Investigation into Urban Traffic Mobility in Malaysia for Intelligent Transportation System

Robin Chan Kuok Cheong

Supervisor:

Ir. Ts. Dr. Joanne Lim Mun-Yee

Co-supervisor:

Professor Rajendran Parthiban

A thesis submitted for the degree of *Doctor of Philosophy* at

Monash University in 2023

School of Electrical and Computer Systems Engineering

Copyright notice

Abstract

Intelligent Transportation Systems (ITS) have fundamental features such as traffic prediction and rerouting, as well as important functions such as traffic data analysis and missing data imputation. This research investigated each of these topics and thus has four main objectives: i) To investigate and model urban traffic patterns in Malaysia, ii) To resolve missing traffic data issues, iii) To accurately and robustly predict future traffic, and iv) To provide an efficient route for vehicles.

Firstly, the proposed mathematical model for the traffic patterns in Malaysian city urban areas, named MOD3D-PAT, has shown a good approximation for the tested maps. The main location tested was Bukit Bintang, with an R-square of 0.93 and an RMSE of 0.035 as the worst result and an R-Square of 0.97 and an RMSE of 0.024 as the best. The suburban area of Malaysia's Damansara Utama has shown similar performance. While Bandar Sunway has shown that sharp and varied peaks in traffic during the evening can affect the results somewhat, the proposed model was still able to provide a reliable approximation of the traffic pattern nonetheless.

Secondly, to overcome the missing traffic data problem, this research has proposed a novel ensemble method utilising ARIMA and Quantile regression forest with clustering for data imputation and tensor factorization for training named AQT. The proposed AQT model performed better than the existing state-of-the-art method, displaying a Root-Mean-Squared-Error (RMSE) of 0.3988, 0.473, 0.5306, 0.6617, and 0.837 for 10%, 30%, 50%, 70%, and 90% missing data respectively. Similarly, the Mean Absolute Percentage Error (MAPE) of the proposed AQT model is 0.8263%, 0.9476%, 1.0711%, 1.365%, and 1.7319%, respectively, for the percentages of missing data specified above.

Thirdly, for traffic predictions, a long-term traffic prediction model was proposed, which utilises data clustering and classification via CNN to resolve the issue of lacking traffic data and features before using LSTM for traffic prediction. This proposed model is called Cluster Augmented LSTM (CAL). The proposed model is compared with existing machine learning models and evaluated using MAPE and RMSE performance metrics. The results showed that the proposed CAL model could achieve better results by 1.42% to 1.76% and 0.25 to 0.41 for MAPE and RMSE, respectively. A comparison between LSTM and GRU was also conducted and showed that GRU tends to outperform LSTM in most cases, but the proposed method with the best performance still utilised LSTM.

Finally, a Decentralized, Intersection-Balancing (DCIB) routing model was proposed for traffic rerouting, which considered the intersections of each road and increased the weights of roads with more intersections compared to those that do, as those roads would be important to ensure both connected and non-connected vehicles that need those intersections are able to make use of it. The simulation was conducted on two urban networks, namely Bukit Bintang and Bandar Sunway, and showed that the proposed DCIB model improved the successful arrivals by 4.4% and 31% for 100% connected vehicles and a 21.4% and 23.8% improvement for 50% connected vehicles for Bukit Bintang and Bandar Sunway respectively.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: ……………………

Print Name: ………………….

Date: ………………………….

Publications during enrolment

Journals

[1] R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban, "Missing Traffic Data Imputation for Artificial Intelligence in Intelligent Transportation Systems: Review of Methods, Limitations, and Challenges," *IEEE Access*, vol. 11, pp. 34080–34093, 2023, doi: 10.1109/ACCESS.2023.3264216. [Published] [Q2]

[2] R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban "Ensemble ARIMA-QRFC with Tensor Factorization Training for Real-Time Missing Data Imputation for Traffic Speed," [Submitted to Future Generation Computer Systems]

[3] R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban "Long-Term Traffic Speed Prediction Utilising Data Augmentation Via Segmented Time Frame Clustering," [Submitted to Expert Systems with Applications]

[4] R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban "An Intersection-Balancing Decentralized Routing Model," [Submitted to IEEE Intelligent Transportation Systems Magazine]

Conferences

[2] R. K. C. Chan, J. M. Y. Lim, and R. Parthiban, "MOD3D-PAT - A novel modified 3rd degree polynomial approximation for modelling traffic congestion in urban areas," *International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2021*, no. October, pp. 7–8, 2021, doi: 10.1109/ICECCME52200.2021.9591081. [Published]

Acknowledgements

I would like to first thank God for blessing me with the opportunity, strength, and good health in order to conduct this research.

