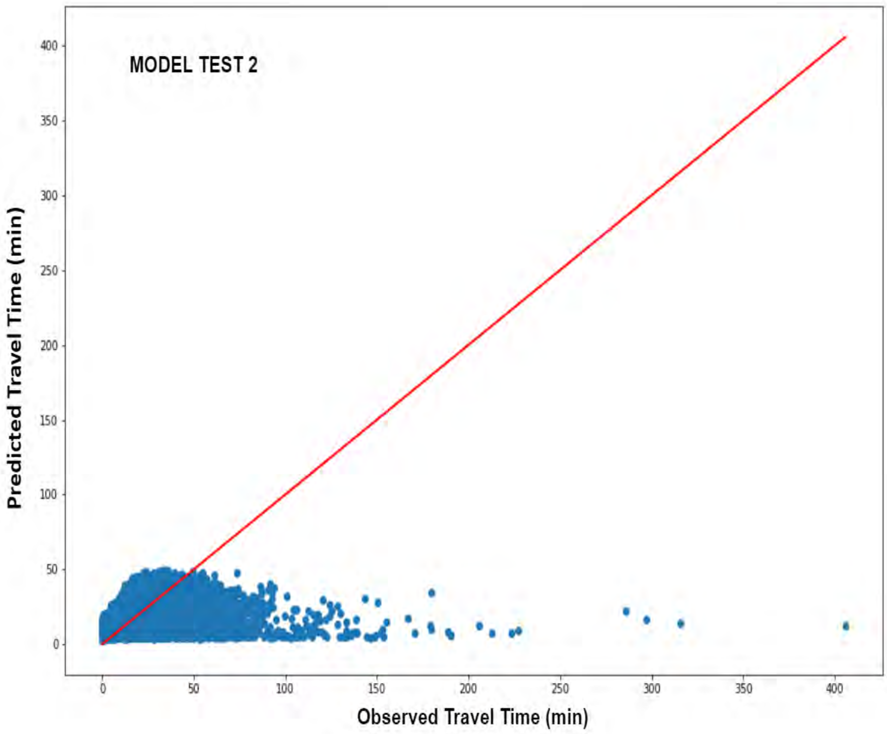

Figure 56: Artificial Neural Network (ANN) model 2 test correlation

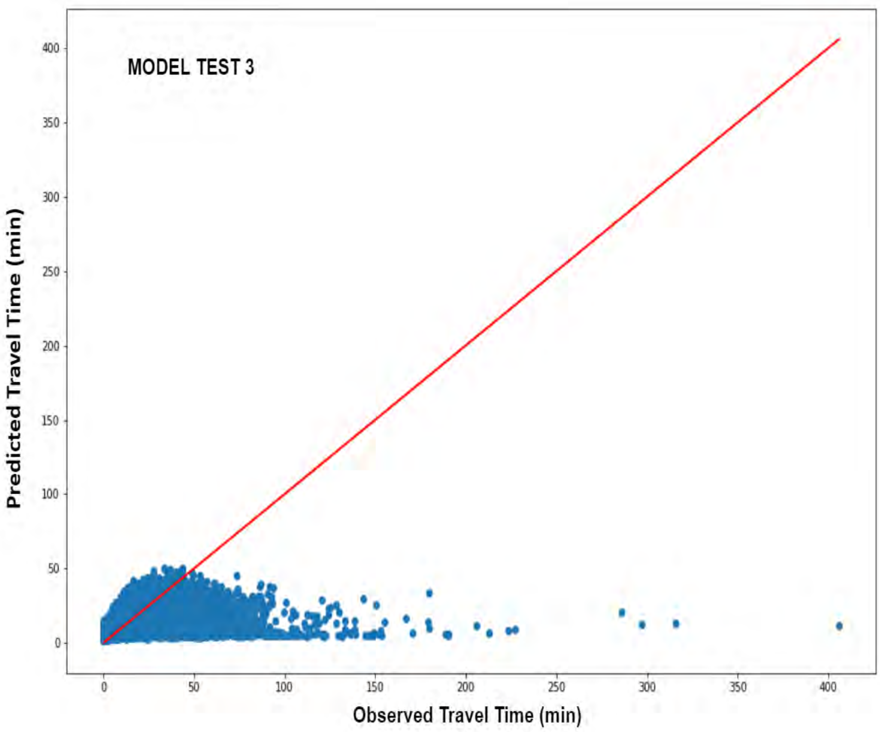

Figure 57: Artificial Neural Network (ANN) model 3 test correlation

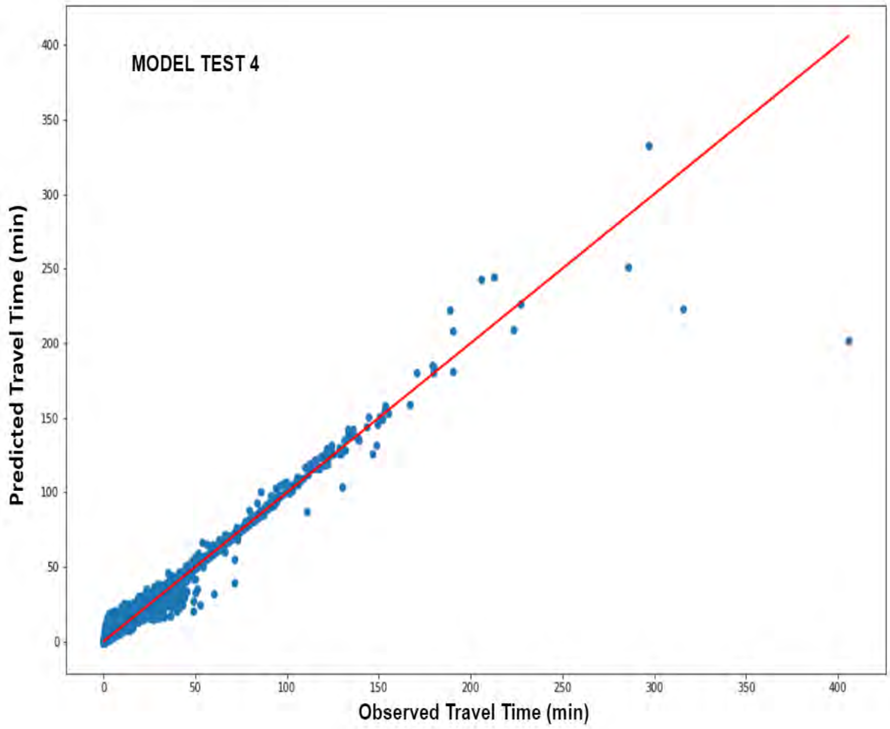

Figure 58: Artificial Neural Network (ANN) model 4 test correlation

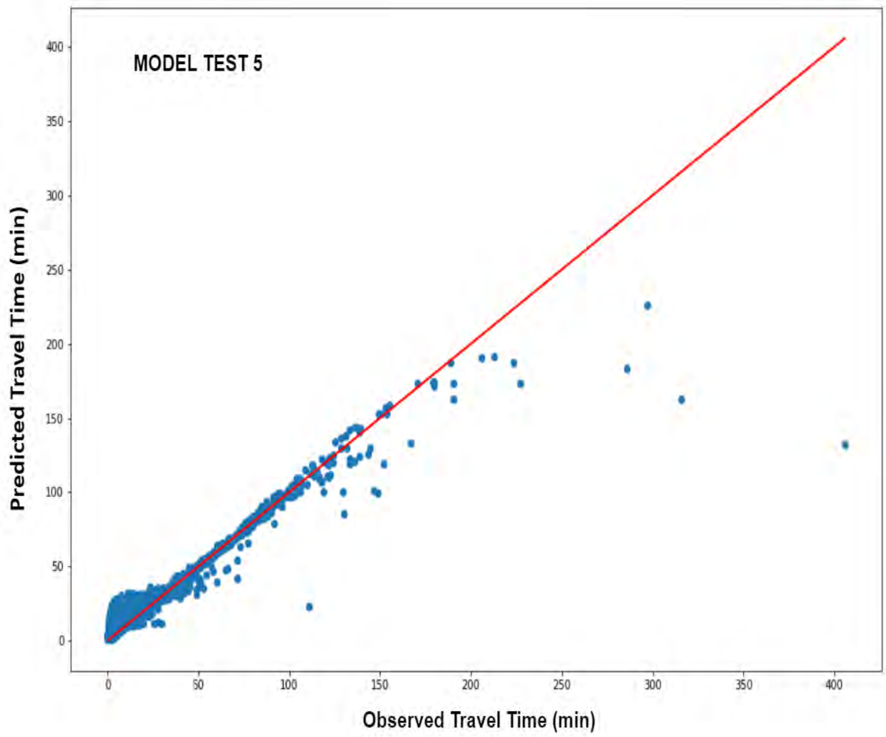

Figure 59: Artificial Neural Network (ANN) model 5 test correlation

To further ascertain the accuracy of the models, the correlations between the predicted travel time and the actual travel time are in Figures 55, 56, 57, 58 and 59 respectively. The correlation figures have substantial differences among models 1-3, and models 4 and 5 tend to have more correlation plots align with the red 45 degrees line than the former. The same goes for the error difference between predicted and actual travel time, models 4 and 5 have virtually all their errors on +/-0 minutes compared to model 1-3.

In essence, the correlations in the models 4 and 5 are more linear than models 1, 2, and 3, and they mean the predicted travel time has a strong relationship with real travel time in models 4 and 5. As the predicted travel time increases, the actual travel time increases, and once the predicted travel time decreases, the real travel time also reduces across the test data. These indicate very low prediction errors and high accuracy in prediction form models 4 and 5.

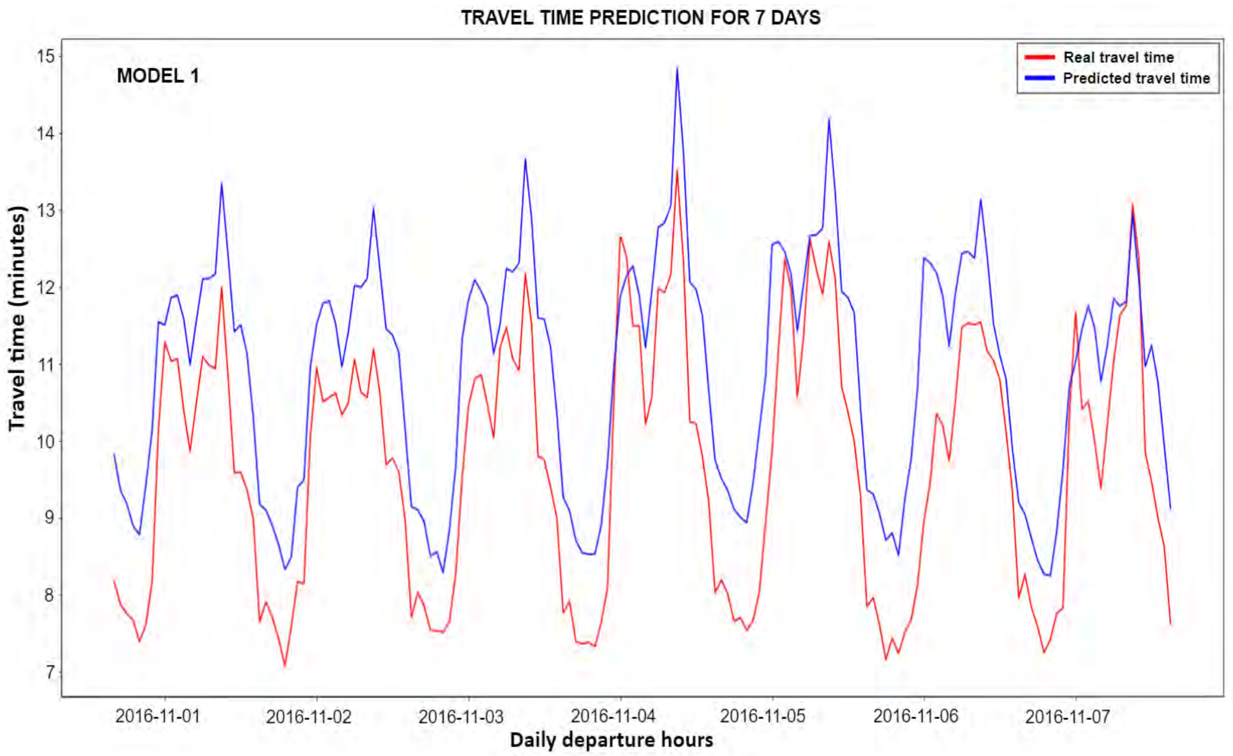

Figure 60: Artificial Neural Network (ANN) models 1 travel time prediction per hour

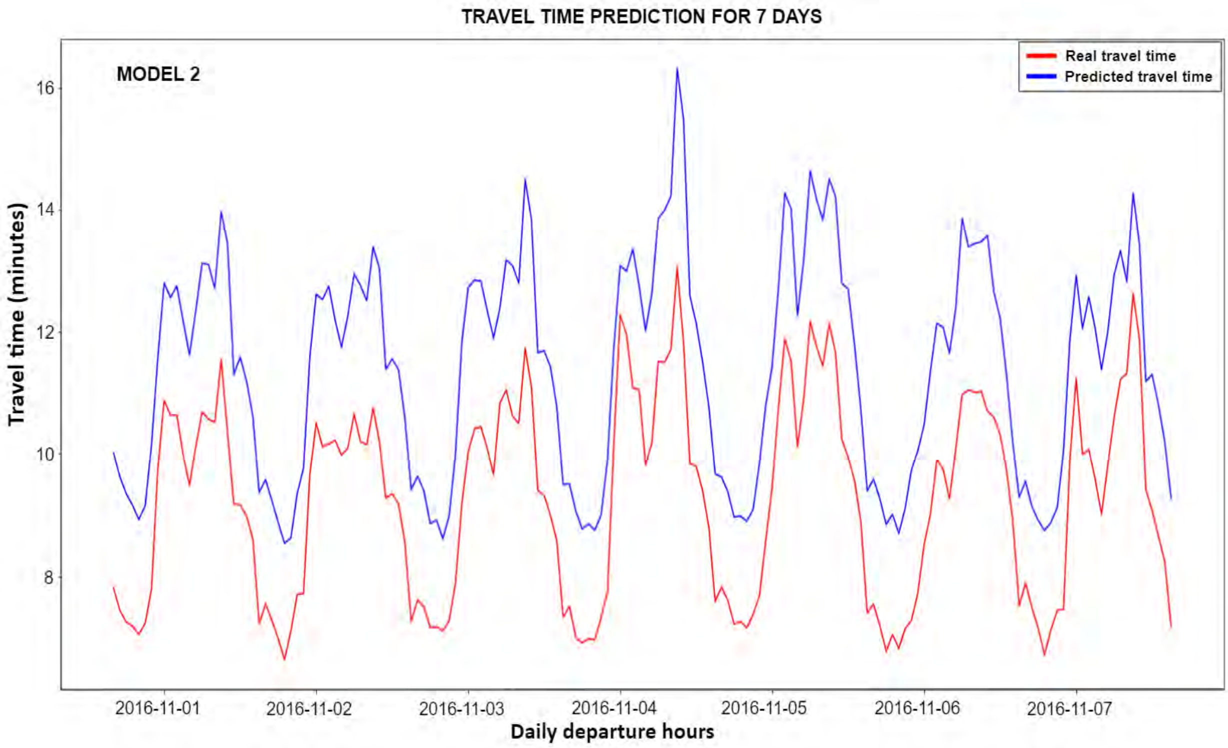

Figure 61: Artificial Neural Network (ANN) models 2 travel time prediction per hour

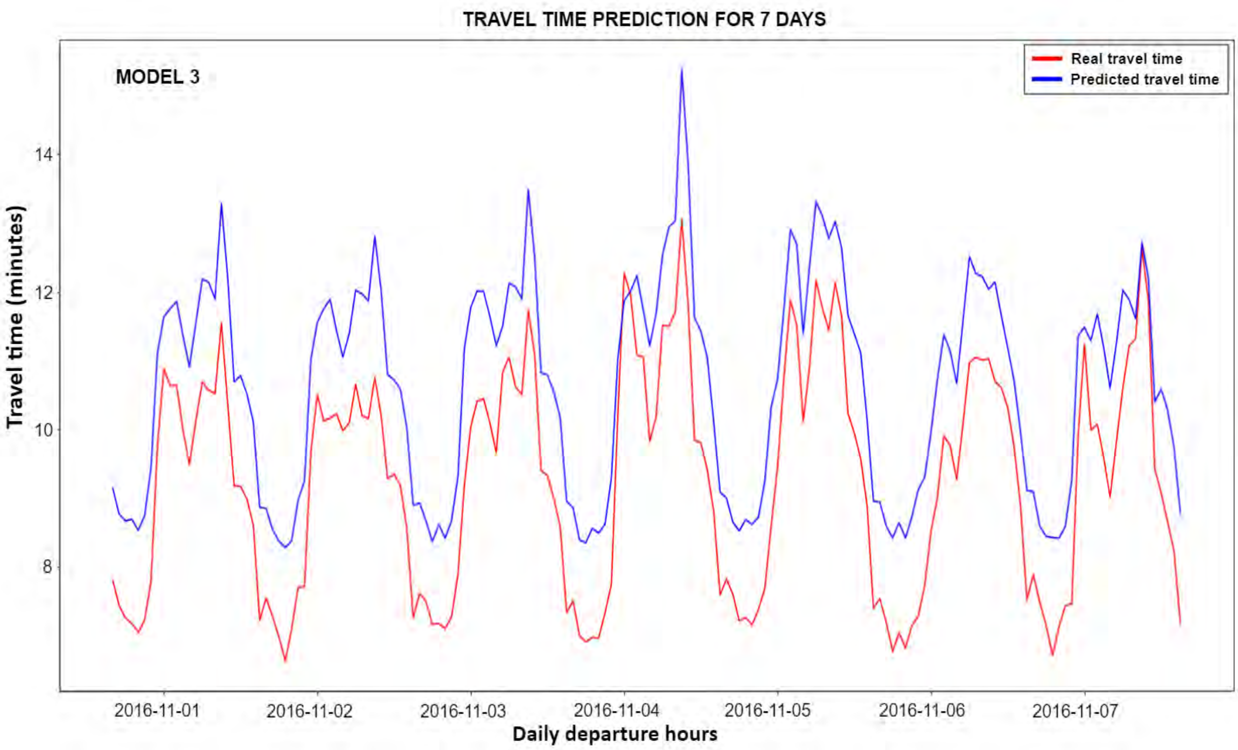

Figure 62: Artificial Neural Network (ANN) models 3 travel time prediction per hour

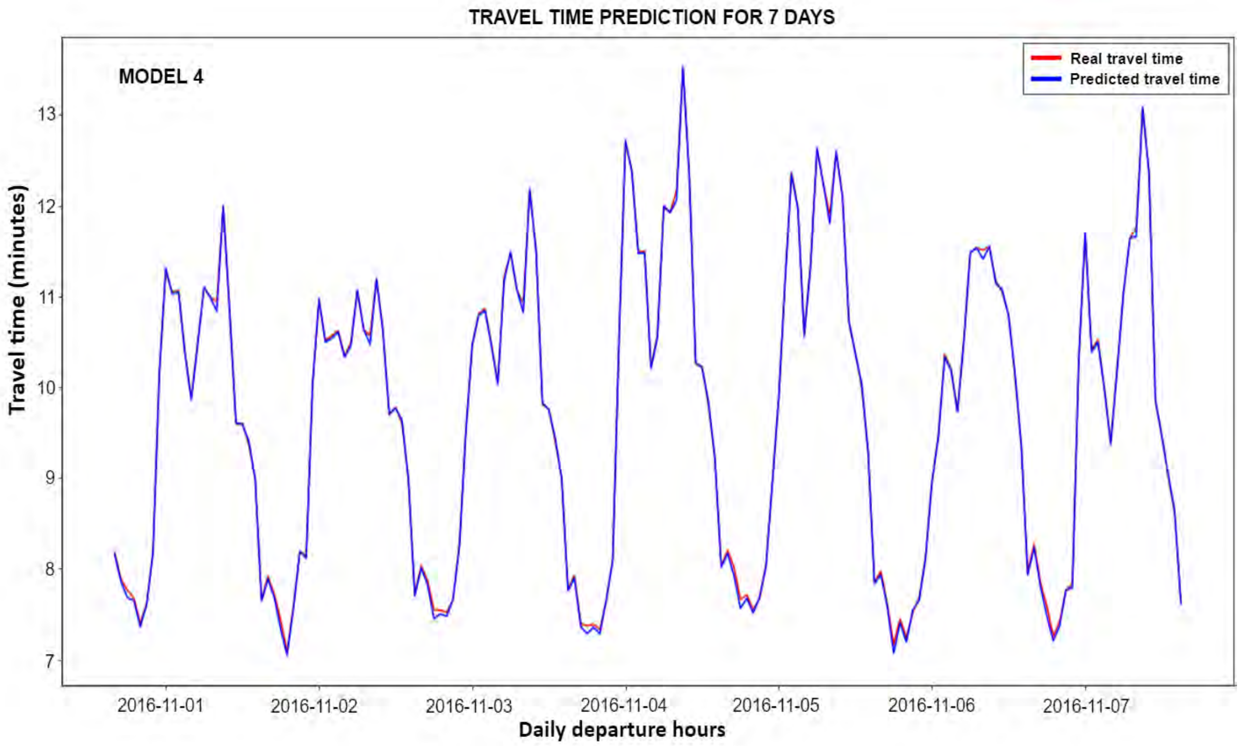

Figure 63: Artificial Neural Network (ANN) models 4 travel time prediction per hour

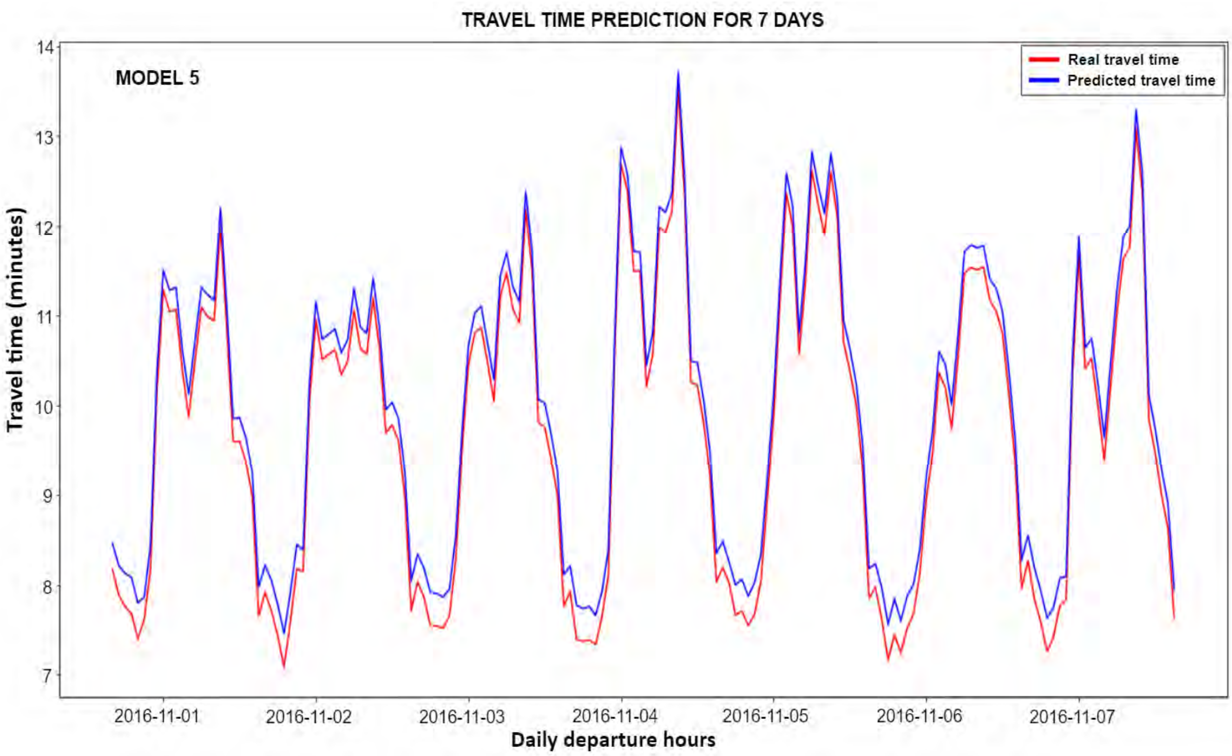

Figure 64: Artificial Neural Network (ANN) models 5 travel time prediction per hour

The results of the travel time prediction for each model in Figures 60, 61, 62, 63 and 64 show model 4 and 5 with blue prediction line is very close to the red actual travel time line, thereby having very high accuracy in travel time prediction. The prediction outcome of model 1-3 shows distant red and blue lines, which denote over-prediction of the travel time. In summary, model 4 shows the highest accuracy for travel time prediction using the Artificial Neural Network (ANN). These prediction plots are for the first seven days of the test data, and the projections are for every hour per day.