I am immensely grateful for the support and patience of my supervisor, Ir. Ts. Dr. Joanne Lim Mun-Yee, and co-supervisor, Professor Rajendran Parthiban has shown me. Your many advice, insights, as well as constant encouragement throughout this research period has continued to motivate me and has been instrumental in the completion of this research.

This research would not be possible if not for the funding provided by the Malaysian Ministry of Higher Education's Fundamental Research Grant Scheme (Grant Number: FRGS/1/2019/TK08/MUSM/03/1) and the facilities and support provided by the Department of Electrical and Robotics Engineering, School of Engineering, Monash University Malaysia.

I would like to thank my friends, seniors, as well as all the lab technicians for all their support and help throughout this research period, such as giving me advice, helping me with my computer in the lab, or just listening to me talk about my research. It has been an honour knowing and working with all of you.

Lastly, I would like to thank my family for their unconditional love and support throughout my studies, for nothing would have begun without them.

I wish everyone the best in the future and good health. Once again, thank you all.

Table of Contents

List of Figures

List of Tables

Table of Notations

Notation	Definition
V	Traffic speed on one road
Vf	Traffic free-flow speed on one road
V_{Avg}	Map average traffic speed
Vf_{Avg}	Map average traffic free-flow speed
N	Number of road segments
D	Number of data points. Subscripts will be attached to further explain the type of data points in the relevant sections.
Ts	The time when traffic congestion begins to increase
t_{peak_start}	The time when the peak jam factor starts
JF	Jam Factor, which describes how congested a road is
JF_{peak}	average peak jam factor between the JF value of t_{peak_start} until it falls to a similar value again
JF_{min}	The average JF from the start of the day until Ts
X	Traffic speed dataset
C	Clustered datasets of traffic speed
x_{pred}	Predicted missing traffic speed data
Q_{pred}	Estimated quantile to use for Quantile Regression Forest
K	Parameter used for K-means clustering representing the number of clusters to be created
W	Weights of the roads in the network.
B	Bounded values, subscripts can be upper or lower to represent upper or lower bound values.

Chapter 1: Introduction

Globally, urban development has seen great growth in recent years, with countries continuously aiming to improve the quality of their city and provide state-of-the-art infrastructures. A key part of developing a successful urban district is to ensure a proper transportation network. However, while it may be possible to plan for future transportation for new areas, old urban districts are subjected to their previous design but are yet subjected to the year-on-year increase of vehicles on the road. While the government could take actions to modify their road networks, it would take time due to the various societal and environmental impact analyses involved, as well as resources, to carry the plans out, during which drivers may encounter even worse traffic conditions than before. As such, a well-implemented system has to be put in place with the current infrastructure in order to resolve this transportation issue, and this global effort has given birth to the idea of an Intelligent Transportation System (ITS).

ITS is the aggregation of various state-of-the-art technologies to bring about a safer, more efficient, and more convenient road traffic environment [1], which would, in turn, increase productivity due to less time wasted on traffic issues such as traffic congestion or accidents. The technologies used include, but are not limited to, communication, information, sensors, cameras, control systems, and machine learning. Studies into improving the various aspects of ITS have continued to grow along with its importance as the global population increases.

The research described in this thesis covers various facets of the ITS functions that affect the drivers, which are traffic state predictions and traffic route suggestions. However, beyond those, there are backend functions which do not affect drivers but are instrumental towards a properly functioning ITS, which are traffic analyses and modelling, as well as ensuring that the traffic data collected is complete and accurate.

The background and motivation of these four aspects, which are traffic analysis and modelling, missing traffic data imputation, traffic prediction, and traffic rerouting, are introduced in this chapter. It also touched upon the problems currently present in these topics and their impact on the ITS. It explained the reasons behind the occurrence of the problems, along with what is being done to resolve them. This chapter also highlighted the problems in the state-of-the-art technologies implemented and clearly defined the research aims and objectives of this research. Finally, the chapter provided the outline of the proposed thesis for the subsequent chapters.

1.1) Background and Motivation

According to the Transport Statistics of Malaysia for the year 2021 [2], the total number of registered land vehicles in Malaysia is greater than 33.57 million, which is very similar to that of Malaysia’s population. Even discounting unused vehicles, it can be certain that the number of vehicles in Malaysia is very high relative to its population. This situation is not limited to just Malaysia but is a global issue. Vehicles on the road are increasing to the extent that traffic management becomes a more pressing issue due to increased road accidents and traffic congestion, among other issues. The reduced barrier of entry to owning a vehicle is also another factor that can be attributed to the current dilemma. Alongside urbanization and the growing interest in online shopping and deliveries, the demand for all kinds of vehicles will continue to increase.

This growth trend has inevitably caused a rise in traffic congestion and, consequently, has led to an increase in vehicle emissions, causing further harm to the environment. While governmental bodies do their best in road planning and reconstruction, these actions take time and resources, which could even worsen traffic problems in the short-term. Furthermore, in order for effective road planning to be done, traffic data and analyses are required. These issues have led to studies being made in various fields in order to resolve the worsening traffic condition. The culmination of the studies in traffic engineering comes down to the development of a smart traffic management framework which is ITS.

ITS, utilising its various technologies, is comprised of multiple functions and leverages various state-of-the-art technologies such as computer vision, edge computing, data mining, Internet of Things (IoT), as well as deep learning. The functions have been developed through the studies of various researchers after delving into various facets of traffic management, such as traffic management systems ([3], [4]), traffic pattern analyses and modelling ([2], [5]), traffic forecasting ([6], [7]), traffic control and routing ([8]–[10]).

These functions and technologies are integrated into a single system in order to provide an efficient and reliable transportation operation to achieve a single goal: To manage traffic in a more efficient manner. A reliable ITS is important as it will reduce traffic congestion and pollution, increase road safety, reduce traffic violations, and provide reliable traffic data to city planners for urban planning, among many other reasons.

As traffic congestion is an issue that will continue to grow in severity as time passes and the global population continues to increase, it is imperative that the research into improving the existing ITS does not stagnate. There is a constant need for improvement as there are still many studies that can be done to advance the development of the ITS further, especially with how quickly modern technologies are advancing.

Despite the continuous efforts to develop the ITS, there remain issues that continue to plague the industry, such as the lack of traffic analyses into traffic behaviours, missing traffic data, difficulty in prediction future traffic speed, and lack of an optimal traffic rerouting model.

1.2) Problem Statement

The first issue of traffic analyses and modelling is caused by the reliance on machine learning in obtaining accurate traffic pattern predictions, as machine learning methods can provide a model that would model the non-linear relationships between the observed variable and the traffic state with high accuracy. However, while this method is indeed able to prove the relationships and correlation between traffic parameters, it does not translate well into a more understandable manner like an explicit mathematical formula. Hence, this method does not improve the existing understanding of traffic behaviours in that respect. Furthermore, the related parameters tend to be specialised to the particular locality where the traffic training data was obtained, which may differ depending on the location. Future machine learning models can be improved upon by further understanding and incorporating the laws between these traffic parameters, as highlighted in [11]. The fundamental aspect of traffic management is to first have an understanding of traffic, such as its patterns and behaviours. In-depth analysis of traffic patterns can lead to better urban and road planning, as well as help the ITS to predict, detect, and respond to possible road problems that will or have occurred. Hence, it is important for there to be a model that would provide a general understanding of said traffic patterns, especially in an urban setting.