4.2.2 Convolutional Neural Network (CNN) model results

Model training and validation

Table 12: Convolutional Neural Network Models Trained for Travel Time Prediction

CNN Models	Epochs/ Iterations	Optimizer	Loss Function	Activation Function	FC Layers	Neurons/ Units	MAE (min)	RMSE (min)	Coefficient of determination (R^2)	Train/ Validation (%)
Model 1	50	Adam	MAE	ReLU	4	9	4.968	7.956	0.635	70/30
Model 2	50	Adam	MAE	ReLU	4	49	4.637	7.438	0.681	70/30
Model 3	50	Adam	MAE	ReLU	4	63	4.650	7.549	0.671	70/30
Model 4	50	Adam	MAE	ReLU	4	64	0.288	0.685	0.997	70/30
Model 5	50	Adam	MAE	ReLU	4	10	0.442	1.224	0.991	70/30

The Table 12 is the result of trained Convolutional Neural Network (CNN) models through 50 iterations with 4 Fully Connected (FC) layers and nine neurons per hidden layer using Adam optimizer, ReLU activation function and Mean Absolute Error (MAE) loss function. Model 1 has MAE of 4.968 minutes, RMSE of 7.956, and an R-squared score of 0.635 without trip speed as input. The model performance improves up to model 3 after including features like distance range, departure minute, and departure period of the day.

Only model 2, among others without trip speed, is high performing with R-squared, MAE and RMSE values of 0.681, 4.637 minutes and 7.438 minutes, respectively. The inclusion of speed in model 4 and 5, rapidly increased the performance of the models. The models recorded maximum value of R-squared score of 0.997, RMSE of 0.685 minutes and MAE of 0.288 minutes for Model 4, and this makes the 4th model, again as in ANN models, the highest performing model. The reason for the high performance is the same as discussed for Artificial Neural Network (ANN) Models. The discussions ascertain trip speed to be the most important indicator for travel time prediction for on-demand trips.

Figures 65, 66, 67, 68 and 69 show the MAE loss function plots for training and validation datasets. The curves examine the model's accuracy during the training and validation by checking the curves' stability in reduction and their convergence. It can be seen that the training loss curves converge and stable while the validation loss curves are found unstable.

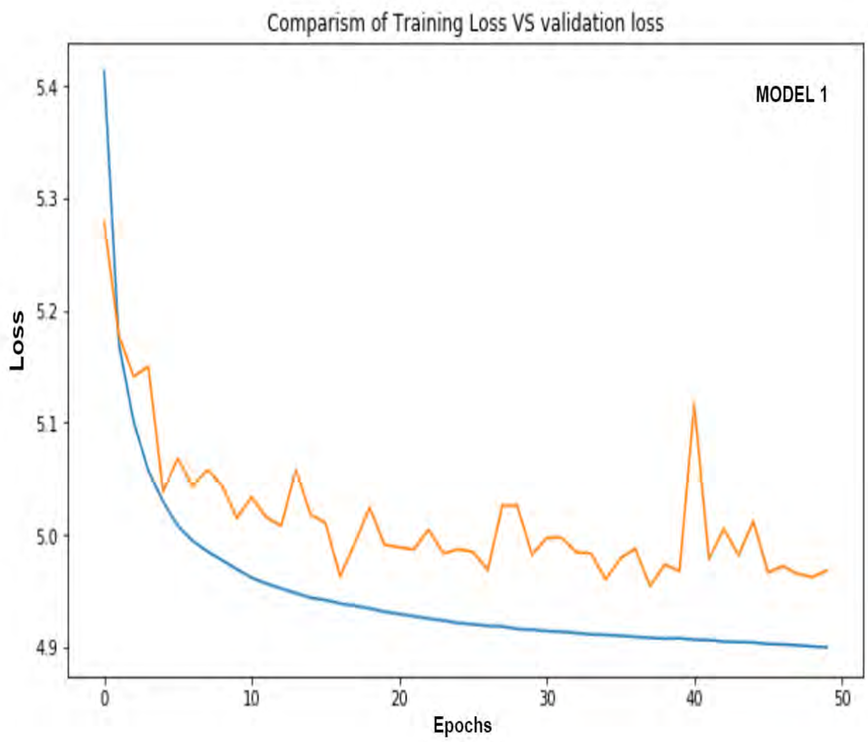

Figure 65: Convolutional Neural Network (CNN) model 1 training and validation loss curves

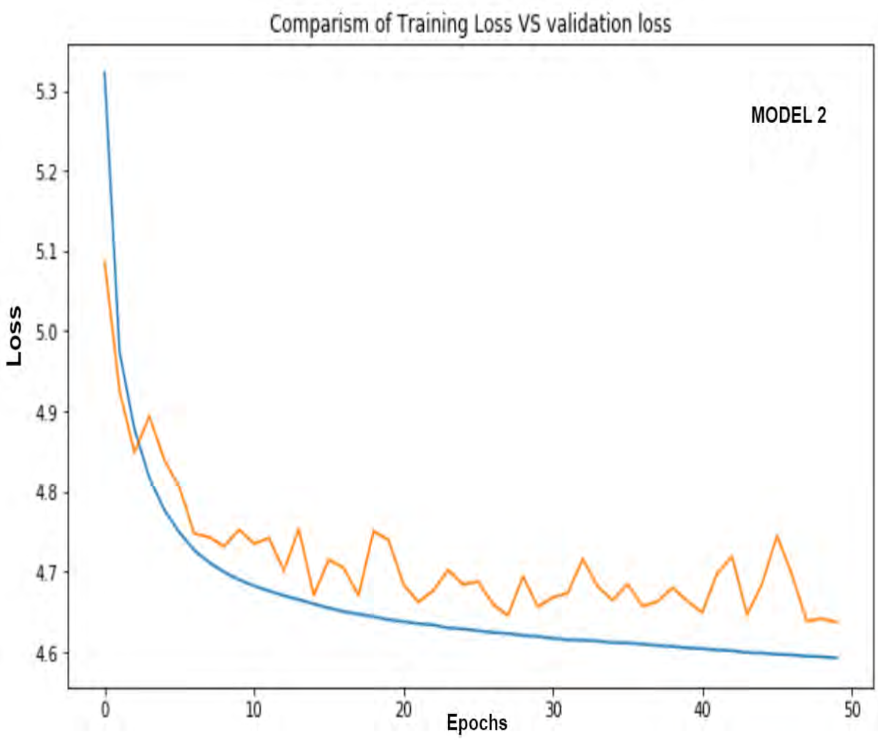

Figure 66: Convolutional Neural Network (CNN) model 2 training and validation loss curves

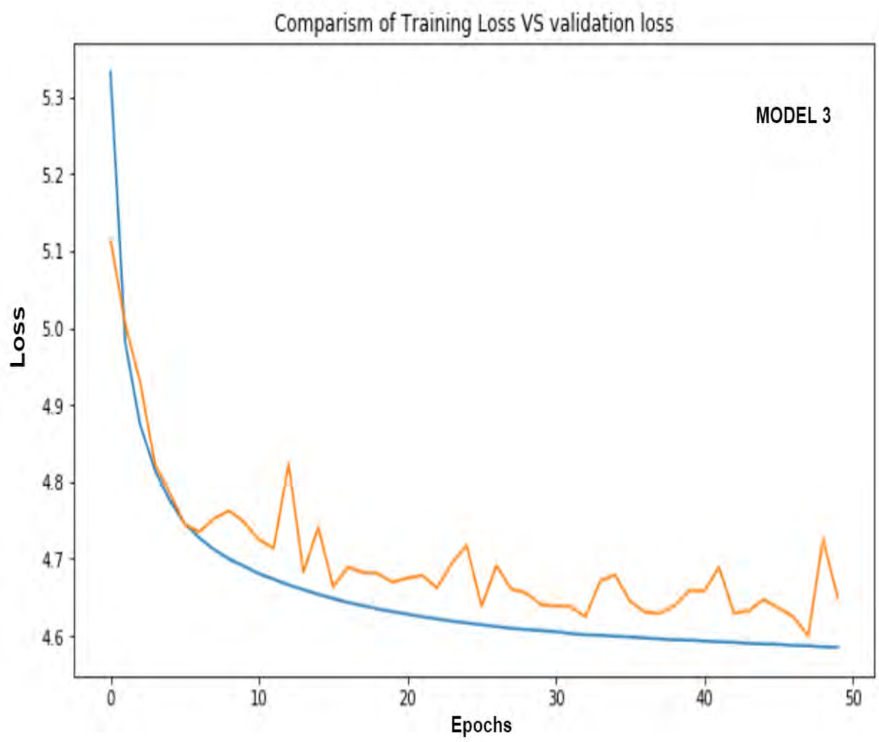

Figure 67: Convolutional Neural Network (CNN) model 3 training and validation loss curves

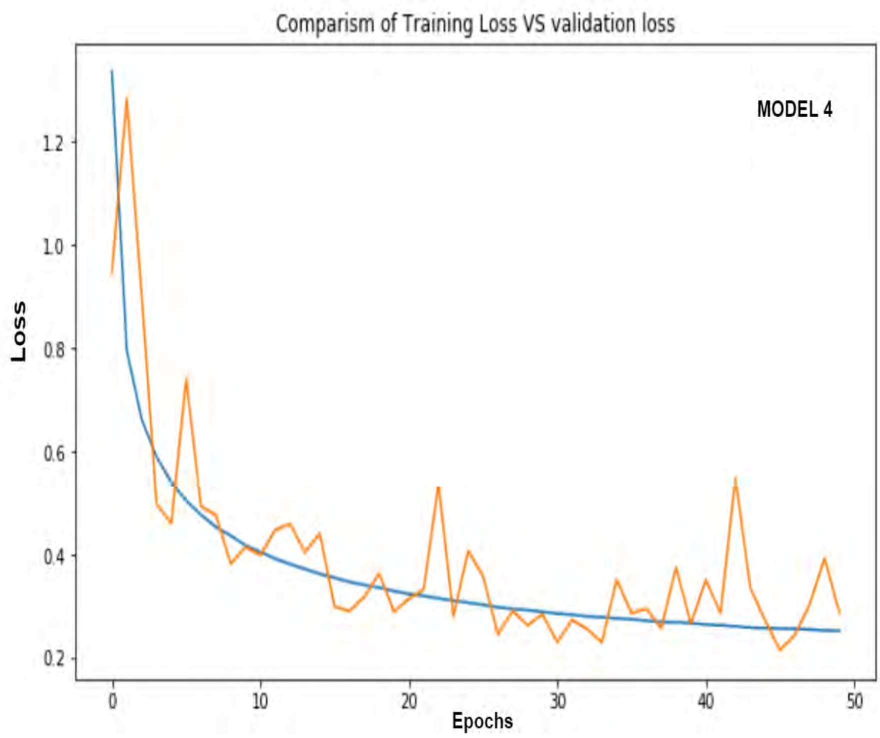

Figure 68: Convolutional Neural Network (CNN) model 4 training and validation loss curves

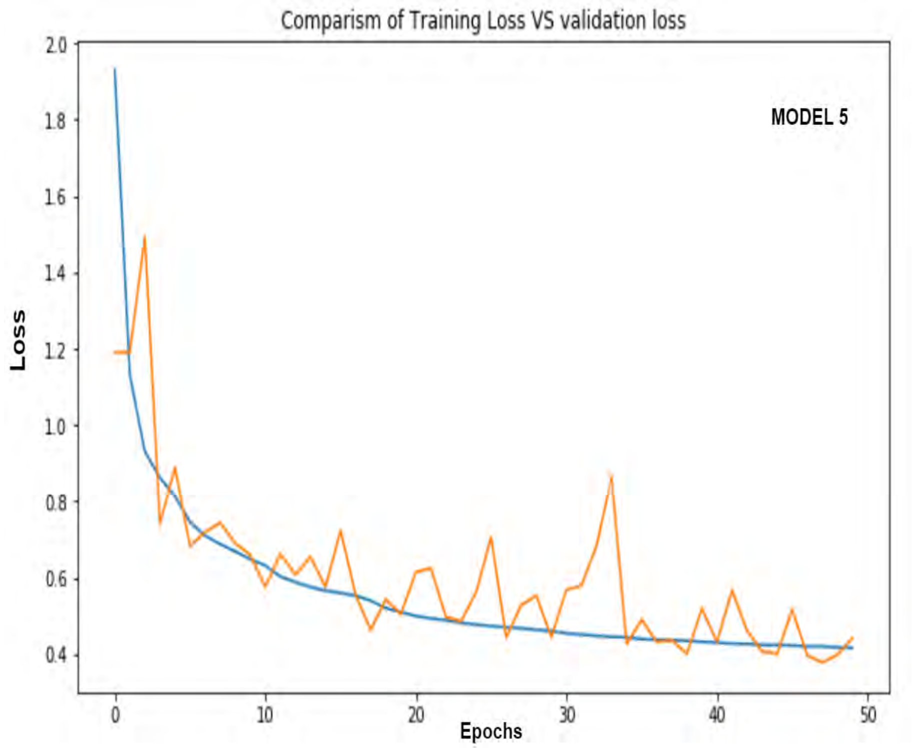

Figure 69: Convolutional Neural Network (CNN) model 5 training and validation loss curves

The instability of the validation loss curves can be corrected by applying early stop loss at the point of convergence of each curve. Mean Absolute Error (MAE) losses for models 1-3 tend to reduce at every epoch or iteration from 5.4 minutes to less than 4.6 minutes, after which they started maintaining flat curves. It shows high accuracy in training the models, and for models 4 and 5, their MAE loss curves started at 2.0 minutes and further reduced up to 0.2 minutes, which confirms very high model training accuracy.

The correlation plots between the validated data and the trained data are as shown in Figures 70, 71, 72, 73 and 74. The models 1-3 correlation lines maintain angles of 45 degrees through the scattered plots, and very high correlations in models 4 and 5, between predicted travel time and actual travel time. Models 4 and 5 have a higher correlation than models 1-3 due to their scattered plots aligning themselves along the red lines, which means higher accuracy in prediction for models 4 and 5. Furthermore, the error differences between predictions from the validation data and the actual training data show large error differences ranging between +50 minutes and -50 minutes in comparison with models 4 and 5 whose error range is approximately between -10 minutes and +10 minutes with the majority of the errors close to zero. This further confirms a very high accuracy of travel time prediction and that model 4 has the highest accuracy using Convolutional Neural Network (CNN) models to predict travel time.

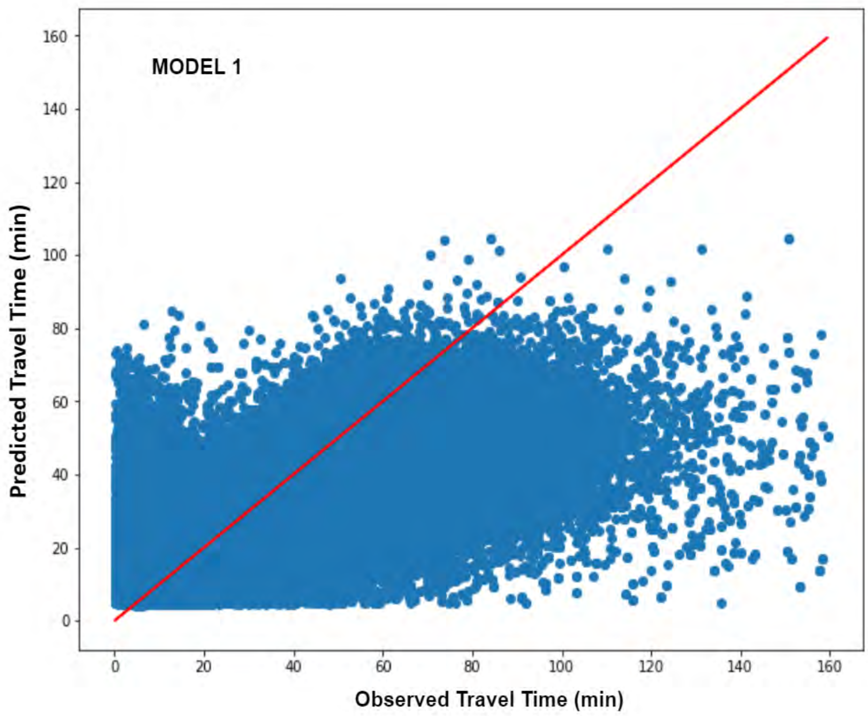

Figure 70: Convolutional Neural Network (CNN) models 1 training and validation correlation

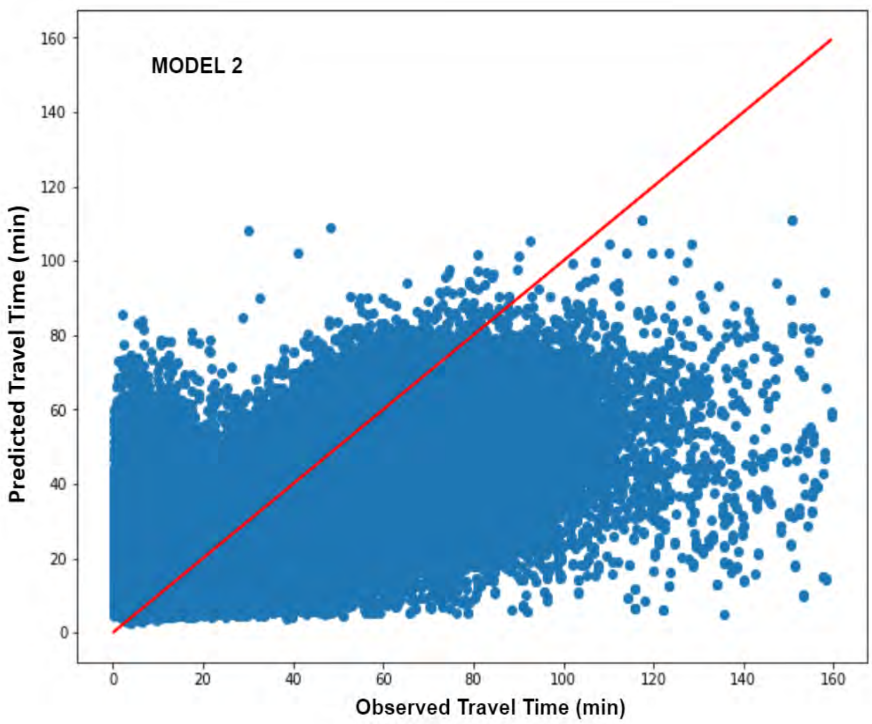

Figure 71: Convolutional Neural Network (CNN) models 2 training and validation correlation

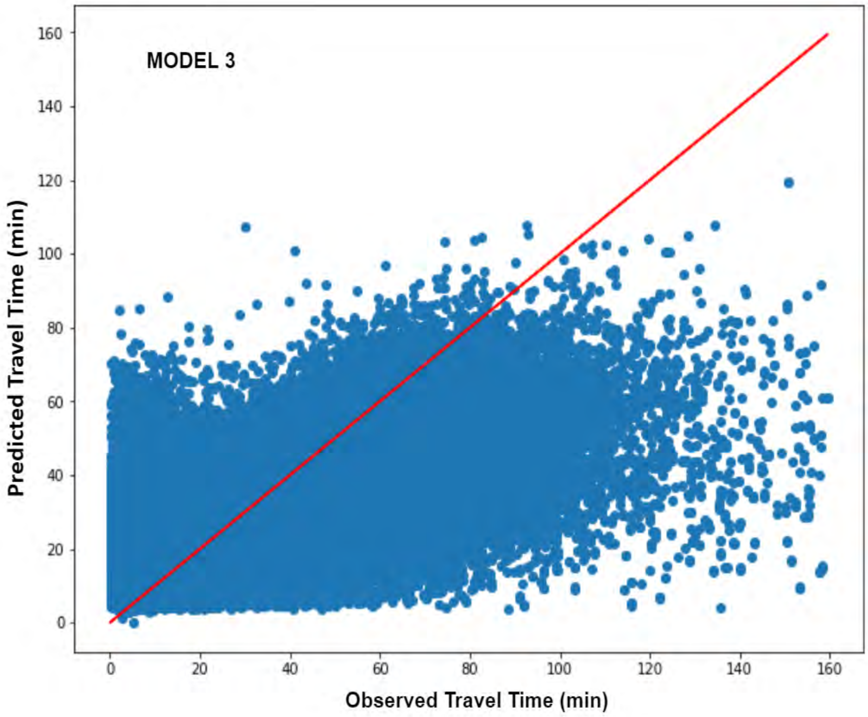

Figure 72: Convolutional Neural Network (CNN) models 3 training and validation correlation