However, although there are studies done in order to analyse traffic congestions in certain areas such as [12], a mathematical model to approximate the related traffic behaviour with regards to the time of day has not been proposed prior to the research done in this thesis, to the best of the author's knowledge. While there could be classifications of traffic demand patterns or machine learning models that aim to capture similar traffic patterns like [13] for other traffic studies, to the best of the author's knowledge, there is no direct mathematical modelling of traffic patterns using these parameters. Additionally, the time of the day shows higher granularity than when basing off just traffic classifications. Furthermore, machine learning methods that are used to model traffic patterns do not explain the relationship between the various parameters and could also be too specialised based on the training dataset. This is especially troublesome when trying to design similar traffic patterns like a generalised traffic model, as with the use of online traffic APIs, it is now possible to obtain traffic data for a variety of locations, but having to repeatedly train the model for each separate location is not efficient nor beneficial in the long run. The model proposed in this research would give further insight as to what parameters may affect traffic behaviour and how they would affect the traffic at different times of the day without the need to retrain any dataset and only with some local traffic parameters as input. It is believed that having a mathematical model to represent the traffic congestion behaviour would turn out to be a useful tool in assisting further studies in understanding traffic behaviour and, ultimately, overcoming traffic congestion.

Secondly, as ITS relies heavily on collected traffic data, whether via sensors or communication networks, it is prone to problems such as missing data. The occurrence of missing data is seen to be a disruption to the data analyses conducted by the ITS, where mishandling of it would result in a lack of reliable data and incorrect analysis and prediction. Hence, missing data imputation has been a constant field of study in various fields of data analytics due to the challenge it poses. This is true for various fields, including ITS. In the field of ITS, however, real-time data imputation is especially important as traffic management is done in real-time, and a slow data imputation would end up redundant. Although there are missing data imputation methods that impute missing data after the collection is done, such as the ones in [14], [15] there are fewer studies that predict and impute missing traffic data in real-time, and those that do tend to suffer from high computation times. Furthermore, traffic data has a heteroscedastic nature, which makes imputing such data using normal linear regression methods result in less accurate results.

This research has proposed an ensemble model making use of the strengths of different models to resolve the various traffic-related issues when it comes to missing data imputation, such as the real-time problem and the heteroscedastic nature of the traffic data. In addition, there are different definitions of missing data between different studies, making things confusing. This thesis has defined three different missing data cases, as shown in the later chapters.

The third issue is related to one of the most identifiable functions of ITS, namely traffic forecasting. Alongside traffic routing, these two functions both directly affect the choices of drivers. Due to this, traffic forecasting is one of the most common types of research done when it comes to traffic engineering. Unfortunately, while there are many studies relating to short-term traffic forecasting, such as [16]–[20], there are significantly fewer studies concerning long-term traffic forecasting, such as [6], [21], [22]. It is important to establish the fact that the thesis defines short-term traffic predictions to be traffic predictions of an hour ahead or less, as shown by traffic engineering literature such as [16]–[20], with traffic prediction horizons of 10 minutes at the shortest and an hour at the longest. In fact, some literature like [6] even defines hour-ahead predictions as long-term, which shows that the definition between short and long-term traffic prediction can vary around the hour-ahead mark. Meanwhile, [21], [22] have the longest traffic prediction to be 24-hour or day-ahead predictions, which is what the following research would be basing its definition of long-term traffic prediction on. The lack of long-term traffic forecasting studies could likely be due to the significant drop in accuracy when predicting beyond a certain horizon. As the sensitivity of error propagation increases for long-term prediction [22], long-term traffic prediction is widely considered to be more challenging compared to short-term traffic prediction [23]. This problem is exacerbated by the fact that many prediction models require a high number of traffic features to obtain an accurate prediction. However, not every area has many traffic features available to be collected, and this would result in drops in the performance of the traffic prediction model.

Despite that being the case, these predictions should not be neglected as they could help provide insight into traffic further into the future, giving traffic authorities enough time to prepare a countermeasure in the event traffic intervention is required. Beyond that, [24] talks about the lack of studies on long-term traffic prediction and how this information could be used as an additional feature input to improve existing short-term traffic prediction models. Building on that, [25] has shown that including long-term temporal data segments would improve traffic predictions, which proves the aforementioned point. Furthermore, short-term traffic forecasting is only practical when one plans to depart to their destination immediately and not when planning the departure itself. Planning a journey ahead of time could save a person from being involved in traffic congestion and reduce travel time.