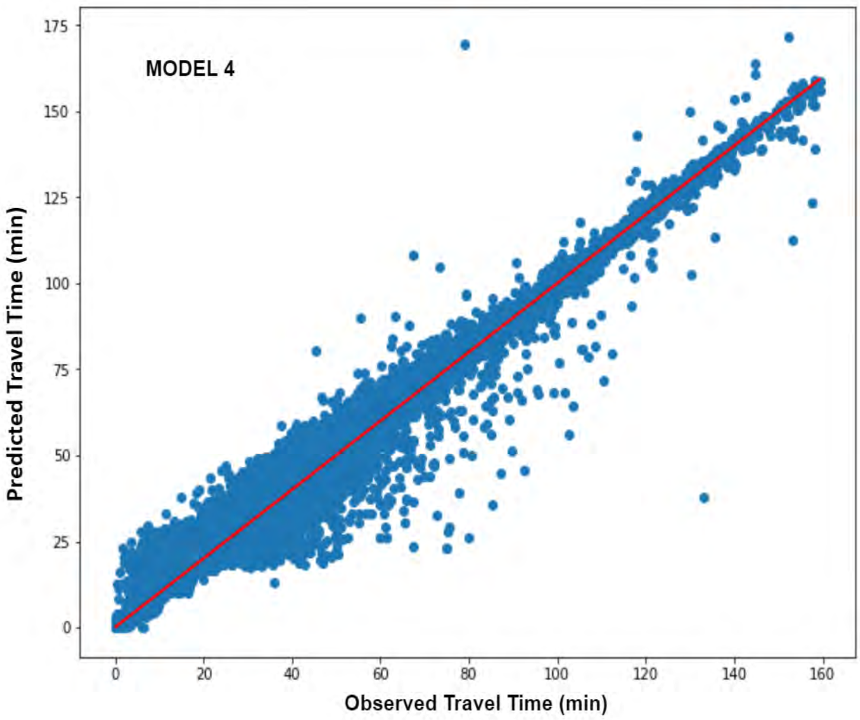

Figure 73: Convolutional Neural Network (CNN) models 4 training and validation correlation

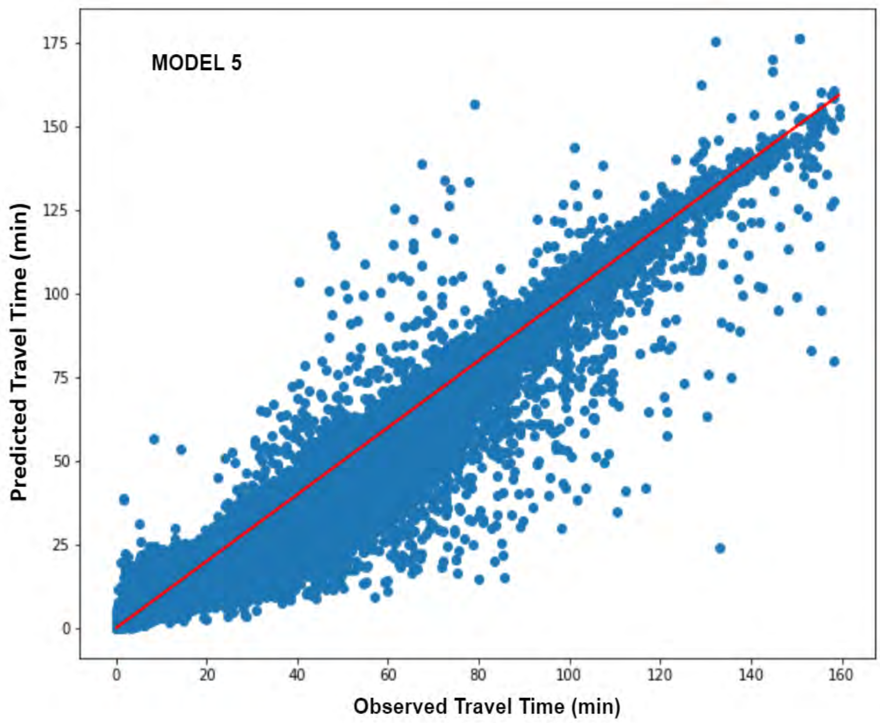

Figure 74: Convolutional Neural Network (CNN) models 5 training and validation correlation

Model prediction test results

Table 13: Convolutional neural network model test results for travel time prediction

CNN Models	MAE (min)	RMSE (min)	Coefficient of determination (R^2)
Model 1	3.505	5.039	0.515
Model 2	3.841	5.118	0.516
Model 3	3.665	4.976	0.542
Model 4	0.197	0.403	0.997
Model 5	0.225	0.575	0.993

To use these models to test the real-world travel time, it is required to carry out tests using the data collected every 2-4 seconds from the coordinates of the commuter's movements during the trip, from the origin up to the destination. The test outcome, as shown in Table 13, confirms model 4 as the highest performing Convolutional Neural Network model with high prediction accuracy. The outcome of model 4 having the highest performance shows an R-squared score of 0.997 equivalent to that of the model training-validation, the MAE value of 0.093 minutes, and the RMSE value of 0.373 minutes, which are both the lowest with the highest performance in both model validation and model testing.

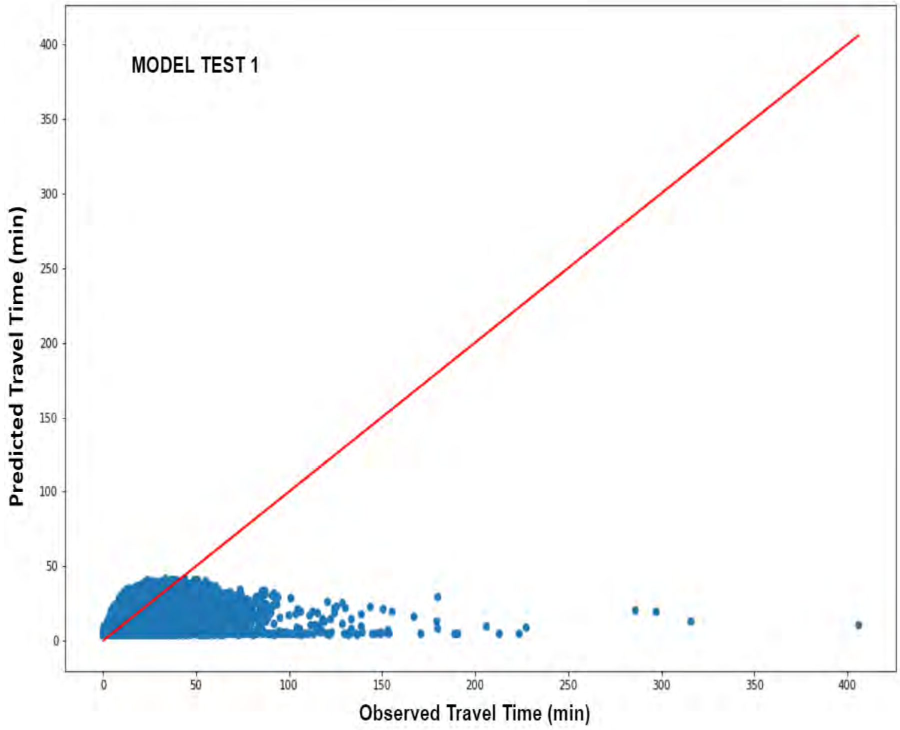

Figure 75: Convolutional Neural Network (CNN) model 1 test correlation

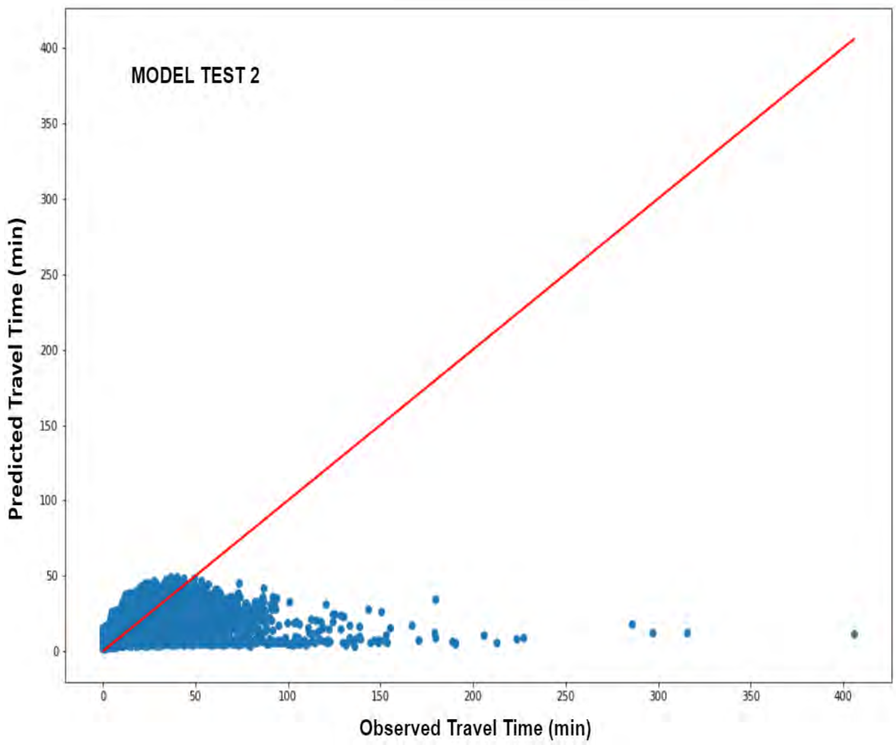

Figure 76: Convolutional Neural Network (CNN) model 2 test correlation

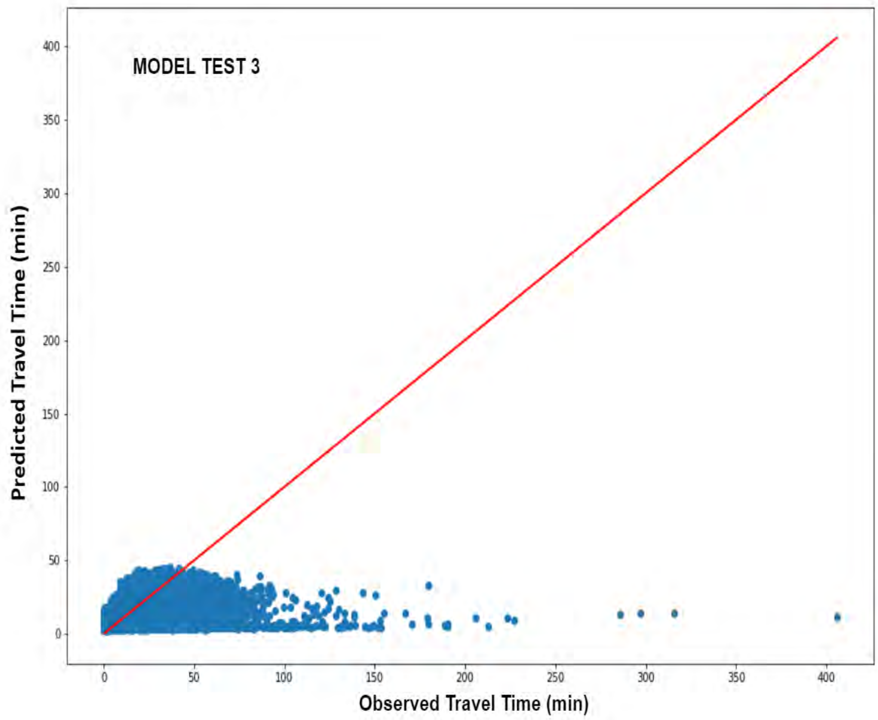

Figure 77: Convolutional Neural Network (CNN) models 3 test correlation

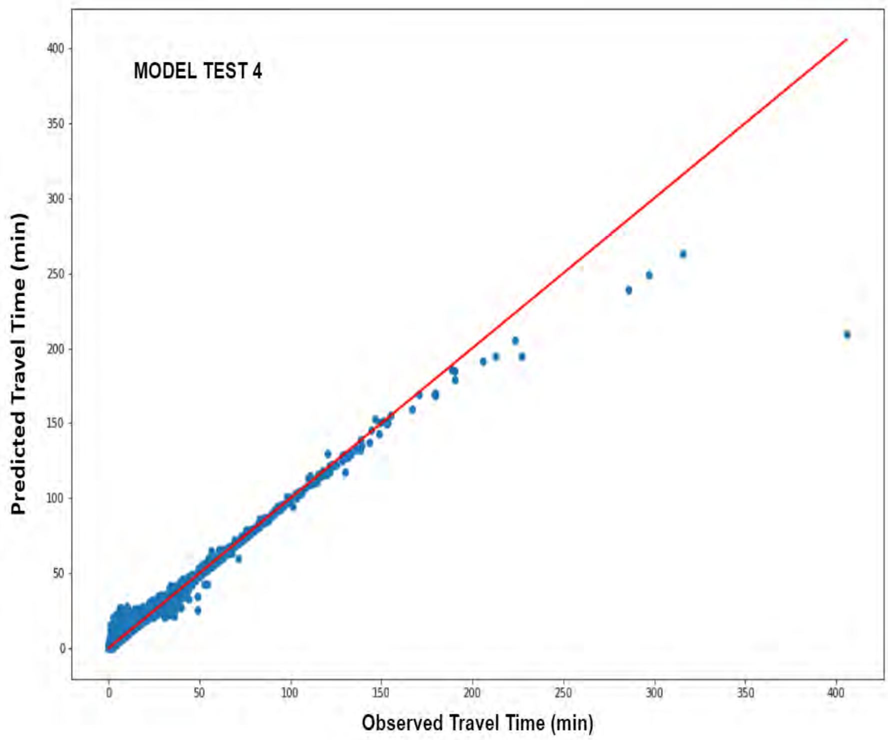

Figure 78: Convolutional Neural Network (CNN) model 4 test correlation

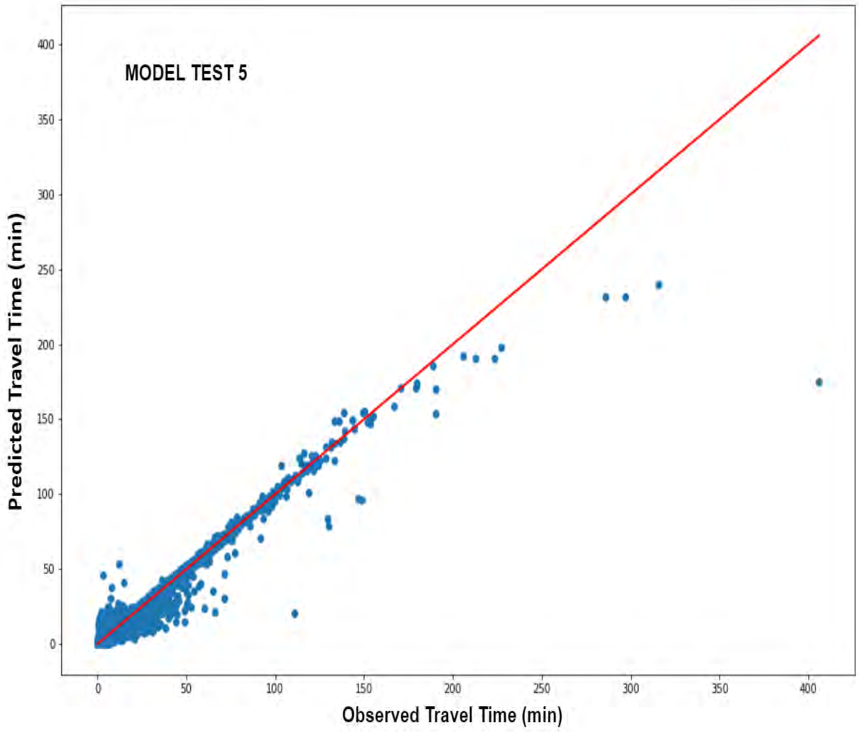

Figure 79: Convolutional Neural Network (CNN) model 5 test correlation

The plots of the correlation between the models used in predicting the travel time and the real travel time are as shown in Figures 75, 76, 77, 78 and 79 respectively. The correlation plots record huge differences among models 1-3, and models 4 and 5 tend to have more correlation plots align with the red line than the former. Also, the plots of error difference between predicted and actual travel time, show model 1-3 to have their errors spread out compared to model 4 and 5, which virtually have all their errors close or on +/-0 minutes.

Furthermore, the results of the travel time prediction for each model are plotted in Figures 80, 81, 82, 83 and 84. These prediction plots are for the first seven days of the test data, and the predictions are for every hour per day. Model 4 and 5 whose blue prediction line is very close to the red real travel time line, have very high accuracy in prediction. The outcomes of the models 1-3 show distant red and blue lines predicted per hour, which denote over-prediction of the travel time. In summary, again, model 4 shows the highest accuracy for travel time prediction using the Convolutional Neural Network (CNN).

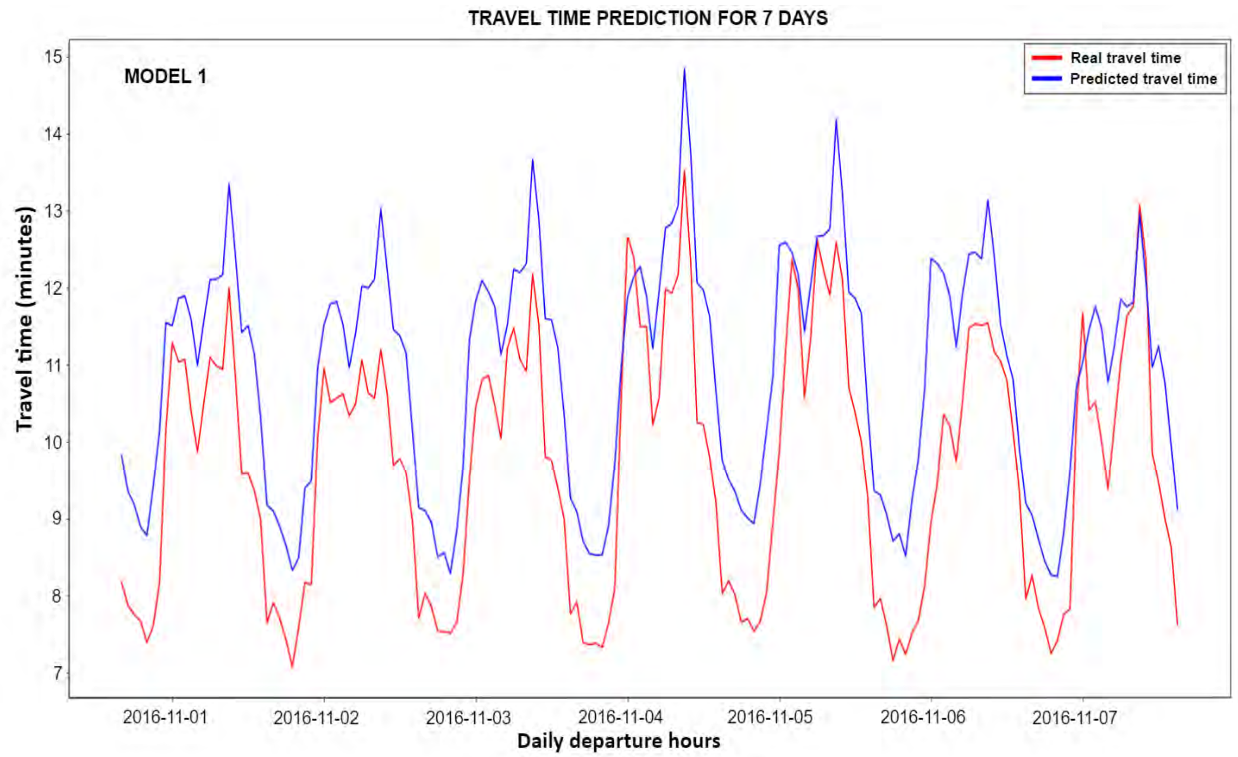

Figure 80: Convolutional Neural Network (CNN) model 1 travel time prediction per hour

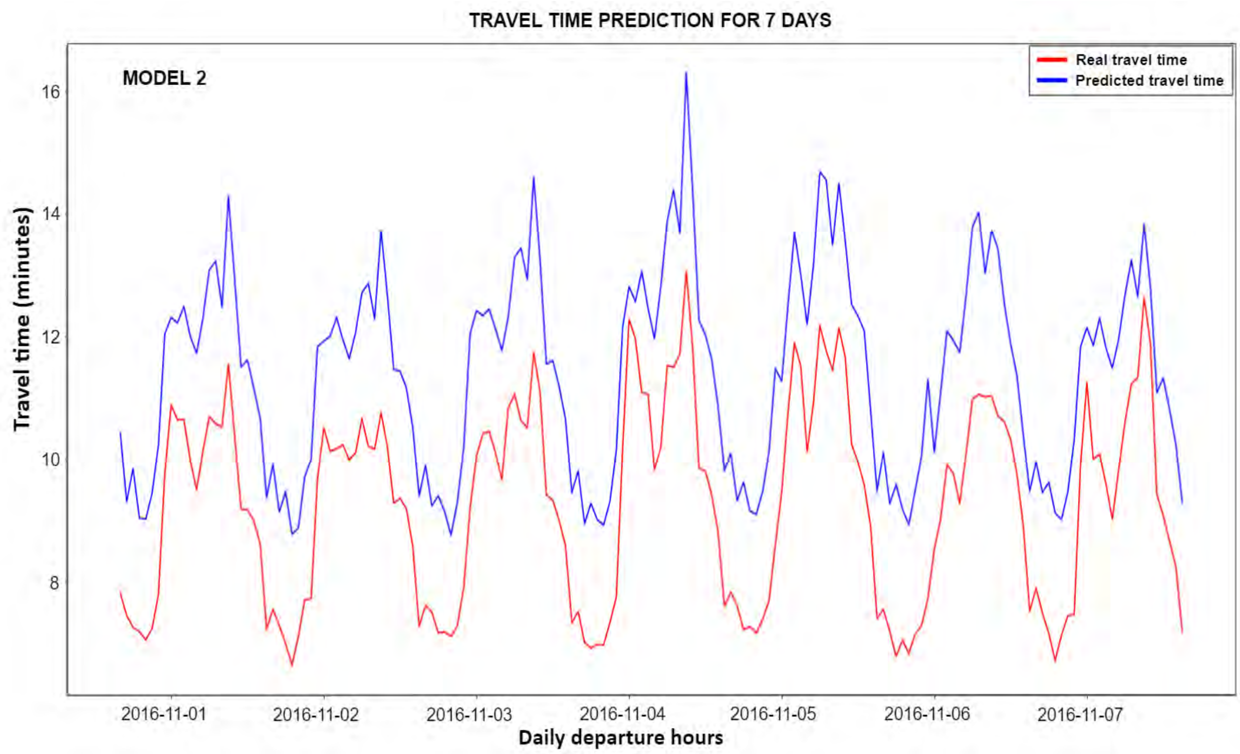

Figure 81: Convolutional Neural Network (CNN) model 2 travel time prediction per hour

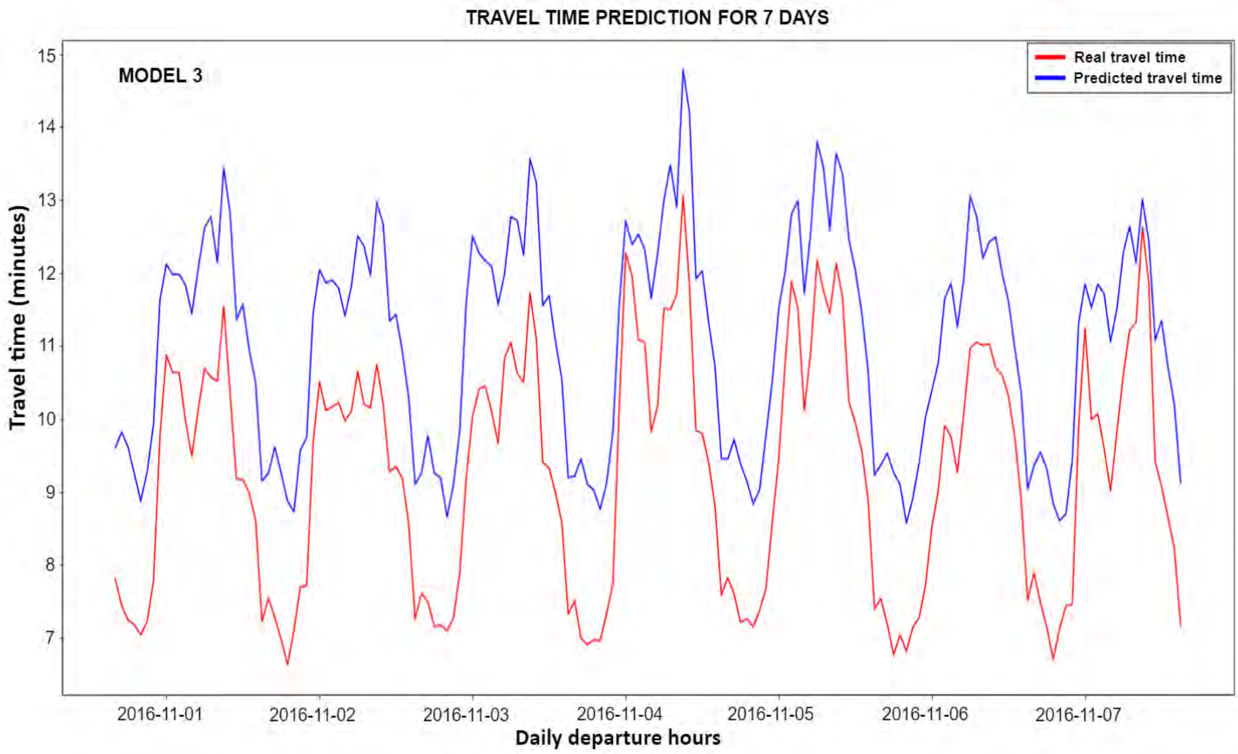

Figure 82: Convolutional Neural Network (CNN) model 3 travel time prediction per hour

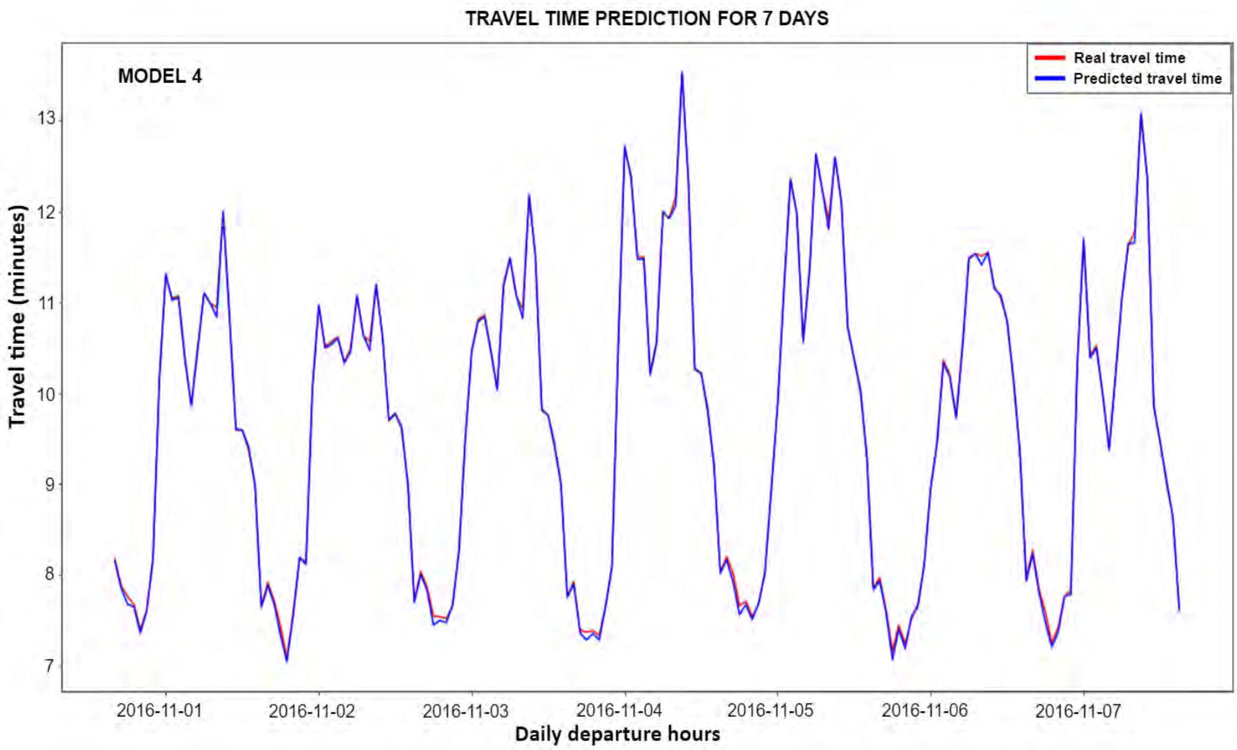

Figure 83: Convolutional Neural Network (CNN) model 4 travel time prediction per hour

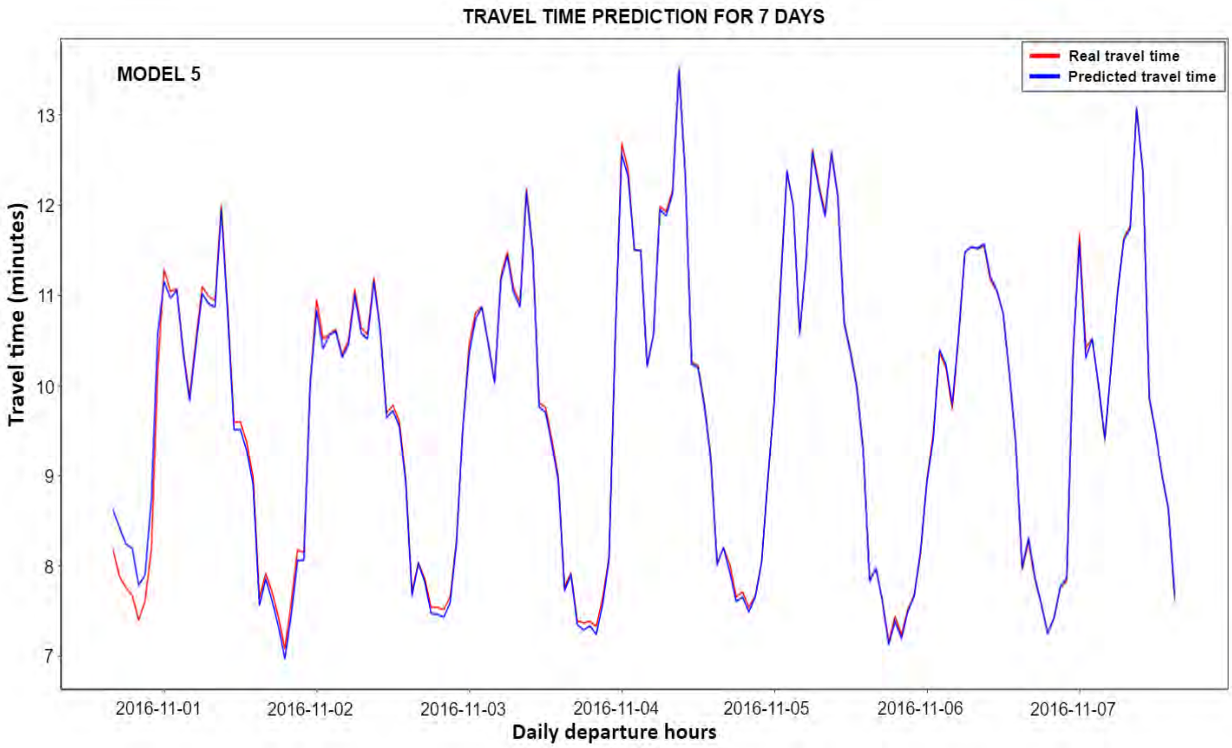

Figure 84: Convolutional Neural Network (CNN) model 5 travel time prediction per hour

4.2.3 Long Short-Term Memory (LSTM) model results

Model training and validation

Table 14: Long Short-Term Memory (LSTM) Models Trained for Travel Time Prediction

LSTM Models	Epochs/ Iterations	Optimizer	Loss Function	Activation Function	LSTM Layers	Neurons/ Units	MAE (min)	RMSE (min)	Coefficient of determination (R^2)	Train/ Validation (%)
Model 1	50	Adam	MAE	ReLU	4	9	5.059	8.065	0.625	70/30
Model 2	50	Adam	MAE	ReLU	4	49	4.641	7.631	0.664	70/30
Model 3	50	Adam	MAE	ReLU	4	63	4.686	7.710	0.657	70/30
Model 4	50	Adam	MAE	ReLU	4	64	0.394	0.978	0.994	70/30
Model 5	50	Adam	MAE	ReLU	4	10	0.234	1.008	0.994	70/30

The Table 14 shows 50 iterations of 5 Long Short Term Memory (LSTM) Models with 4 LSTM layers, and starting with nine neurons per layer, Adam optimizer, ReLU activation function and Mean Absolute Error (MAE) as the loss function. The first model has MAE of 5.059 minutes, RMSE of 8.065 minutes, and an R-squared score of 0.625 without trip speed as input. In subsequent models, input features like distance range, departure minute, and departure period of the day are added, and improvements are seen up to the third model.

Among the first three models, only model 2 has high-performance accuracy with R-squared, MAE and RMSE values of 0.664, 4.641 minutes and 7.631 minutes. Models 4 and 5 include travel speed as inputs, and sharp increments show in their performance. Out of all the models, the 4th model recorded a similar value of R-squared score of 0.994 with model 5 (the highest so far), the RMSE for model 4 is 0.978 minutes, which is lower compared to RMSE of model 5 with 1.008 minutes. The MAE of model 4 has 0.394 minutes, larger than that of model 5, which has 0.234 minutes.

In essence, this makes the models 4 and 5 the ones with the highest accuracy in predicting travel time for LSTM. However, the evaluation metrics' visualization plots shed more light into which of these two models is more suitable. Travel or trip speed remains the most powerful indicator for travel time prediction for on-demand trips.

The plots of MAE loss functions for both training data and validation data in Figures 85, 86, 87, 88 and 89 reveal the accuracy of the model during the training and validation through checking the curves' stability, reduction, and their convergence. As in the plots, the training loss curves have convergence and are stable, but the validation loss curves are unstable. The instability of the validation loss curves can be corrected by applying early-stop-loss. Mean Absolute Error (MAE) losses for models 1-3 tend to reduce at every epoch until they maintained flat curves.

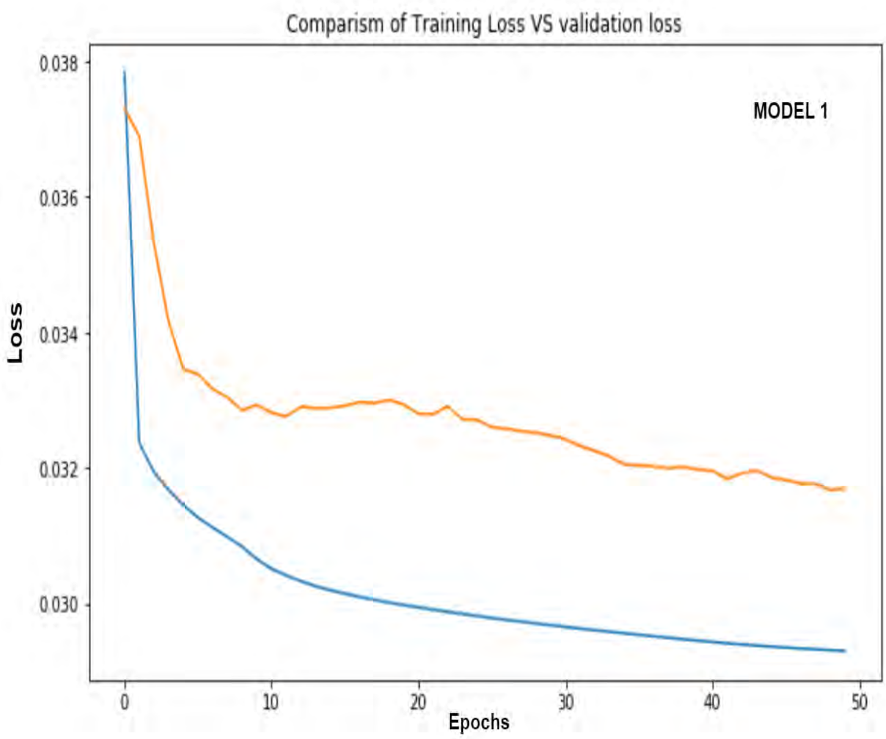

Figure 85: Long Short-Term Memory (LSTM) model 1 training and validation loss curves

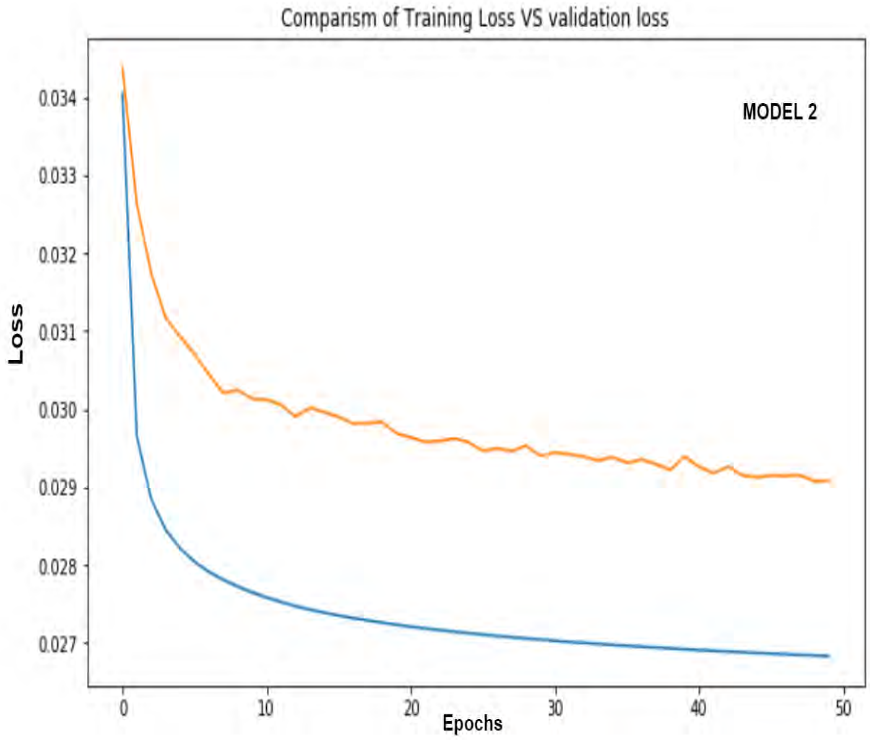

Figure 86: Long Short-Term Memory (LSTM) model 2 training and validation loss curves

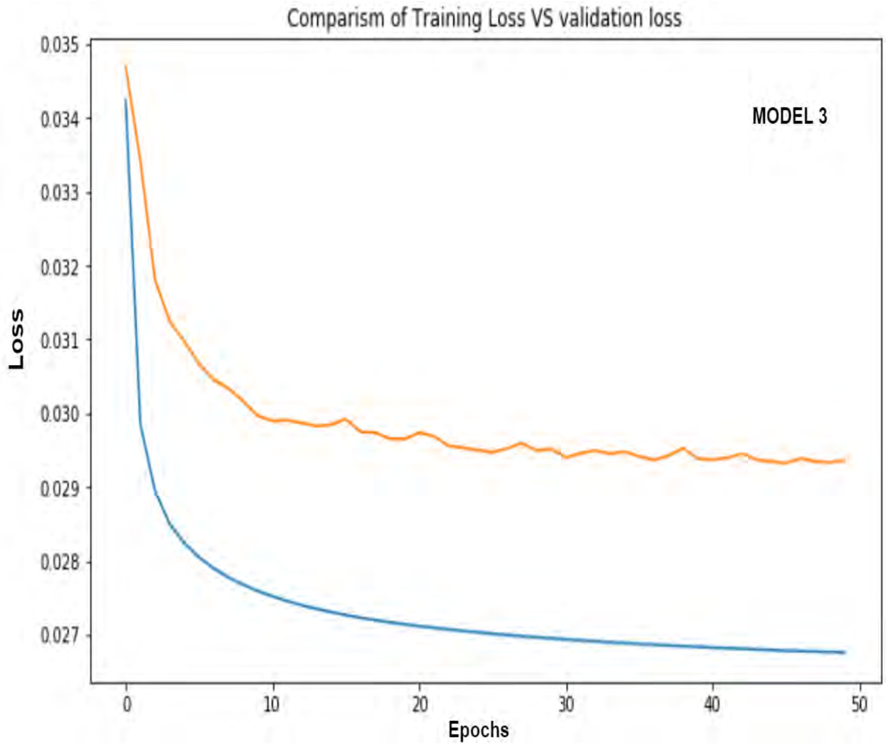

Figure 87: Long Short-Term Memory (LSTM) model 3 training and validation loss curves

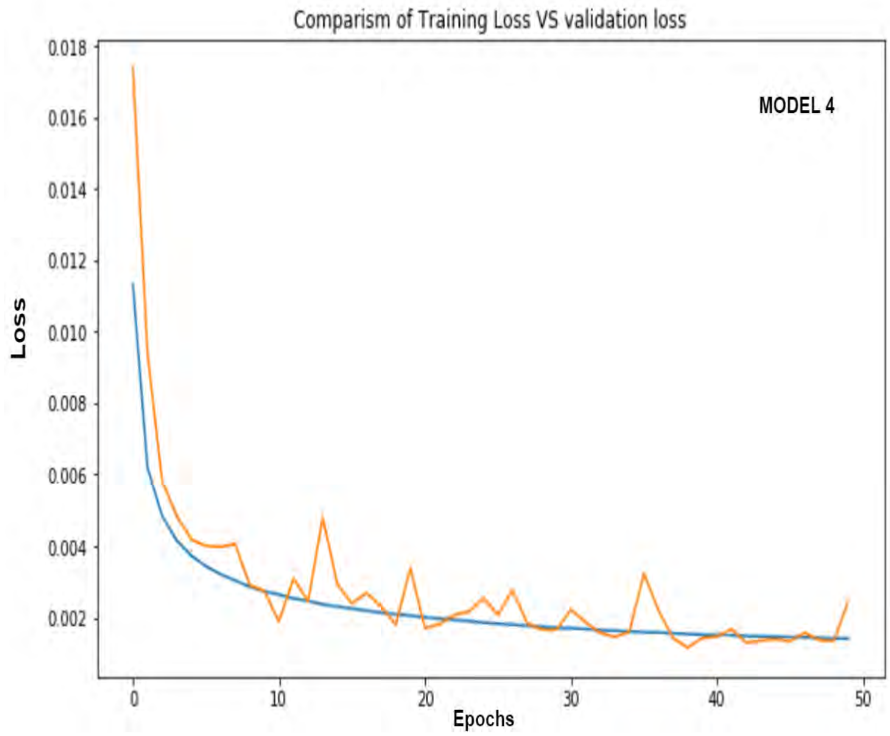

Figure 88: Long Short-Term Memory (LSTM) model 4 training and validation loss curves

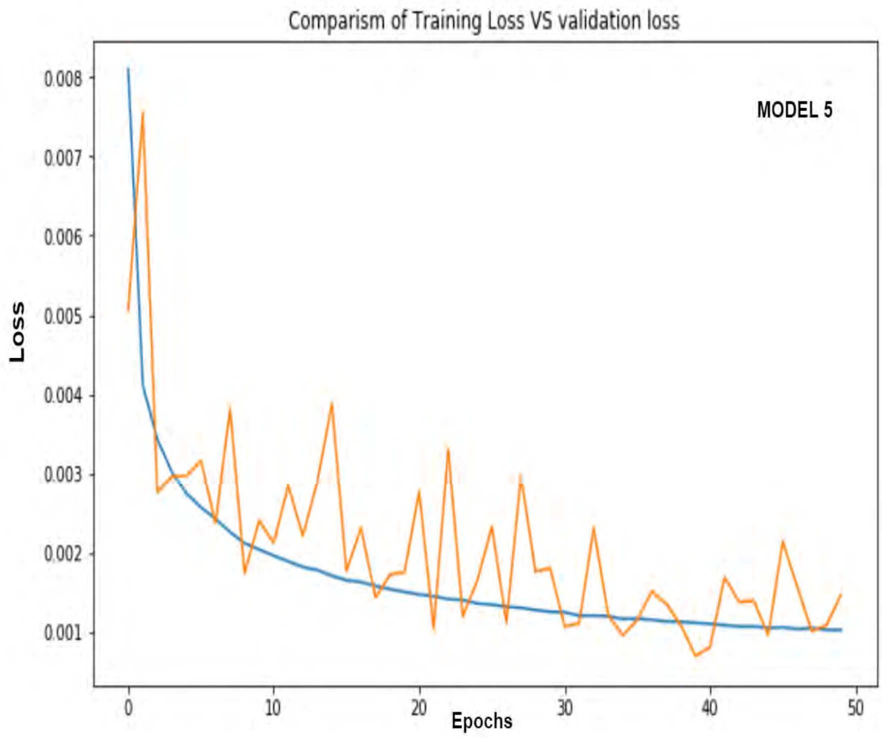

Figure 89: Long Short-Term Memory (LSTM) model 5 training and validation loss curves

The plots of the linear correlation between the validated data set and the trained data set are shown in Figures 90, 91, 92, 93 and 94. The models 1-3 correlation lines maintain angles of 45 degrees through the scattered plots, and very high correlations in models 4 and 5, between predicted travel time and actual travel time. The scattered plots for models 4-5 tend to have a high linear correlation between prediction and actual values. Also, the error differences between predictions using the validation data and the actual training data are huge, and they spread between +50 minutes and -50 minutes for models 1-3 compared to models 4 and 5, which range from -10 minutes to +10 minutes with the majority of the errors close to zero. The error plot of model 5 has almost all predictions close to zero or on zero minutes, and it indicates very high accuracy of the prediction of travel time with the model using Long Short-Term Memory (LSTM) technique.