Furthermore, the dataset used in the research is obtained via online data using online traffic APIs, making it more easily accessible and flexible in terms of the location of interest. Although it has its caveats, it is hoped that by using such a method of obtaining data, further studies could be done without worries of having to invest in too many resources or limitations. Also, proper use of data augmentation can help to alleviate the issue of lacking traffic data features. This has been done via the proposed traffic speed prediction model, which makes use of a novel traffic data segmentation and clustering method in order to augment the existing traffic data and obtain a more accurate long-term (One day ahead) traffic speed prediction.

Finally, navigational software has become an integral part of people's lives as they use it to navigate from their origin point to their destination, where even drivers familiar with the route there opting to use it due to its features of making use of current and predicted traffic information in order to provide a more optimal route. Furthermore, the rise in ride-hailing services such as Grab [26] being used has also led to increased usage of these traffic routing services. However, obtaining an optimal routing is difficult as there are various factors to consider, but the criteria most studies focus on is the travel time. This makes sense as all rational drivers would take the shortest path, but this may not be the most optimal and would eventually lead to worsening traffic congestion instead.

Rather than just focusing on the travel time, rerouting models should also investigate the availability of the roads and how important these roads are for certain routes based on their intersections. This is what this research has proposed: an intersection-based rerouting model that considers the intersections along the road and assigns weights according to that in addition to the current prediction of traffic congestion. This frees up important roads and allows vehicles that truly need it to have priority.

A summary of the research gaps and the proposed methods is shown in TABLE 1.1. Further explanations for each topic and their research gaps can be found in the next chapter, which reviews the recent literature for each topic.

TABLE 1.1

Summary of research gaps and proposed methods for each topic of the research

Topic	Research Gaps	Proposed Methods
Traffic Analyses and Modelling	There is a lack of mathematical analysis of traffic patterns and behaviour. While machine learning methods are able to prove that there are indeed correlations between traffic parameters, it does not show the relationships clearly by translating them to a more traditional mathematical model. Additionally, the trained model may be specific to the locality of the data trained. No direct mathematical modelling of traffic patterns based on the time of the day which provides better granularity than traffic demand patterns.	Proposed a mathematical model that generalises traffic patterns based on the time of day and their local traffic parameters.
Missing Traffic Data Imputation	Accurate missing data imputation methods tend to be completed after data collection, and the missing data is not imputed in real-time, which is a requirement in ITS.	Proposed an ensemble missing traffic data imputation model that is robust towards various levels of missing data while maintaining real-time capabilities.
Traffic Speed Prediction	There is a lack of long-term traffic speed prediction due to its inaccuracies. Short-term traffic forecasting alone is not enough for drivers to make informed decisions when planning their trips.	Proposed a long-term traffic prediction model utilising a novel traffic data segmentation and clustering method in order to provide data augmentation and accurate long-term traffic speed prediction.
Traffic Rerouting	Current studies focus on the shortest travel time and ignore the importance certain roads have towards vehicles whose route requires using that road with no other alternative.	Proposed a traffic routing model that takes into account the number of intersections and assigns a higher weight to those in order to reduce the usage of these roads by vehicles which have other possible routes available to them, opening up the road for vehicles that truly require them.

1.3) Research Objectives

The research gaps mentioned were referenced throughout the research, and the resulting proposed methods were produced by keeping each of these gaps in mind. These proposed methods were developed to achieve four main, overarching objectives based on the aforementioned research gaps. The four objectives correspond to each phase of the traffic rerouting system, which are traffic analyses and modelling, missing traffic data imputation, traffic speed prediction, and, finally, traffic rerouting.

The descriptions for the four objectives are listed below:

i. To investigate and model Malaysia's developing cities' travel behaviours using online network information.

ii. To design an accurate real-time missing data imputation method utilising machine learning.

iii. To develop an accurate long-term traffic prediction model by implementing machine learning models.

iv. To formulate an efficient traffic routing model in order to reduce travel time and air pollution.

These objectives will be done with a focus on the Malaysian urban road network and traffic data but are aimed to be used for any other road networks. For the sake of the scope of the research, only Malaysian urban road networks are considered, with a focus on Bukit Bintang due to it being situated in Kuala Lumpur, which is the capital of Malaysia and is a very busy area due to it being a shopping and entertainment district. Other noteworthy locations looked at are Bandar Sunway and Damansara Utama, although to a lesser extent due to fewer collected traffic data.

Chapter 2 discussed these research gaps in more detail in order to give a clearer idea of the recent state-of-the-art models for these research topics and justify the research objectives mentioned in this section. Additionally, the following chapter also reviews popular traffic simulators to determine which traffic simulator is the best fit to achieve the objectives of this research.

1.4) Scope of Work

Figure 1.1 shows the scope of work done for each research objective and a summary of their contribution, topics of focus, tested scenarios, and evaluation criteria, providing a brief overview of the works mentioned in the following chapters.

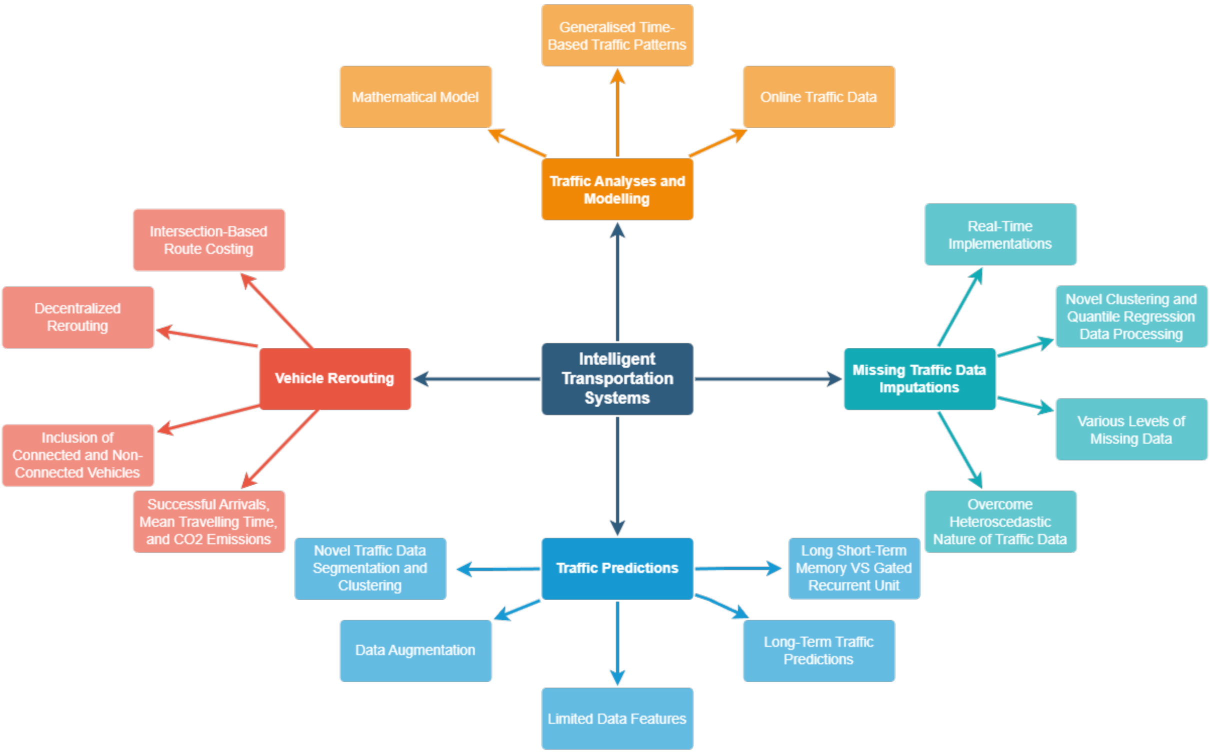

Generalised Time-Based Traffic Patterns
Mathematical Model
Online Traffic Data
Traffic Analyses and Modelling
Intersection-Based Route Costing
Real-Time Implementations
Decentralized Rerouting
Novel Clustering and Quantile Regression Data Processing
Intelligent Transportation Systems
Vehicle Rerouting
Missing Traffic Data Imputations
Inclusion of Connected and Non-Connected Vehicles
Various Levels of Missing Data
Successful Arrivals, Mean Travelling Time, and CO2 Emissions
Overcome Heteroscedastic Nature of Traffic Data
Novel Traffic Data Segmentation and Clustering
Traffic Predictions
Long Short-Term Memory VS Gated Recurrent Unit
Data Augmentation
Long-Term Traffic Predictions
Limited Data Features