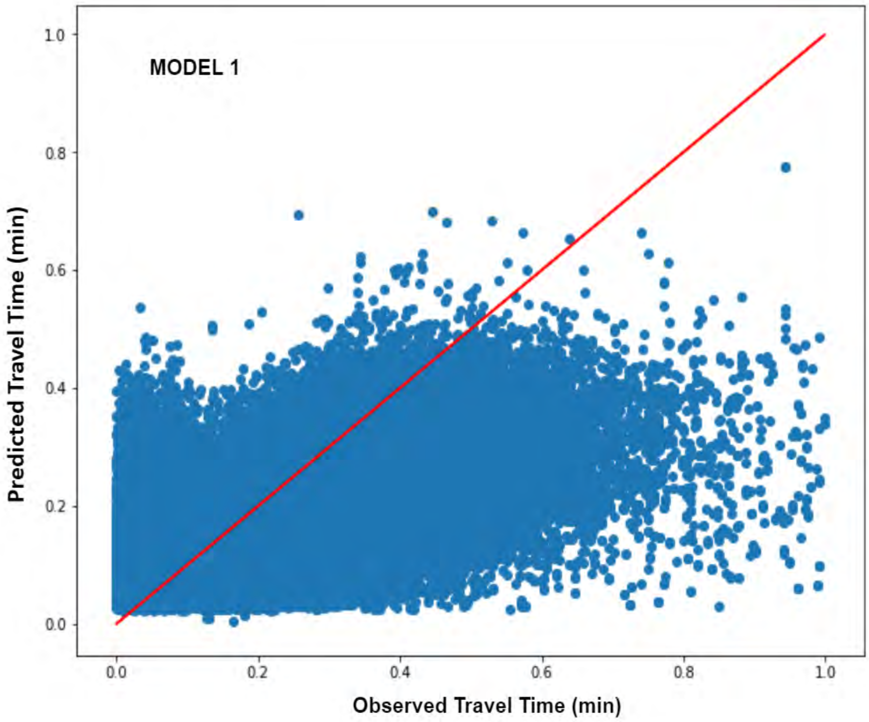

Figure 90: Long Short-Term Memory (LSTM) model 1 training and validation correlation

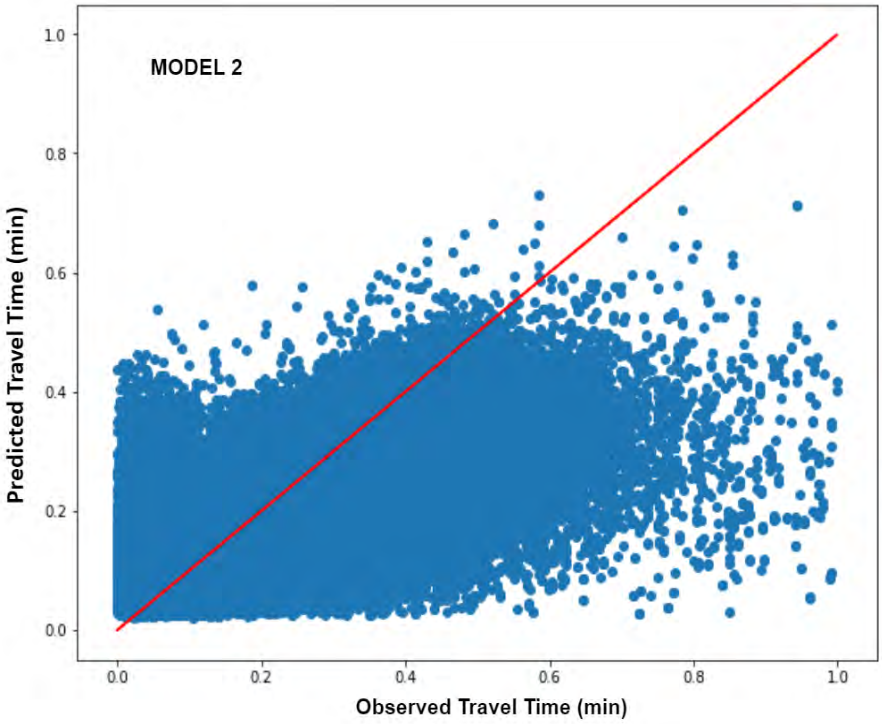

Figure 91: Long Short-Term Memory (LSTM) model 2 training and validation correlation

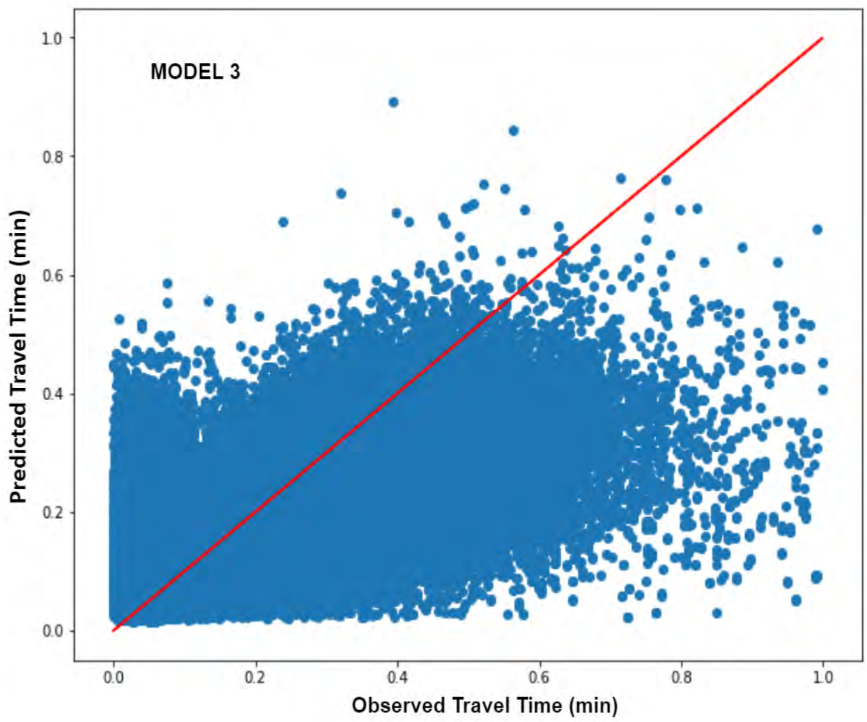

Figure 92: Long Short-Term Memory (LSTM) model 3 training and validation correlation

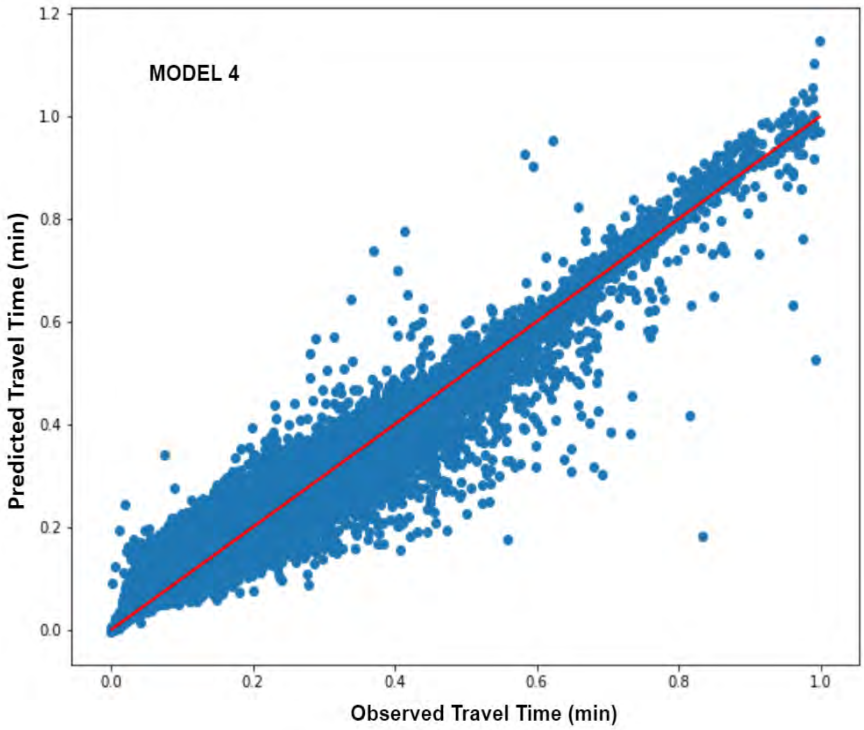

Figure 93: Long Short-Term Memory (LSTM) model 4 training and validation correlation

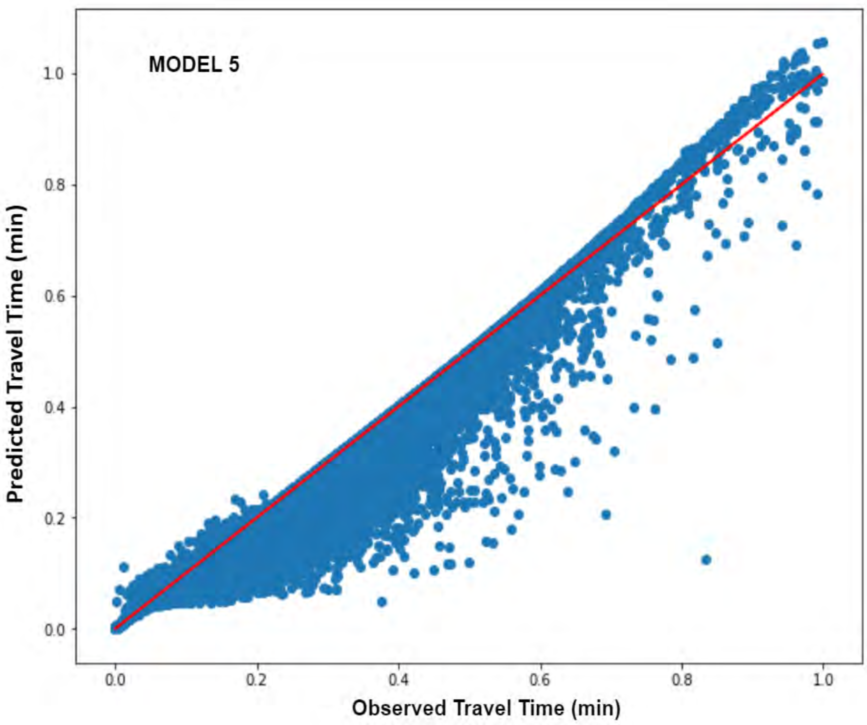

Figure 94: Long Short-Term Memory (LSTM) model 5 training and validation correlation

Model Test results

Table 15: Long Short-Term Memory (LSTM) models test results for travel Time prediction

LSTM Models	MAE (min)	RMSE (min)	Coefficient of determination (R^2)
Model 1	3.555	5.082	0.507
Model 2	3.557	4.964	0.545
Model 3	3.383	4.837	0.568
Model 4	0.201	0.740	0.990
Model 5	0.059	0.364	0.997

The trained models require testing with real-world travel time, and the test data were collected every 2-4 seconds from the coordinates of the commuter's trips from the origin up to the destination. The results are as shown in Table 15, and model 5 displays the highest performance denoting high prediction accuracy. This model has its R-squared score to be 0.997, the MAE value of 0.059 minutes, and the RMSE value of 0.364 minutes, both the lowest MAE and RMSE of all the models. Furthermore, the plots of the correlation between the predicted travel time and the real travel time are shown in Figures 95, 96, 97, 98 and 99, respectively. These plots record high linear correlations for models 4 and 5. In the same instance, the error differences between the predicted and actual for models 4 and 5 are close or on +/-0 minutes.

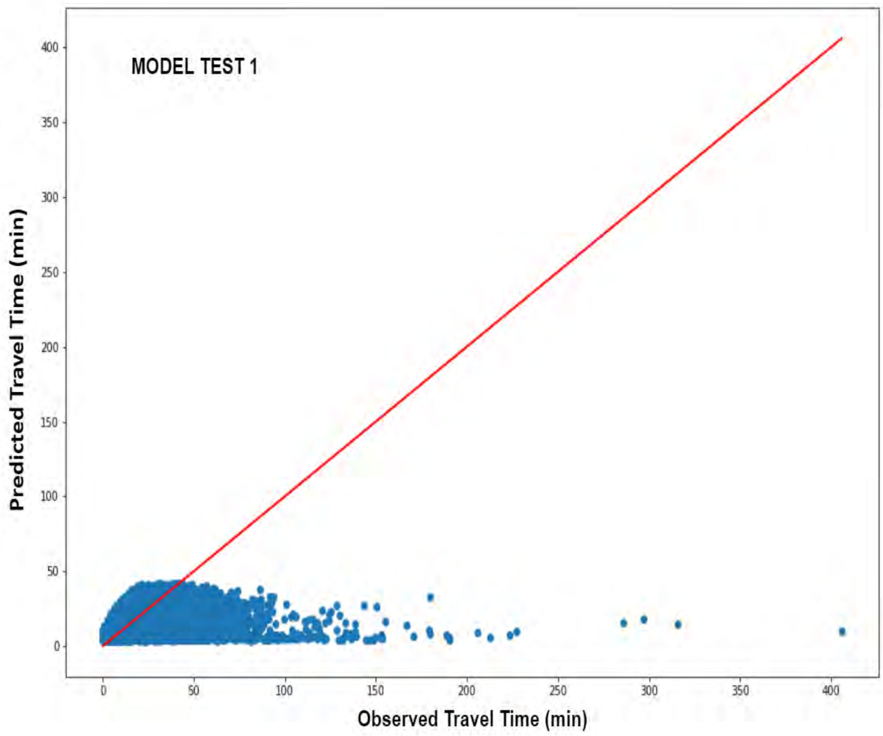

Figure 95: Long Short-Term Memory (LSTM) model 1 test correlation

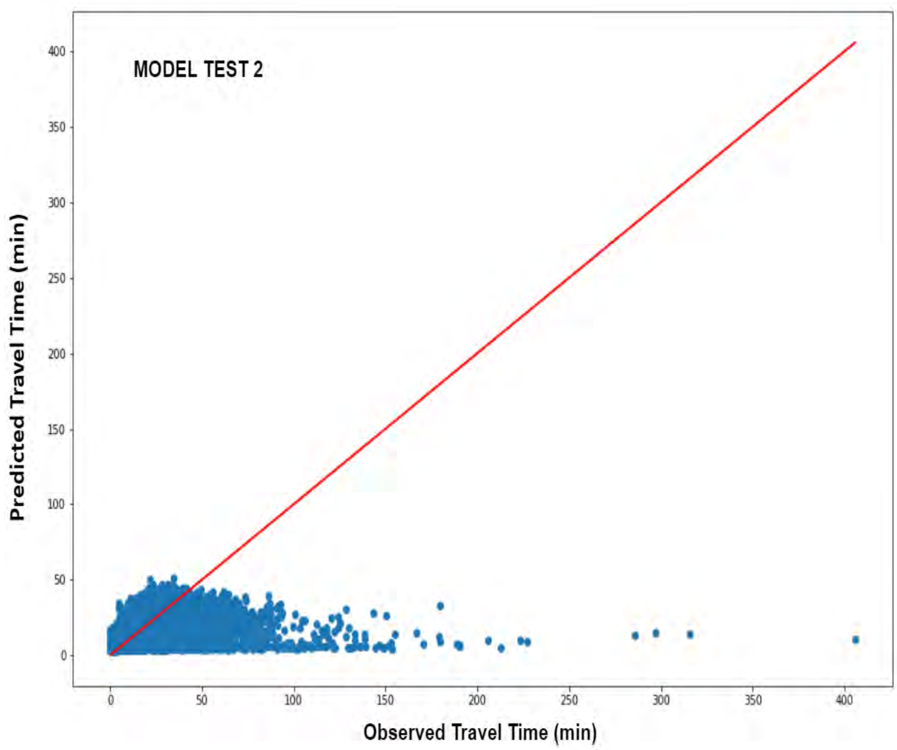

Figure 96: Long Short-Term Memory (LSTM) model 2 test correlation

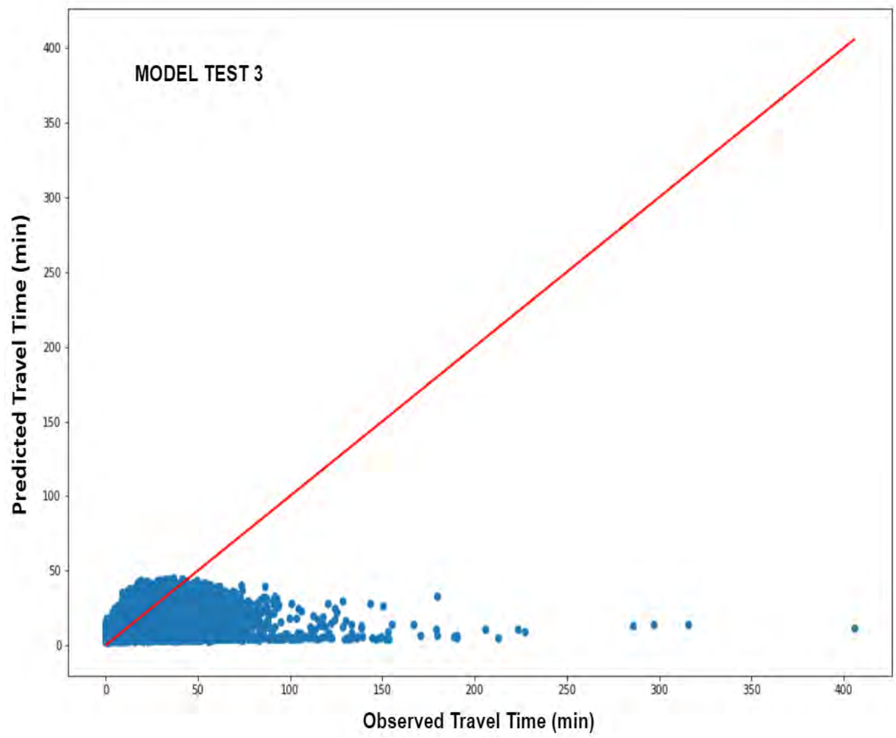

Figure 97: Long Short-Term Memory (LSTM) model 3 test correlation

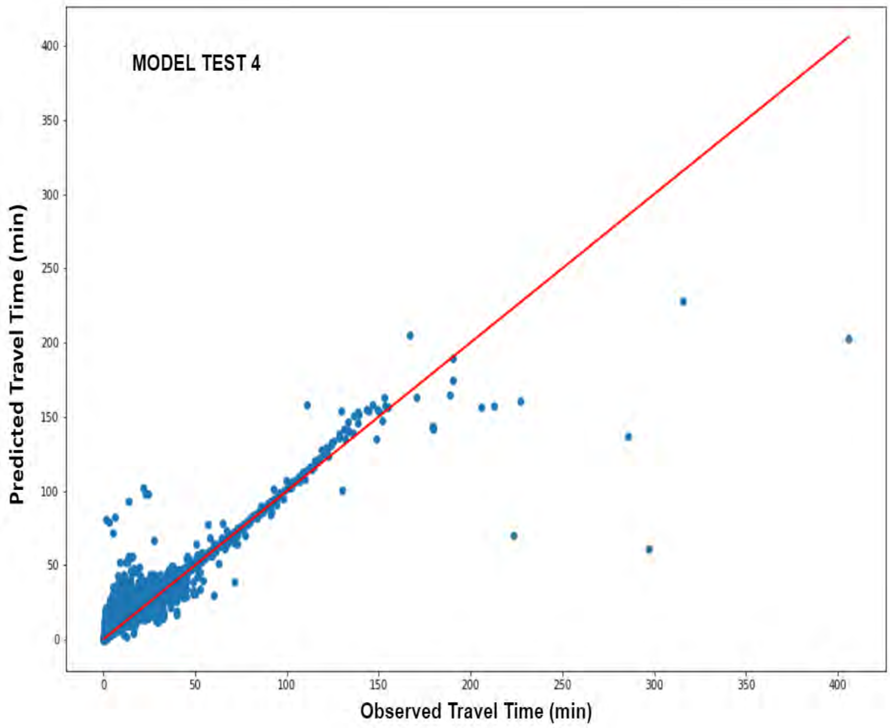

Figure 98: Long Short-Term Memory (LSTM) model 4 test correlation

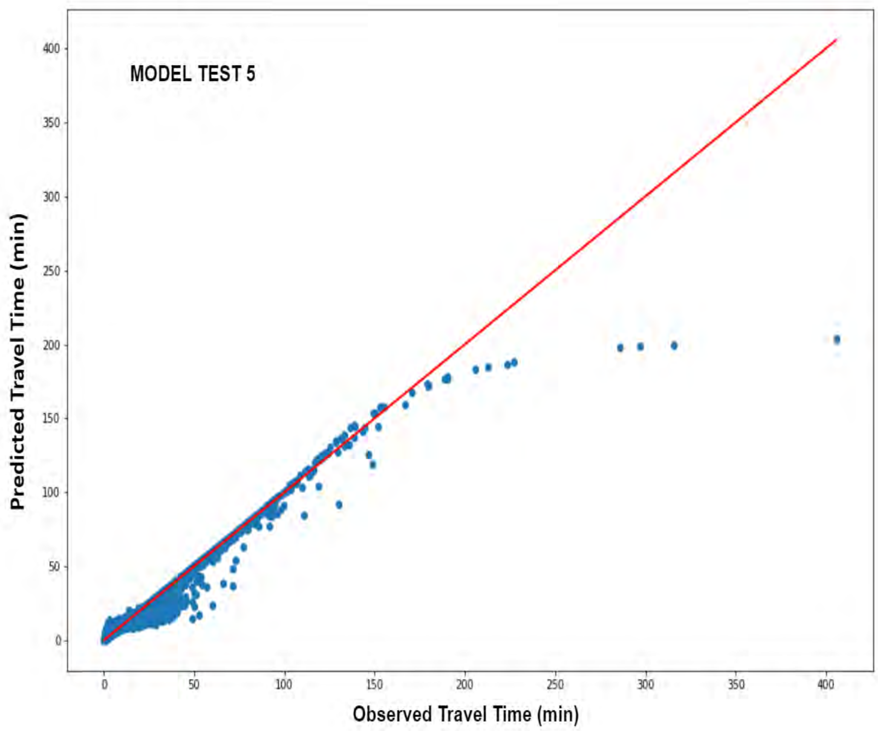

Figure 99: Long Short-Term Memory (LSTM) model 5 test correlation

Finally, the travel time prediction for each model are plotted in Figures 100, 101, 102, 103 and 104. These prediction plots are for the first seven days of the test data, while the predictions are for every hour per day. Models 4 and 5 have high prediction accuracy, and models 1-3 show distant red and blue lines predicted per hour, which means over-prediction of the travel time. In summary, models 4 and 5 have the highest performance for travel time prediction using the Long Short-Term Memory (LSTM). The prediction plots confirm previous result visualization and values like correlation, training and validation curves, and error difference descriptions. These mean that with models 4 and 5, mobility operators and commuters can forecast their upcoming trips demand-responsively.