Figure 1.1. Summary of the Scope of Work of the Research

Chapter 2: Literature Review

In order to achieve the research objectives mentioned in Chapter 1.3, various research and reviews had to be done as preliminary work, and their research gaps were identified. The following subsections reviewed topics relevant to each of the listed objectives above and talked about their research gaps in more detail while reviewing recent works for each topic. Additionally, a review of the traffic model simulator was also included to justify the traffic simulator used in this research.

Through the review of recent literature, the significance of the research objectives is highlighted by thoroughly analysing the recent works and identifying their strong and weak points. Working in tandem with past research is how future research is able to improve, and this research is no different.

2.1) Traffic Models and Simulation

To validate the proposed methods and ensure their reliability, a suitable environment must be used. As it is not feasible to put into practice every solution proposed in a thesis, traffic simulation models have been developed. These traffic simulation or agent-based modelling software packages are tools used in traffic studies that mainly focus on road designs such as road layouts as well as traffic light regulations, as well as traffic studies involving the validation of car-following models [27].

2.1.1) Traffic Simulation Software

Traffic simulation software is what provides researchers with a semi-realistic environment in which they are able to simulate and test their proposed traffic prediction or routing algorithm in different environments. By simulating their models and algorithm using reliable traffic simulation software, researchers are able to obtain reliable results and confirm the robustness of their model or algorithm. While there are various software popularly used in research for traffic simulation purposes, the author will prioritise open-source software rather than commercial when reviewing the software as researchers are able to customise the open-source codes if they need to, as well as taking the issue of funding into account.

(A) Mobility Simulators

Mobility simulators use Agent-Based Modelling (ABM) to simulate the activities of various autonomous agents and their interactions in a network. In the case of traffic studies, it is generally a road network or a grid network to simulate vehicle behaviour.

There are many variations of open-sourced mobility simulator software in active development, such as Simulation for Urban Mobility (SUMO) [28],[29], MATSim [30], Repast Simphony [31], SimMobility [32], and NetLogo [33].

Among the open-sourced mobility simulators, SUMO, MATSim, and SimMobility specialise in traffic studies, while Repast Simphony and NetLogo are built with a more general purpose-focused agent-based modelling in mind.

Looking at the documentation, SimMobility is lacking when compared to SUMO and MATSim, as can be seen from their Github page [34] and as such, it would be difficult for inexperienced users to make use of. Meanwhile, MATSim takes an activity-based approach when processing agent activities, which may lead to better representability of traffic behaviour. On the other hand, SUMO specialises more in microscopic traffic activities and interactions, which MATSim does not simulate [35]. Hence, the choice of simulator between the two would depend on the focus of the research, whether it be behaviour-focused [36] or traffic model-focused ([37], [38]).

(B) Network Simulators

While mobility simulators simulate vehicle movements, network simulators simulate the communication done between vehicles. This is important for studies that require data to be transmitted to vehicles, be they from other vehicles or roadside infrastructures.

One of the most popular network simulators is the Network Simulator series. Prior to 2010, Network Simulator Version 2 (NS-2) was shown to be one of the most popular choices for network simulators due to its support for most MANET (Mobile Ad-Hoc Network) protocols and implementation of the IEEE 802.11 MAC (Medium Access Control) protocol among other features, although it may not be the best option [39]. However, the development for NS-2 stopped in 2010, and while there are still studies that utilise NS-2 [40], it is recommended to use the third version, which was completely rebuilt, which is NS-3, when taking recent and future technology into account. It is important to note that NS-3 is built primarily for the Unix/Linux environment and is not suitable for the Windows platform without using roundabout methods such as virtualization or Windows Subsystem for Linux.