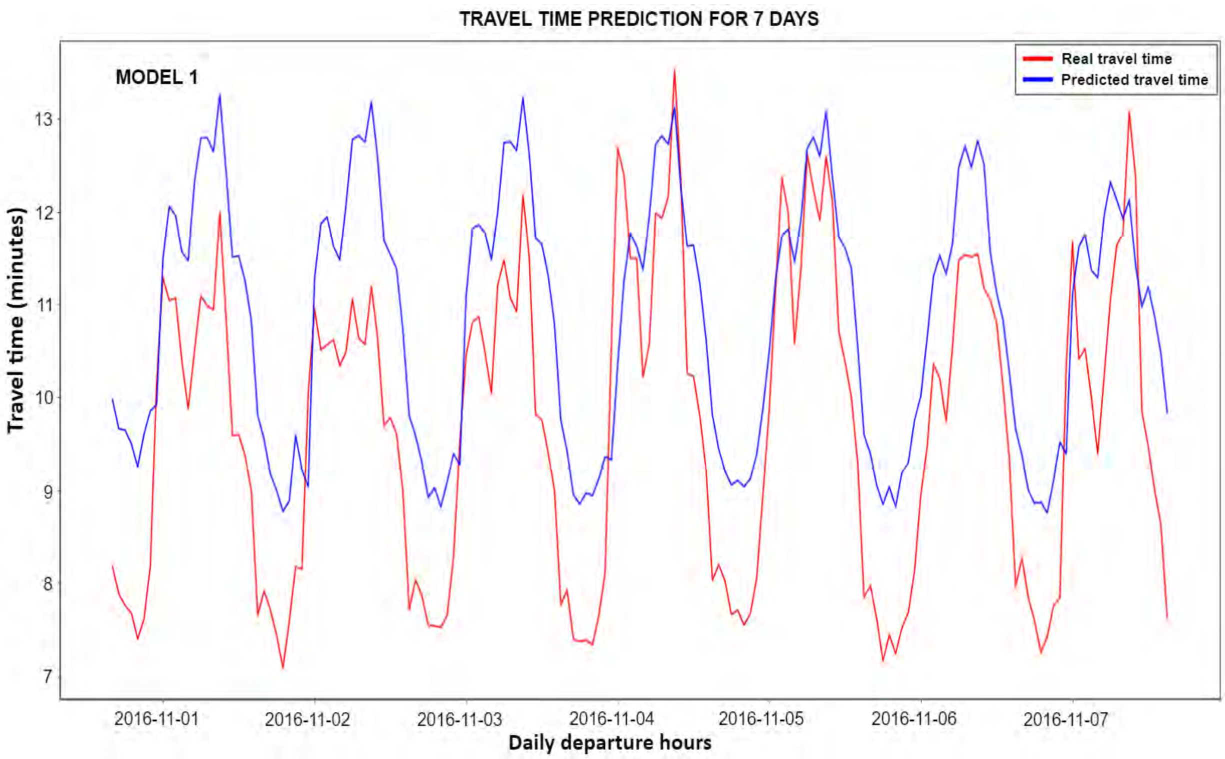

Figure 100: Long Short-Term Memory (LSTM) model 1 travel time prediction per hour

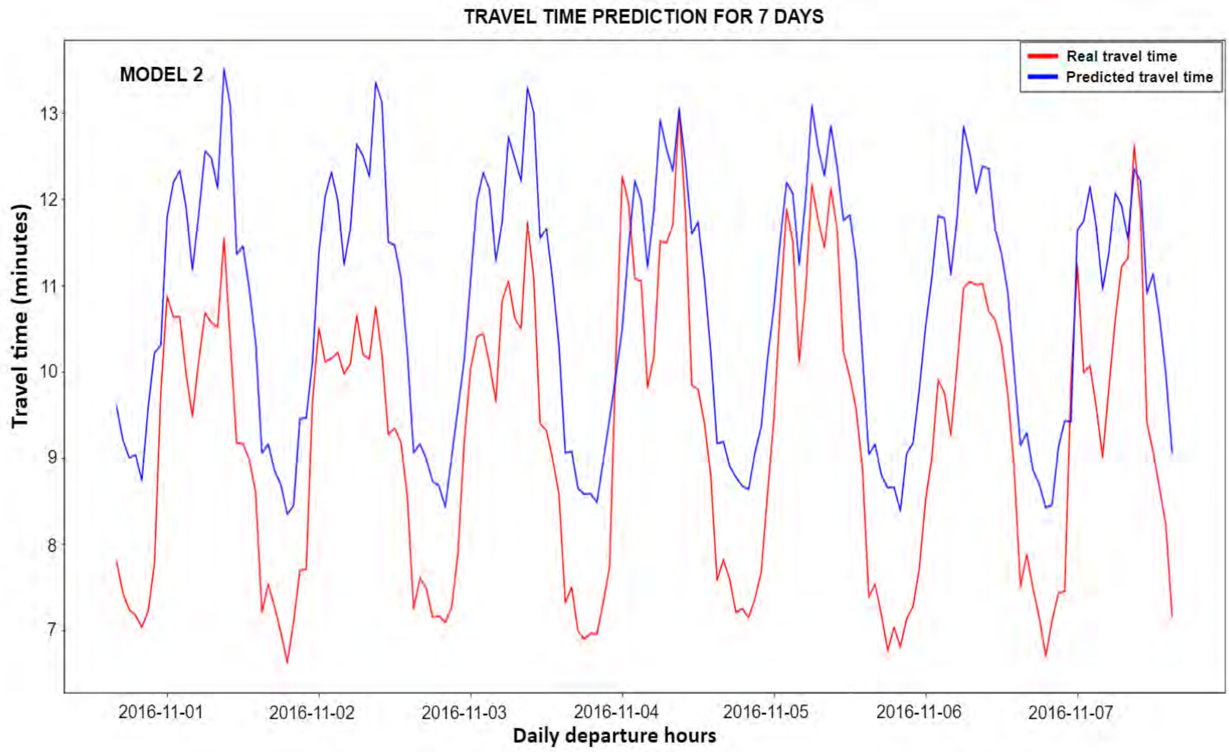

Figure 101: Long Short-Term Memory (LSTM) model 2 travel time prediction per hour

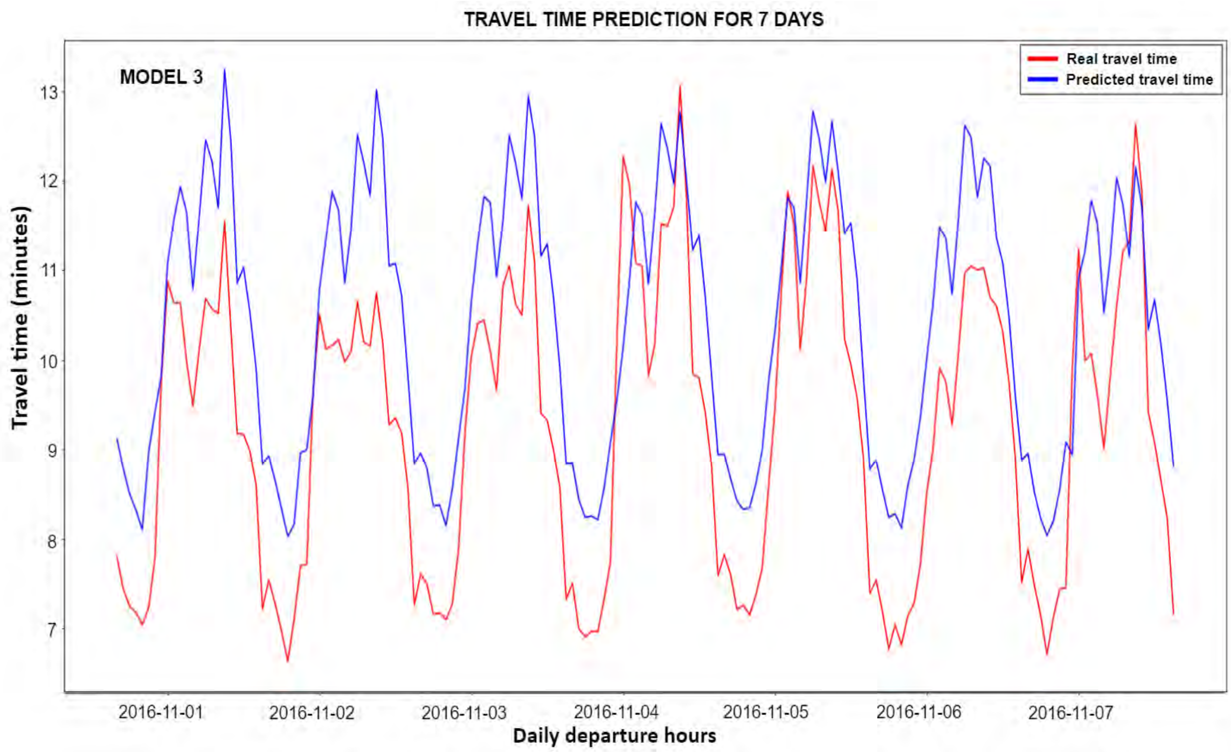

Figure 102: Long Short-Term Memory (LSTM) model 3 travel time prediction per hour

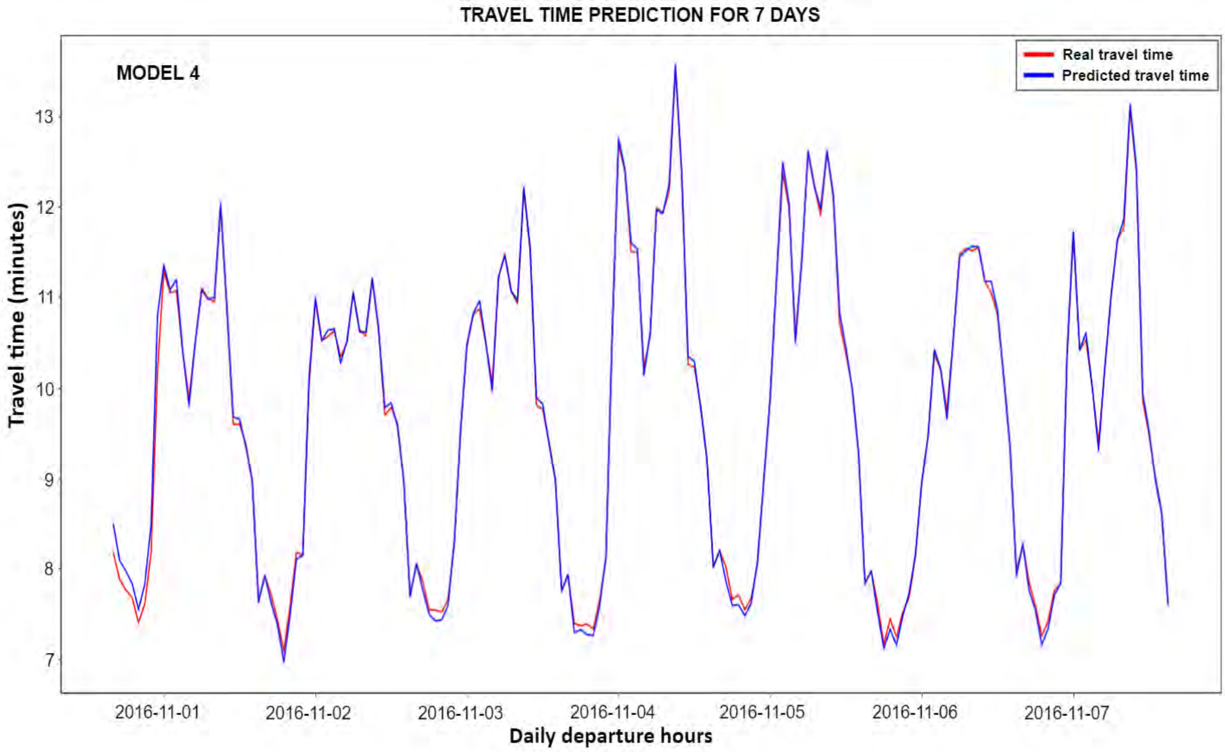

Figure 103: Long Short-Term Memory (LSTM) model 4 travel time prediction per hour

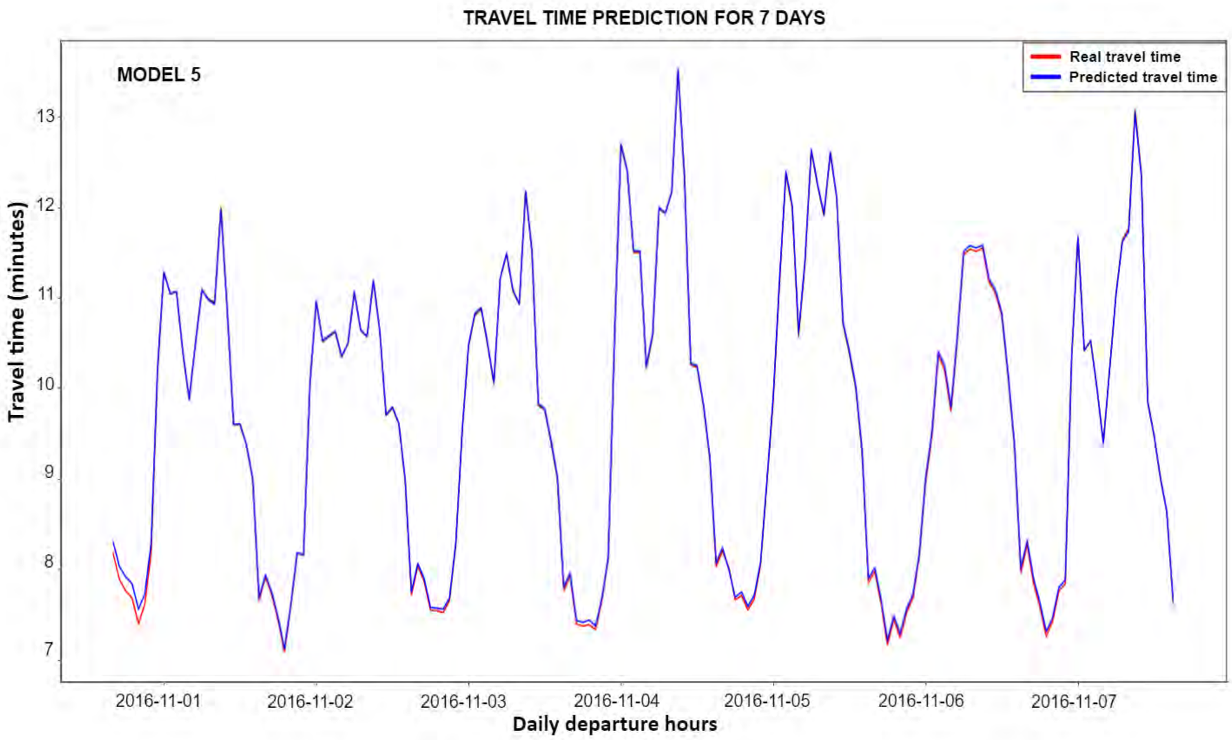

Figure 104: Long Short-Term Memory (LSTM) model 5 travel time prediction per hour

4.3 Comparison and evaluation of deep learning neural networks performance

In this research, three (3) deep learning modeling types are developed and compared for predicting the travel time of individual on-demand trips. The deep learning types are Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and a Recurrent Neural Network (RNN) technique called Long Short-Term Memory (LSTM). The five models developed for each type of deep learning modeling techniques were visualized, analyzed, and discussed in the previous section. However, this section compares one model from each deep learning technique that has the highest performance after using the three evaluation metrics as measurements and determinants. These evaluation metrics are R-squared or coefficient of determination, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Simultaneously, some other visualization plots similar to those of the previous section are also used for comparison and evaluation. These measurement approaches were also used in previous research works, as reviewed in this study's literature chapter.

Table 16: Comparison results of deep learning neural networks performance

Deep Learning Models	Epochs	Optimizer	Loss Function	Activation Function	Neurons/ Units	Model training			Model testing		
						MAE (min)	RMSE (min)	R^2	MAE (min)	RMSE (min)	R^2
ANN Model 4	50	Adam	MAE	ReLU	64	0.212	0.769	0.997	0.093	0.373	0.997
CNN Model 4	50	Adam	MAE	ReLU	64	0.288	0.685	0.997	0.197	0.403	0.997
LSTM Model 4	50	Adam	MAE	ReLU	64	0.394	0.978	0.994	0.201	0.740	0.990
LSTM Model 5	50	Adam	MAE	ReLU	10	0.234	1.008	0.994	0.059	0.364	0.997

Table 16 shows the parameters and results of the four high-performing deep learning models, which ran through 50 iterations using adam optimizer, MAE loss function, Rectified Linear Unit (ReLU) activation function and 10-64 neurons equivalent to the number of inputs or indicators. The Artificial Neural Network (ANN) trained model 4 validates with outcomes of 0.212 minutes (12.7 seconds) for Mean Absolute Error (MAE), 0.769 minutes (46.1 seconds) for Root Mean Squared Error (RMSE) and R-squared of 0.997. After testing these models with the test data and compared with the real travel time, the MAE reduces to 0.093 minutes (5.6 seconds), the RMSE also levels down to 0.373 minutes (22.4 seconds), and the R-squared remains 0.997. It means an improvement in the model with a prediction accurately close to the real travel time.

Similar results also in the Convolutional Neural Network (CNN) model with 0.997 of R-squared for train-validation and test. The MAE for testing also reduces to 0.197 minutes (11.8 seconds) from 0.288 minutes (17.3 seconds). Similarly, the RMSE for train-validation is higher than the test with values of 0.685 minutes (41.1 seconds) and 0.403 minutes (24.2 seconds) in a respective manner. However, the LSTM has two highly performing models, and the train-validation has similar R-squared values of 0.994 for both models 4 and 5. In contrast, the model tests have 0.990 and 0.997 R-squared values for 4 and 5, respectively. According to the validation and testing results, LSTM model 5 has high performance with R-squared increasing from 0.994 for train-validation to 0.997 for model testing. The MAE and RMSE also reduced to 0.059 minutes (3.5 seconds) and 0.364 minutes (21.8 seconds), respectively, for model 5.

The reduction in values discovered in MAE and RMSE shows the models' high performance in a real-world application for travel time, and having the R-squared of train and test equal or increased also means the high performance of the model. LSTM model 4 is lower in performance compared to model 5, with increased MAE and RMSE values for both model validations and testing. The reasons for selecting model 4 is as a result of its validation loss curve, having the best convergence and stability, which is more than five epochs before it starts to spike (unstable). This observation is different from other model loss curves, which start spiking or unstable from the beginning of training. Another reason for selecting LSTM model 4 is as a result of having the highest number of relevant model inputs or indicators of 64. For other models, their instability starts from the second epoch, and the stability and convergence of the loss curves denote well-trained models with high accuracy for prediction.

Figures 105, 106, 107 and 108 all show the differences in the convergence and stability of the curves. Early stopping and training with more data are recommended in previous literature to resolve unstable curves. However, LSTM model 4 shows a high convergence and stability level between epoch 1-10 compared to the other three models. This indicates a well-trained LSTM model 4 with the validation loss sequential reduction, and with more data, an improvement will be recorded.

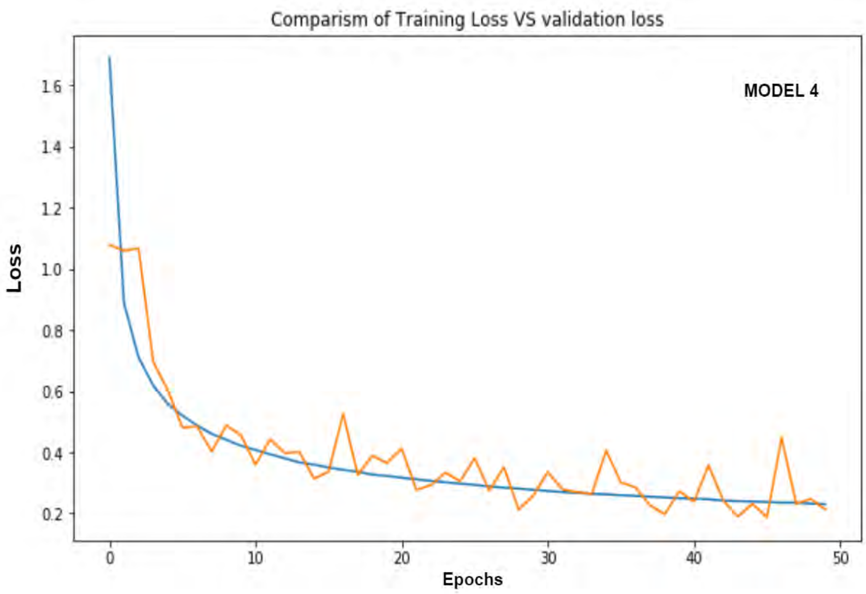

Figure 105: Artificial Neural Network (ANN) model 4 training and validation loss curves

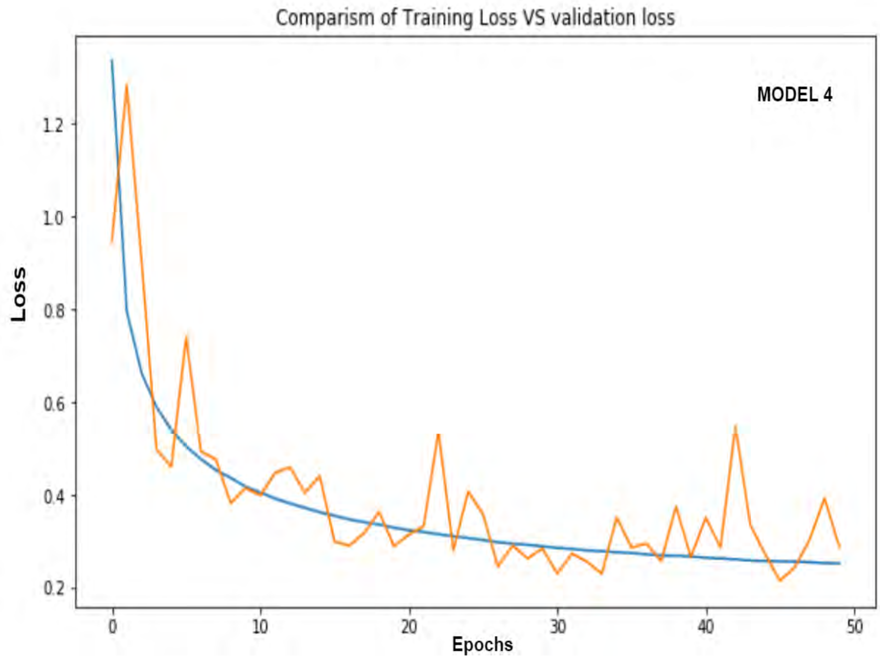

Figure 106: Convolutional Neural Network (ANN) model 4 training and validation loss curves

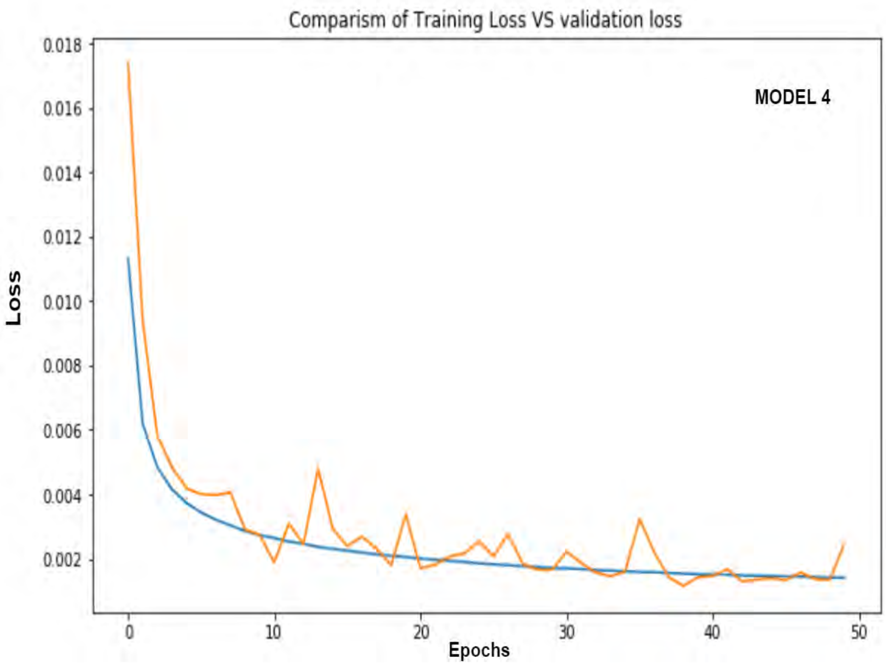

Figure 107: Long Short-Term Memory (LSTM) model 4 training and validation loss curves

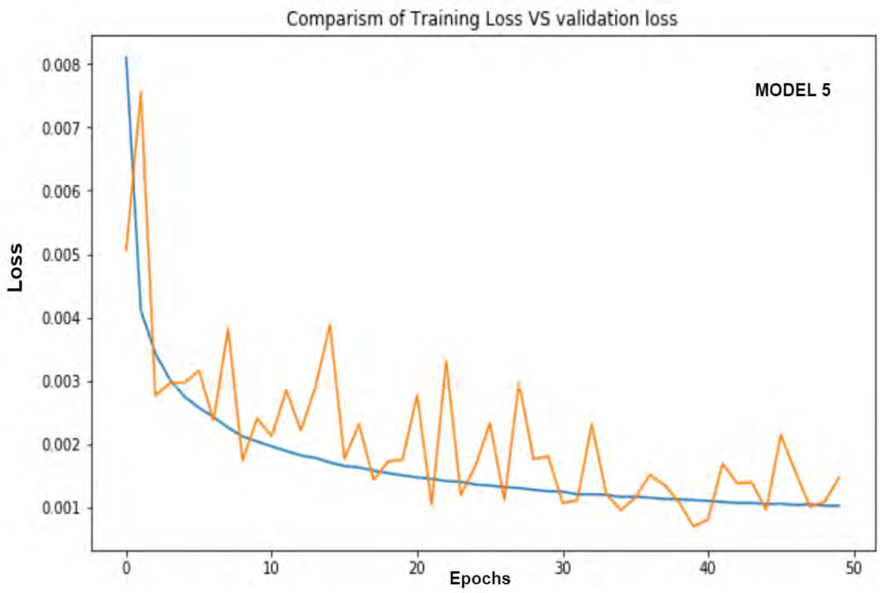

Figure 108: Long Short-Term Memory (LSTM) model 5 training and validation loss curves

The correlation plots of models for the predicted travel time and the real travel time are shown in Figures 109, 110, 111 and 112 respectively. These plots have a similarly high correlation in line with their coefficient of determination known as R-squared score of 0.997 except LSTM model 4 with 0.990. As earlier described during each model development, the error differences between predicted and actual travel time have almost all the error values close or on +/-0 minutes. These outcomes denote high prediction accuracy for the models developed.

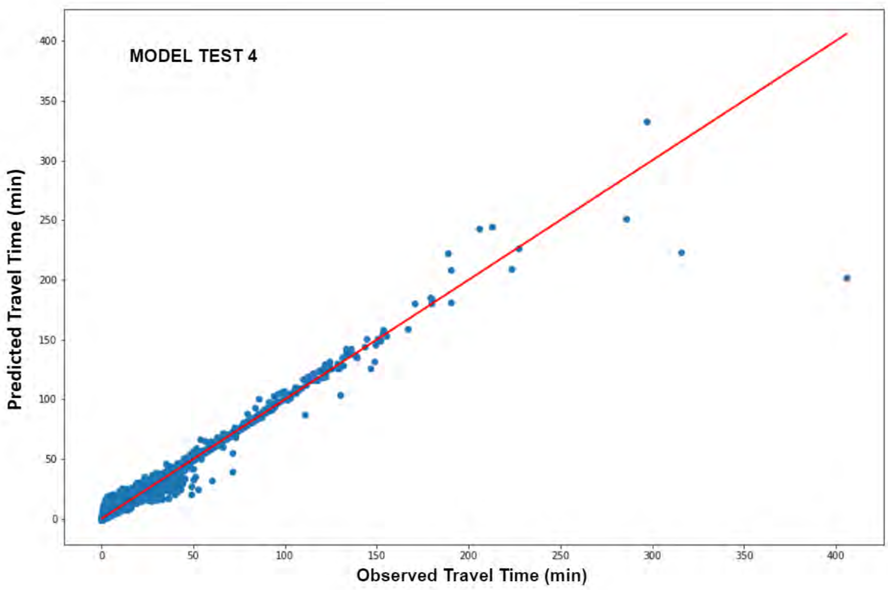

Figure 109: Artificial Neural Network (ANN) model 4 test correlation plot

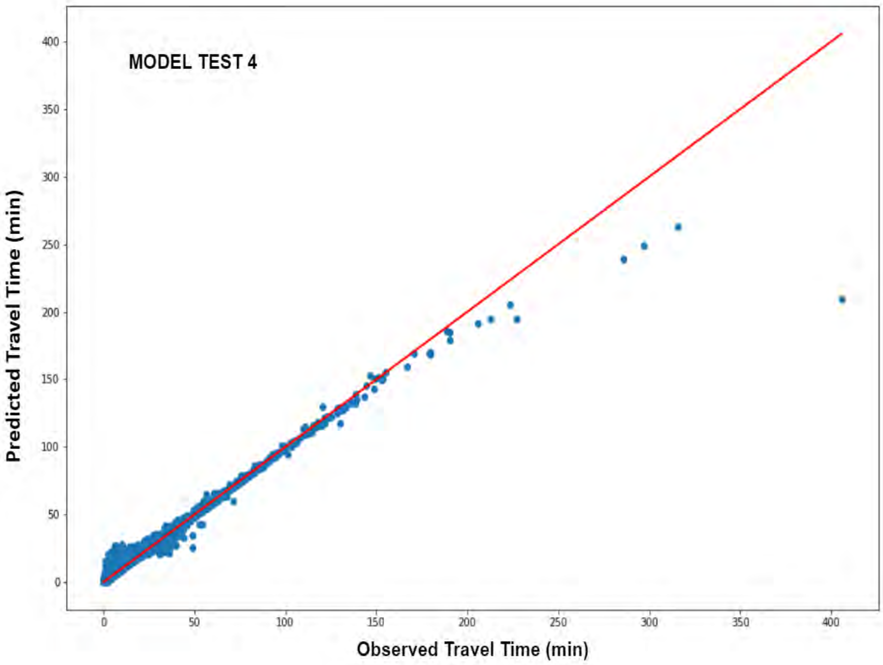

Figure 110: Convolutional Neural Network (ANN) model 4 test correlation plot

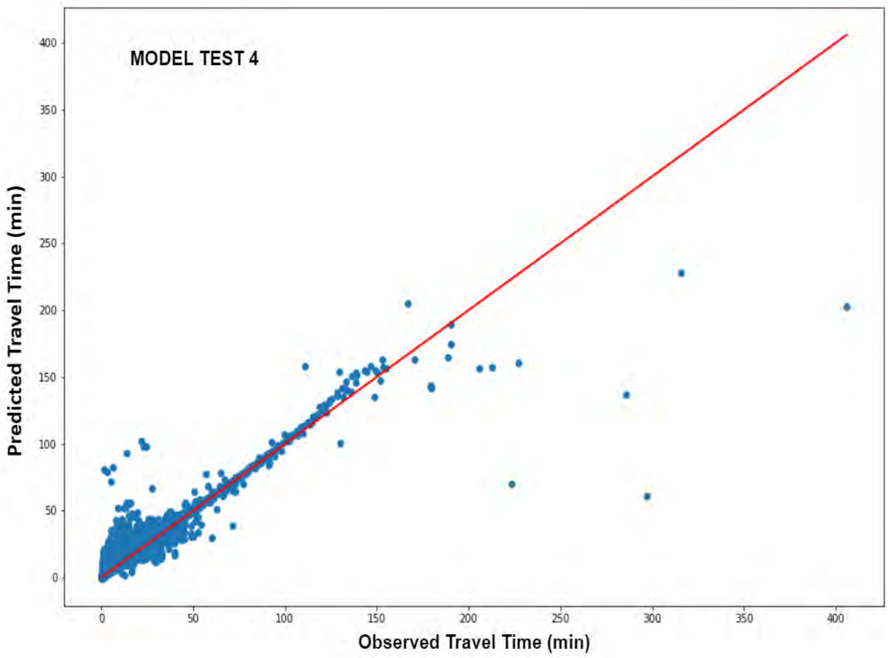

Figure 111: Long Short-Term Memory (LSTM) model 4 test correlation plot

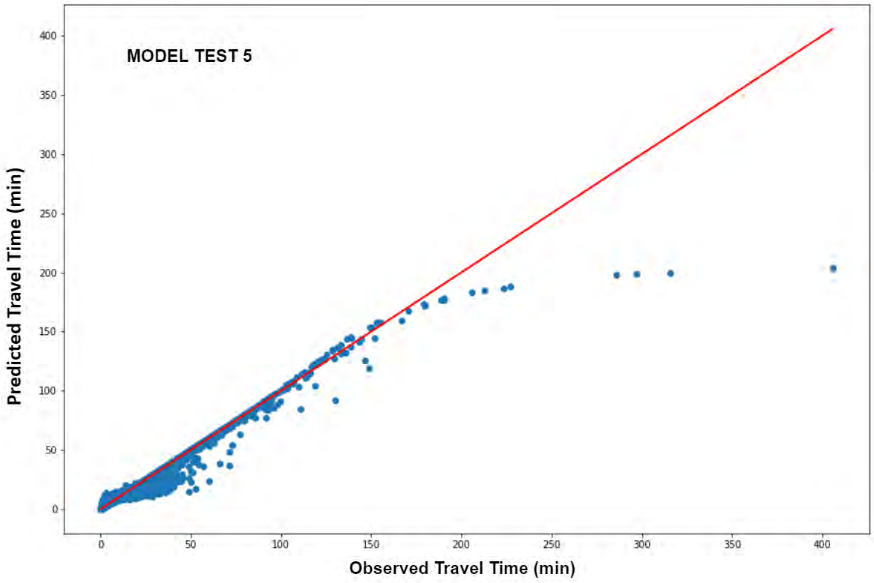

Figure 112: Long Short-Term Memory (LSTM) model 5 test correlation plot

The visualization results of the travel time prediction for each model are in Figures 113, 114, 115 and 116. These predictions are for the first seven days of the test data, and the predictions are for every hour per day. The predicted and real travel time lines are very close together, confirming very high accuracy in prediction. LSTM models further show from these plots to have the highest accuracy for travel time prediction. Therefore, mobility operators, transport authorities, and researchers can adopt these models to predict travel time, however, with more large data to train the models.

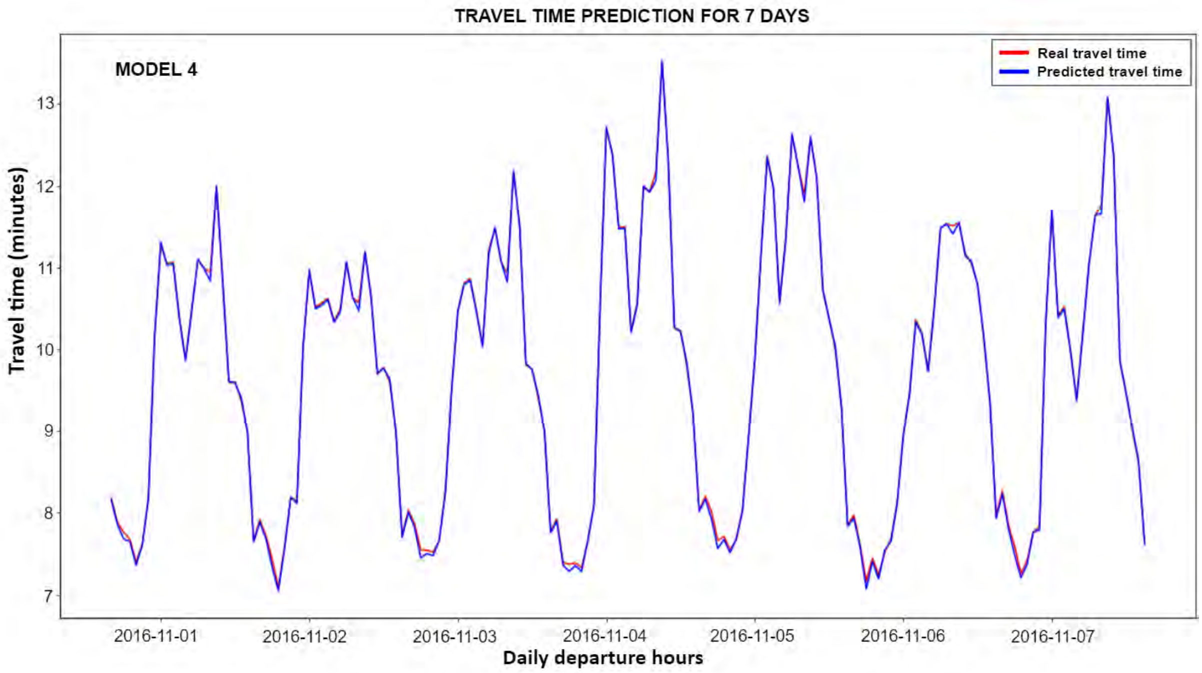

Figure 113: Artificial Neural Network (ANN) model 4 real travel time prediction per hour

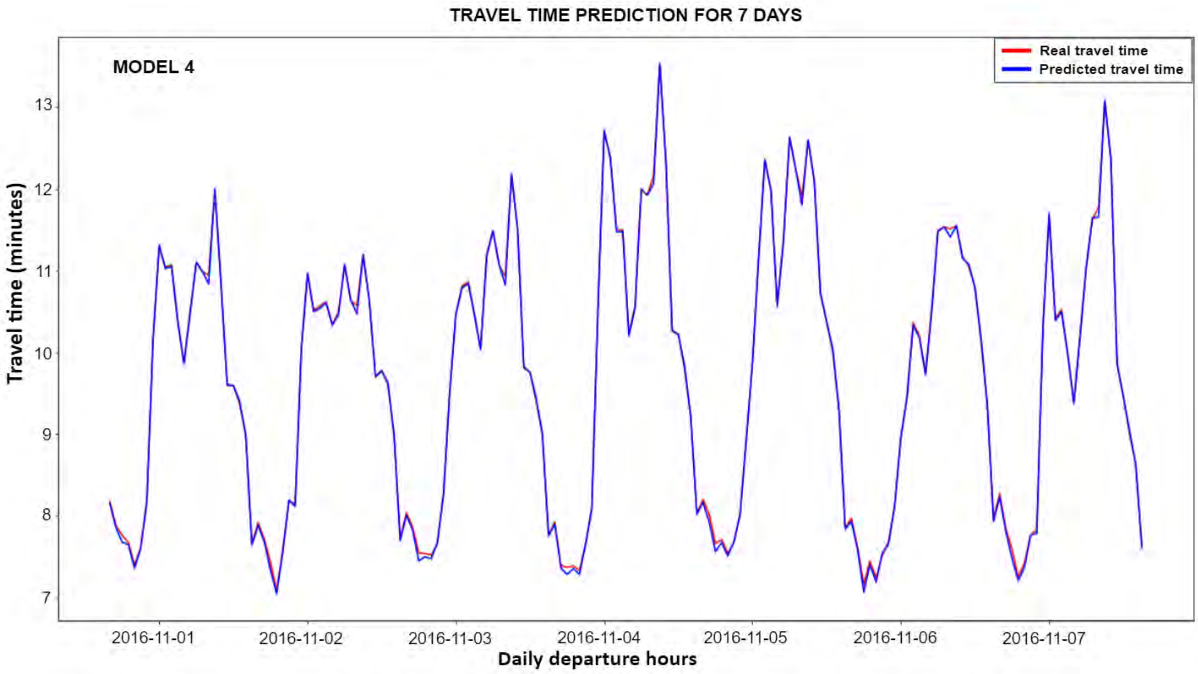

Figure 114: Convolutional Neural Network (CNN) model 4 real travel time prediction/hour

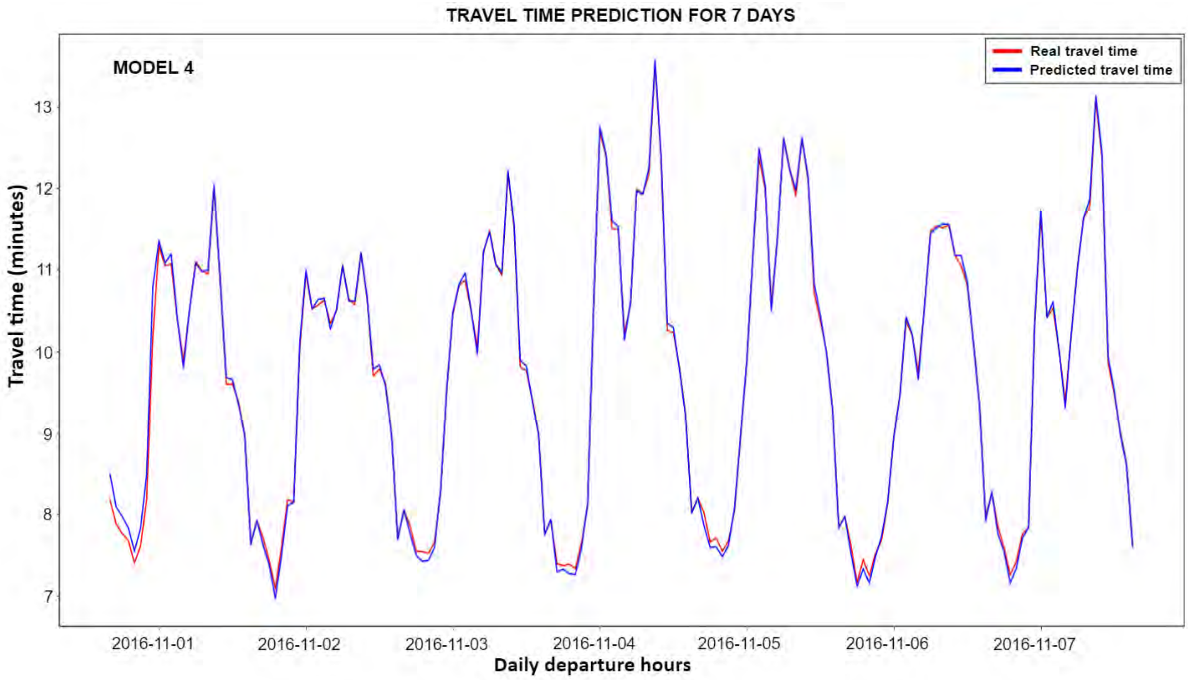

Figure 115: Long Short-Term Memory (LSTM) model 4 real travel time prediction per hour

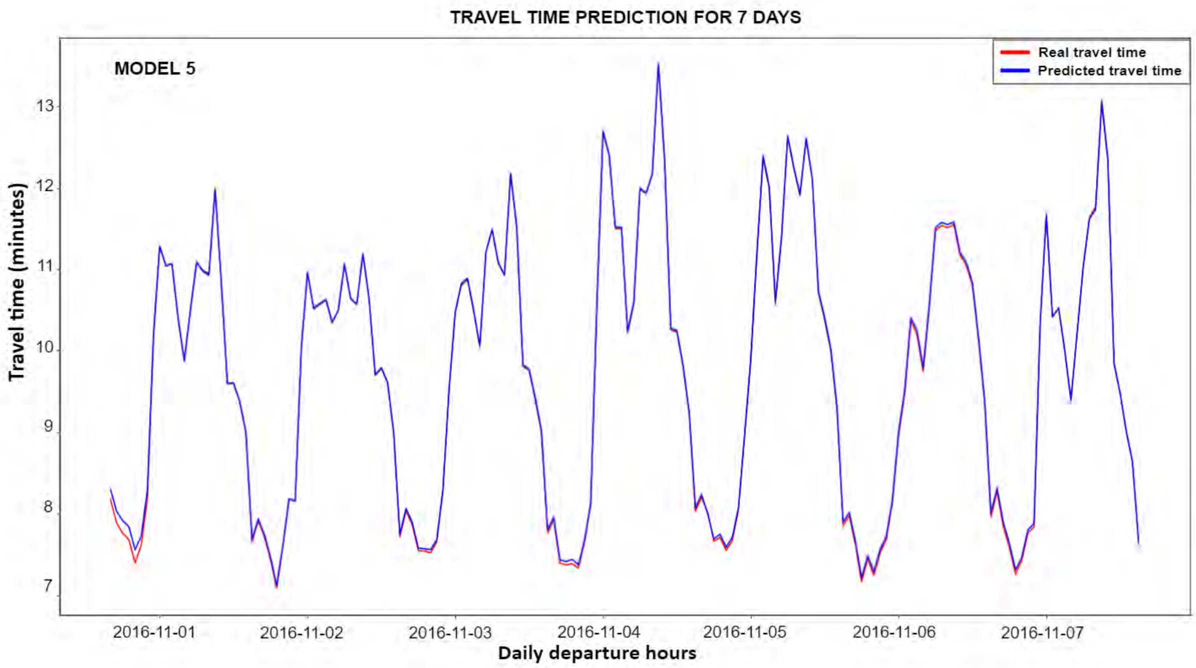

Figure 116: Long Short-Term Memory (LSTM) model 5 real travel time prediction per hour

4.4 Computational costs of developing the deep learning models

This section discusses the computational costs of developing deep learning models to predict the travel time of individual on-demand trips. The computational costs identified include the run-time for training each neural network model, the time expended in data processing before modeling, and the overall time used to accomplish the data processing, feature engineering, model training, and model testing up to prediction with the humongous data collected. Furthermore, the computational power of the computer system used in accomplishing the research is also identified.

Model information and parameters

Table 17 shows the results of the identified computational costs of running each model and the overall data task. Fifteen (15) neural network models were developed, and each neural network has five models. These neural networks include Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). The model information and parameters include the number of epoch or iterations which the models looped through to train and learn the data.

The inputs are the number of identified indicators trained and validated for each model while the trips show the quantity of on-demand trip data that trained and validated, which is slightly over 7 million trips, also equivalent to the number of rows in the training and validation datasets. However, the layers exclude input and output layers and include four hidden layers for ANN, 4 LSTM layers and one convolution, max-pooling, and flattening layers for CNN. These layers' parameters are equal across all models. The model information is essential to understand the structure and intensity of the networks and to relate them with other information like computer system specification and the computation run-time for the tasks.

Overall, 15 neural network models were developed and tested under 250 epochs or iterations on average for each of ANN, CNN and LSTM deep learning methods. A maximum of 64 inputs, also known as travel time indicators, were used for the model training of the seven (7) million (7,000,645) on-demand trips for each of the 15 models. The inputs vary as in Table 17, and there is an average of 4 layers for the architecture of each neural network model, which includes LSTM layers, hidden layers, and other layers stated previously.

Table 17: Computational costs from data processing up to model development and predictions

Deep learning models		Model information and parameters				Computer specifications					Computation time (Run-time = DD:HH:MM:SS)		
Neural Network	Model	Epochs	Inputs	Trips	Layers	Operating System	RAM	Processor	Memory usage	System type	Training & validation	Testing & data processing	Overall computation
ANN	Models 1	50	9	7000645	Hidden layers (4)	Microsoft Windows 10 Home	8.00 GB (7.88 GB usable)	Intel core i5-8250u @1.80GHz, 4 cores, 8 threads	Average of 8.5/250 GB (SSD)	64-bit OS, x64-based processor	00:03:25:20	00:19:55:00	00:23:20:20
	Models 2	50	49	7000645							00:04:10:41	00:00:16:00	00:04:26:41
	Models 3	50	63	7000645							00:04:16:53	00:00:17:00	00:04:33:53
	Models 4	50	64	7000645							00:04:09:18	00:00:17:00	00:04:26:18
	Models 5	50	10	7000645							00:03:47:18	00:00.14:00	00:04:01:18
CNN	Models 1	50	9	7000645	Convolution, max-pooling, flattening & Hidden layers (4)	Microsoft Windows 10 Home	8.00 GB (7.88 GB usable)	Intel core i5-8250u @1.80GHz, 4 cores, 8 threads	Average of 5.0/250 GB (SSD)	64-bit OS, x64-based processor	00:05:00:17	00:00:15:00	00:05:15:17
	Models 2	50	49	7000645							00:10:35:45	00:00:18:00	00:10:53:45
	Models 3	50	63	7000645							00:10:26:28	00:00:18:00	00:10:44:28
	Models 4	50	64	7000645							00:10:32:48	00:00:19:00	00:10:51:48
	Models 5	50	10	7000645							00:05:18:24	00:00.14:00	00:05:32:24
LSTM	Models 1	50	9	7000645	LSTM layers (4)	Microsoft Windows 10 Home	8.00 GB (7.88 GB usable)	Intel core i5-8250u @1.80GHz, 4 cores, 8 threads	Average of 5.5/250 GB (SSD)	64-bit OS, x64-based processor	00:11:48:08	00:00:15:00	00:12:03:08
	Models 2	50	49	7000645							00:12:04:29	00:00:19:00	00:12:23:29
	Models 3	50	63	7000645							00:09:00:23	00:00:18:00	00:09:18:23
	Models 4	50	64	7000645							00:09:39:33	00:00:18:00	00:09:57:33
	Models 5	50	10	7000645							00:06:05:43	00:00.16:00	00:06:21:43
TOTAL	15 (TOTAL)	250 (AVG)	64 (MAX)	7000645 (AVG)	4 layers (AVG)	---	7.88 GB (AVG)	---	6.3 GB (AVG)	---	04:14:21:28 (TOTAL)	00:23:49:00 (TOTAL)	05:14:10:28 (TOTAL)

Computer system specifications and capacity

The computer used in running this task and developing the models is a Personal Computer (PC) with a limited computational capacity to process large data set. This computer's identified computational power includes its Operating System (OS), the Random Access Memory (RAM), the processor, memory storage, and the system type. The computer is an Intel core i5 system with a Microsoft Windows 10, a RAM of 8.00 gigabytes (GB), a 1.80 Giga-Hertz processor with 4 cores, and 8 threads. The secondary storage is a Solid-state Drive (SSD) of 250 GB, and it is a 64-bit (x64-based) OS. The computer specifications reveal the computer's ability to prepare the huge data and develop the model successfully, considering the computational time or the run-time for each model.

However, a maximum of 7.88 GB RAM is used during the task, on average for each of the 15 neural network models while an average of 8.5 GB SSD memory consumed for data pre-processing (data cleaning, imputation, visualization up to feature engineering), model training up to the final prediction of the Artificial Neural Network (ANN) models. This SSD memory usage includes as part of ANN due to its models being the first set developed. In essence, every task, from pre-processing, aside model training and prediction or testing before developing the first ANN model. In a similar interpretation, 5.0 GB SSD memory is consumed on average to run tasks from the model development up to model prediction for each Convolutional Neural Network (CNN) model. Subsequently, 5.5 GB SSD memory consumed in running tasks from model development up to model prediction for each Long Short-Term Memory (LSTM) model. Therefore, overall, an average of 6.3 GB SSD memory ran each neural network task from the pre-processing data stage to model prediction.

4.4.1 Computational run-time for the deep learning models

The computation run-time is the time taken for the python scripts, functions, and classes to process and execute the development of each of the 15 neural network models, right from data pre-processing up to model predictions. There were several breaks (in days) between the run-time for the researcher to recuperate, read more literature, and do more in-depth research on challenges encountered during the data analysis, modeling, and prediction tasks before completion. The data collected is 50 GB data compressed into 10 GB zip folder, as downloaded from Didi and Gaia initiative database web platform, and it contains 60 files in the ".file" format. The zip folder has one origin-destination and one trip route data file, and these two dataset categories were collected every day for 30 days, which made up the 60 files in the downloaded zip file. The approximate time used to extract, convert the file format and process the data before model development are identified in the subsections.

Run-time for data processing and model testing

As shown in Table 17, it took 19 hours and 55 minutes (00:19:55:00) in total from data pre-processing to model testing or prediction, excluding model development and training, for the first ANN model. As discussed earlier, before ANN model 1 was developed and trained, the datasets were prepared for modeling. An estimated run-time of 19 hours 40 minutes prepares the data up to feature engineering and before developing and training the model. This run-time comprises converting the 60 ".file" formats into ".csv" formats, suitable and easy for the Python scripts or codes. Immediately after conversion, these 60 CSV files were merged and separated into two 30-day datasets (the origin-destination and the trip routes datasets), used as training and test data.

Afterwards, it was discovered that the CSV files contain no column labels, but instead, the labels were stated on the website where the data was retrieved. These labels include Pick-up longitude and latitude, Drop-off longitude and latitude, order Id, driver Id, and the timestamps. All these were used accordingly to label the columns of the dataset on the Jupyter notebook with Python language.

Thereafter, the datasets underwent cleaning, graphical visualization, imputation, and feature engineering up to model development. The trip route dataset, which has a size of about 1.35 billion rows or trip coordinates, was sorted into origin-destination by selecting the departure and arrival coordinates based on trip start time, end time, and order Id of each commuter. These consume more RAM, SSD memory and eventually extended the run-time up to the 19 hours 41 minutes (approximately). However, the remaining 14 minutes run-time estimation was utilized for the ANN model 1 testing process. These all explain the reasons for ANN model 1 having the highest run-time for data processing and model testing, of all the 15 models developed.

Subsequently, the run-time for testing each model, ranges from 14-19 minutes depending on the architecture size, the type of neural networks, model parameters, and the computer capacity. In total, 23 hours, 49 minutes (00:23:49:00), expended in running the 15 models from data processing to model testing. A break down of 20 hours 59 minutes run-time for processing the data and testing all ANN models, 1 hour 24 minutes run-time for testing all CNN models, and close to 1 hour 26 minutes to test LSTM models. This computation time shows that almost one day, in total, was used in processing the data and model testing. However, that this run-time was not on a stretch or consecutive, it only represents the time expended in running the python scripts, considering the computer capacity. This eventually shows that more data can be trained with less time and larger computational power to attain higher prediction accuracy.

Run-time for model training and overall computation

The run-time for training each neural network model, as well as the overall run-time, is discussed. Almost 20 hours are used by the python scripts to train all ANN models, while all CNN models had a total run-time of about 42 hours for training and close to 43 hours run-time total for training all LSTM models. This analysis shows that the LSTM took the longest time to train its models but still proves itself the best model with the highest performance, as discovered at the models' comparison section. The total run-time taken to train all the 15 models is four days 14 hours 21 minutes and 28 seconds (04:14:21:28), and this is also dependent on the computation power of the computer used and the model information and parameters.

The overall computation time for the LSTM is the highest among the three neural networks developed, with almost 50 hours run-time (over two days). Subsequently, the CNN total computation time for all the data analysis and modeling tasks is about 44 hours run-time. Finally, the ANN overall run-time is about 40 hours, 19 hours 41 minutes of data processing and preparation, and 20 hours model training and testing. The lowest computational time of ANN is due to its simple network architecture compared to the LSTM with more complex neural network architecture. Therefore, to apply the LSTM architecture or deep learning techniques, a computer with higher computational power and capacity is needed to reduce the massive run-time and, at the same time, predict with high accuracy.

Chapter 5

Conclusion and recommendations

This section concludes this research, focusing on on-demand modeling and forecasting individual upcoming trips with an in-depth study on travel time. The objectives set to achieve this aim are identifying relevant data-driven indicators influencing the travel time of individual on-demand trips, developing and comparing suitable deep learning models to predict the travel time of individual on-demand trips, and identifying the computational costs of developing the deep learning models. Therefore, this concluding chapter discusses the summary of findings and conclusions and highlights the research limitations elucidated in the research methodology chapter above to give recommendations for future research works.

5.1 Summary of findings and conclusions

Identifying relevant travel time indicators

After running a correlation test to determine which indicators are highly significant and influence travel time, more indicators created became influential apart from the commonly used indicators in previous research works. These common ones include departure time, pick-up longitude, pick-up latitude, drop-off longitude, drop-off latitude, travel distance, and travel speed. However, the raw data of the large model training data set collected for this research only comprises of departure timestamps, arrival timestamps, pick-up longitude, pick-up latitude, drop-off longitude, drop-off latitude. These features became the foundation for creating additional indicators, and in total, the data-driven indicators used as model inputs are sixty-four (64).

Regarding this discovery, on-demand modeling and prediction of individual upcoming trips are realizable with travel time prediction by collecting only three (3) information from the commuter, and they include trip departure time, trip origin coordinates, and trip destination coordinates. The travel distance derived from the trip origins and destinations' longitudes and latitudes had the highest positive correlation with travel time. Simultaneously, the travel speed created from the mathematical speed formula using travel distance and travel time had the most top negative relationship with the travel time. The creation of the traffic analysis origin and destination zones are from the use of longitudes and latitudes of origins and destinations. The three (3) commuters' information is the essential data required on the user interface of mobile applications for transport operators. The back-end of the app then automatically generates the sixty-four (64) indicators as model inputs for travel time prediction.

Developing and comparing deep learning models

Three (3) neural networks used to develop the fifteen (15) deep learning models, comprised of five (5) model set each. All the models have five (5) set of inputs whose maximum do not exceed sixty-four (64). Thereafter, the comparison revealed that travel speed has a very significant influence on increasing the models' prediction accuracy. The coefficient of determination values ranges from 0.990 and close to 1 for models 4 and 5 of each neural network, which comprises travel speed as model inputs. Furthermore, the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) rapidly moved close to zero with values up to 0.059 minutes for MAE and 0.364 minutes for RMSE. Therefore, speed is an essential input or indicator for the modeling and predicting travel time to be as close to reality as possible.

After the models' result comparison, the next was the evaluating and comparing the neural networks, which include Artificial Neural Network (ANN), Convolutional Neural Network (ANN), and Long Short-term Memory (LSTM), having the best performance and highest prediction accuracy values. LSTM models 4 and 5 have the best results. The former had the best converging and stable validation loss curve, and the latter had the lowest MAE of 0.059 minutes and RMSE of 0.364 minutes, including its coefficient of determination of 0.997. Therefore, LSTM is the best neural network for deep learning modeling and predicting the travel time of individual commuter's on-demand trips.

Identifying the computational costs of developing the deep learning models

Computational costs identified disclose the researcher's computer capacity and specifications considering the model parameters and information, and computation time for data processing, model training, and model testing. Small computer capacity like that of the researcher has low computational power to prepare, model, and test large data sets with tens and hundreds of millions or billions of on-demand trips. It took the computer a total run-time of 483,028 seconds equivalent to approximately five days and 14 hours long to run the python scripts used for preparing around 7 million trips for modeling, training, and prediction testing.

Therefore, the LSTM models, which have the best performance in travel time modeling and prediction, have the highest memory drive consumption known as Solid-state Drive (SSD) to model and test for prediction, excluding data processing run-time in Artificial Neural Network (ANN) run-time value for model 1. LSTM also has the highest computation time of over 50 hours, which can reduce when computers having high computational power are utilized.

5.2 Limitations and recommendations

The first identified limitation is the researcher's inability to visit the case study area for in-depth research and identify more factors or indicators capable of influencing the travel time of individual on-demand trips. The research methodology in chapter 3 further elucidates the reasons behind it. Accessibility to the study area is essential to retrieve first-hand information on more essential indicators as inputs for the trained models for the travel time prediction. Future research should consider the need to visit the case study area, as this approach will enable improvement in the model training and prediction accuracy.

The second limitation is the researcher's computer's computation power, which is limited and takes a longer time to process huge datasets, train the deep learning models, and test the model for real-time prediction. Future research requires a computer system with high computation capacity of Random Access Memory (RAM), memory drive of Solid-state Drive (SSD), and the computer processor. It will improve the training models' efficiency with more massive data to avoid over-fitting and ensure high learning and prediction accuracy of travel time. It will also reduce the computation time for the python script or any compiled programming language codes and make the modeling and prediction more effective and efficient.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *Google Research*, pages 1–32.

Agafonov, A. and Yumaganov, A. (2019). Bus arrival time prediction with lstm neural network. *Springer Nature Switzerland AG*, pages 11–18.

Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operation Research and Elsevier B.V.*, 223:295–303.

Aloc, D. and Amar, J. A. (2013). Trip generation modelling of lipa city. *Transportation Engineering Group*.

Amita, J., Singh, J. S., and Kumar, G. P. (2015). Prediction of bus travel time using artificial neural network. *International Journal for Traffic and Transport Engineering (IJTTE)*, 4:410–424.

Biggam, J. (2015). *Succeeding with your Master's Dissertation A step-by-step handbook*. Third Edition. Mc Graw Hill Education and Open University Press.

Bogaerts, T., Masegosa, A. D., Angarita-Zapataa, J. S., Onieva, E., and Hellinckx, P. (2020). A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data. *Transportation Research Part C and Elsevier Ltd.*, 112:62–77.

Chu, Z., Cheng, L., and Chen, H. (2012). A review of activity-based travel demand modeling. *COTA International Conference of Transportation Professionals (CICTP) and American Society of Civil Engineers (ASCE)*, pages 48–59.

Didi-Chuxing (2016). Chengdu city second ring road regional trajectory data set. *DiDi Chuxing GAIA Open Dataset Initiative. https://gaia.didichuxing.com (accessed on 5 July, 2020).*

Didi-Chuxing (2020). Frequently asked questions (faqs). *DiDi Chuxing GAIA Open Dataset Initiative. https://outreach.didichuxing.com/appEn-vue/FAQ (accessed on 5 July, 2020).*

Duan, Y., Lv, Y., and Wang, F.-Y. (2016). Travel time prediction with lstm neural network. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems ITSC*, pages 1783–1788.

Edgar, T. W. and Manz, D. O. (2017). *Research Methods for Cyber Security.* Elsevier Ltd.

Ennen, D. and Heilker, T. (2020). Ride-hailing services in germany: Potential impacts on public transport, motorized traffic, and social welfare. In *Institute of Transport Economics Münster Working Paper*, volume 29.

Geopy-Contributors (2018). Welcome to geopy's documentation: Calculating distance. *Geopy Contributors. https://geopy.readthedocs.io/en/latest/module-geopy.distance (accessed on 6 July, 2020).*

Gurmu, Z. K. and Fan, W. D. (2014). Artificial neural network travel time prediction model for buses using only gps data. *Journal of Public Transportation*, 17:45–65.

Hazan, J., Lang, N., Wegscheider, A. K., and Fassenot, B. (2019). On-demand transit can unlock urban mobility. *BCG Henderson Institute. https://www.bcg.com/en-au/publications/2019/on-demand-transit-can-unlock-urban-mobility.aspx (accessed on 9 June, 2020).*

Hiroi, K., Imai, H., and Kawaguchi, N. (2019). Dynamic arrival time estimation model and visualization method for bus trafc. *Springer Nature Singapore Pte Ltd*, pages 155–173.

Hu, L. (2019). *Data-driven analyses of future electric personal mobility.* PhD thesis, Iowa State University, https://lib.dr.iastate.edu/etd/17215.

Jupyter-Team (2015). The jupyter notebook: User documentation. *The Jupyter Team. https://jupyter-notebook.readthedocs.io/en/stable/notebook.html (accessed on 6 July, 2020).*

Karney, C. F. F. (2013). Algorithms for geodesics. *Springer link and J. Geod*, 87:43–55.

Kayikci, Y. (2018). Mobility-on-demand service in mass transit: Hypercommute options. *Turkish-German University, Dept of Industrial Engineering, Istanbul. ORCID:0000-0003-2406-3164*, 4:447–460.

Khosravi, A., Mazloumi, E., Nahavandi, S., Creighton, D., and Lint, J. v. (2011). Prediction intervals to account for uncertainties in travel time prediction. In *IEEE Transactions on Intelligent Transportation Systems*, volume 12:2, pages 537–547.

Liu, Y., Wang, Y., Yang, X., and Zhang, L. (2017). Short-term travel time prediction by deep learning: A comparison of different lstm-dnn models. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) - 978-1-5386-1526-3/17*.

Liyanage, S. and Dia, H. (2020). An agent-based simulation approach for evaluating the performance of on-demand bus services. *Sustainability 2020: Multidisciplinary Digital Publishing Institute (MDPI)*, 12:1–20.

Masiero, L. P., Casanova, M. A., and Carvalho, M. T. M. d. (2011). *Travel Time Prediction using Machine Learning*. Association for Computing Machinery (ACM) and International Workshop on Computational Transportation Science (IWCTS).

Nuzzolo, A., Crisalli, U., Comi, A., and Rosati, L. (2015). Individual behavioural models for personal transit pre-trip planners. *Transport Research Procedia and Elsevier B.V.*, 5:30 – 43.

Nyaki, P. S., Bwire, H., and Mushule, N. K. (2020). Comparative assessment of dynamic travel time prediction models in the developing countries cities. *International Journal for Traffic and Transport Engineering (IJTTE)*, 10(1):96–110.

Oh, S., Byon, Y.-J., Jang, K., and Yeo, H. (2015). Short-term travel-time prediction on highway: A review of the data-driven approach. *Transport Reviews: A Transnational Transdisciplinary Journal*, 35:1:4–32.

Perlman, E. (2015). *Dense Enough To Be Brilliant: Patents, Urbanization, and Transportation in Nineteenth Century America - Market Access, Information Flows*. Boston University.

Python-Community (2020). Python geo-coding toolbox: Geopy v2.0.0. *Python Software Foundation. https://pypi.org/project/geopy/ (accessed on 5 July, 2020)*.

Ran, X., Shan, Z., Fang, Y., and Lin, C. (2019a). An lstm-based method with attention mechanism for travel time prediction. *Sensors and MDPI*, 19:1–22.

Ran, X., Shan, Z., Shi, Y., and Lin, C. (2019b). Short-term travel time prediction: A spatio-temporal deep learning approach. *International Journal of Information Technology Decision Making and World Scientific Publishing Company*, 18:4:1087–1111.

Sathya Prabha, R. and Mathew, S. (2013). Evaluation of operational efficiency of urban road network using travel time reliability measures. *IJRET: International Journal of Research in Engineering and Technology. eISSN: 2319-1163, pISSN: 2321-7308*, pages 155–160.

Shaheen, S. and Cohen, A. (2020). Mobility on demand in the united states: From operational concepts and definitions to early pilot projects and future automation. *Transportation Sustainability Research Center, University of Carlifonia, Beckley and Innovative Mobility*, pages 1–27.

Sharma, S., Sharma, S., and Athaiya, A. (2020). Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology (IJEAST)*, 4, Issue 12, ISSN No. 2455-2143:310–316.

Siripanpornchana, C., Panichpapiboon, S., and Chaovalit, P. (2016). Travel-time prediction with deep learning. In *2016 IEEE Region 10 Conference (TENCON) - Proceedings of the International Conference - 978-15090-2597-8/16*, pages 1859–1862.

Sun, Y., Wang, Y., Fu, K., Wang, Z., Zhang, C., and Ye, J. (2020). Fusion recurrent neural network. In *In Proceedings of 1st ACM SIGKDD Workshop on Deep Learning for Spatio-temporal Data, Applications, and Systems(DEEPSPATIAL 2020).*

Tang, K., Chen, S., Liu, Z., and Khattak, A. J. (2018). A tensor-based bayesian probabilistic model for citywide personalized travel time estimation. *Transport Research Part C and Elsevier Ltd.*, 90:260–280.

Vazifeh, M. M., Santi, P., Resta, G., Strogatz, S. H., and Ratti, C. (2018). Addressing the minimum fleet problem in on-demand urban mobility. *Macmillan Publishers Limited, part of Springer*, 557:534–521.

Viergutz, K. and Brinkmann, F. (2015). Demand analysis and willingness to use new mobility concepts. In *2017 European Transport Conference.*

Wang, D., Zhang, J., Cao, W., Li, J., and Zheng, Y. (2018). When will you arrive? estimating travel time based on deep neural networks. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, volume 32, pages 2500–2507.

Wei, W., Jia, X., Liu, Y., and Yu, X. (2018). Travel time forecasting with combination of spatial-temporal and time shifting correlation in cnn-lstm neural network. *Springer International Publishing AG*, pages 297–311.

Wilhelm, E., Gupta, A., Modurpalayam, S., Wang, S., Walton, B., Wootton, A., Jeruzalski, T., Andrade, A., and Song, S. G. (2015). Autobikes: Autonomous electric bicycles for first and last-mile mobility on demand. In *International Electric Vehicle Symposium and Exhibition.*

Wu, J., Wu, Q., Shen, J., , and Cai, C. (2020). Towards attention based convlstm for long-term travel time prediction of bus journey. *Faculty of Engineering and Information Sciences - Papers: Part B*, 4062:1–13.

Yang, G., Wang, Y., and Li, X. (2019). Prediction of the no emissions from thermal power plant using long-short term x memory neural network. *Elsevier Ltd. and Energy*, pages 1–32.

Yildirimoglu, M. and Geroliminis, N. (2013). Experience travel time prediction for congested freeways. *Transportation Research Part B and Elsevier Ltd*, 53(7):45–63.

Zeng, X. and Zhang, Y. (2013). Development of recurrent neural network considering temporal-spatial input dynamics for freeway travel time modeling. *Computer-Aided Civil and Infrastructure Engineering*, 28:359–371.

Declaration

I hereby confirm that this Master's thesis is my own work and I have documented all sources and materials used. This thesis has not been previously submitted elsewhere for purposes of assessment.

Munich, August 10th, 2020

Abdullahi Olakunle Fatola