OMNET++ is another popular choice that rivals NS-3. While not exactly a network simulator, it is an extensible, modular, component-based C++ simulation library and framework primarily for building network simulators [41]. This makes OMNET++ a very good choice when integrating with other simulators. Beyond that, there is the Veins framework designed with OMNET++ and SUMO as the base [42]. This, coupled with an easier implementation on Windows systems, makes OMNET++ a more easily accessible choice for researchers compared to NS-3.

(C) VANET simulators

Various vehicle networks are made with different forms of infrastructures in mind, such as Vehicle-to-Vehicle (V2V), in which vehicles transmit data to one another directly utilising their On-Board Unit (OBU), Vehicle-to-Infrastructure (V2I) where vehicles transmit data to roadside infrastructures such as a Roadside Unit (RSU), and other Vehicle-to-Everything (V2X) which encompasses all types of vehicular communication systems which includes Vehicle-to-Network and Vehicle-to-Pedestrian. Generally, routing protocols use either V2V or V2I protocols, [43] reviews several infrastructure-based routing protocols which use V2I communication. Among these, Vehicular Ad-Hoc Network (VANET) refers to vehicular networks that are formed without the support of infrastructures [39]. However, a lot of research includes V2I communication in the VANET communication scheme, which reiterates the fact that both V2V and V2I are what researchers focus on when dealing with the topic of traffic management. The VANET is what allows this communication between vehicles and infrastructures to work [43].

To simulate a VANET environment in a traffic simulation, a mobility simulator and network simulator are combined ([39], [44]). The need to simulate the VANET environment in various traffic studies would, in turn, narrow down the potential candidates for mobility simulators as well as network simulators among the many choices available. As shown by [45], SUMO looks to be a popular choice for VANET simulators, seeing as it is used in conjunction with both OMNeT++ [46] and NS-2 [47] via the Veins [48] and TraNS [49] frameworks, respectively. As the development of NS-2 has stopped and the focus has turned to NS-3, researchers can look into VSimRTI [50] and ITetris [51] for integration between SUMO and NS-3. It is important to note once again that NS-3 is specifically for the Unix/Linux environment and not for Windows.

(D) Selected Traffic Simulation Software

When taking into account software support, documentation, as well as the accessibility of the traffic simulator, the author has decided to use SUMO as the traffic simulator of choice. SUMO is very powerful and is capable of being integrated with various other frameworks, such as OMNeT++ and NS-2 if required, although NS-2 is much less of a consideration due to halted development. Other traffic simulators are either more commercial-focused or do not have comparable support and ease of use compared to SUMO, as discussed above.

2.1.2) Traffic Modelling

As traffic continues to become a growing concern in the world, so is the need for further improvements in existing Intelligent Traffic Systems and a deeper understanding of traffic behaviour.

To date, various simulations and models have been made in an effort to understand further and facilitate studies in traffic. Starting with the fundamental macroscopic traffic flow theories by Greenshield [52], other traffic theorems have been derived, such as the Lighthill-Whitham-Richards model [53]. These models describe the traffic flow on the road based on the traffic flow, speed, and density of the road.

Furthermore, many traffic simulation software has also been developed, such as Simulation for Urban Mobility (SUMO) [28], for researchers to visualize and test out various traffic control theories or models.

However, these studies only focus on the relationship between the vehicles' speed, density, and flow on the road. There aren't many studies done on modelling the behaviour of the traffic congestion of the road itself, and as far as the authors are aware, there are no mathematical models similar to the traffic flow theories that model the traffic congestion of the road based on the time of the day, utilising traffic flow parameters.

As far as traffic modelling is concerned, the most popular research tends to be car-following models [54]. Besides that, there have been studies done in modelling the velocity-density function [55] as well as the density-flow fundamental diagram [56], both of which took a polynomial approximation approach to the matter.

Although there are studies in traffic congestion prediction, these studies utilise machine learning or other predictive methods, as can be seen by the various machine learning methods such as [3], [5], and [10], as well as the popular time-series traffic forecasting methods like ARIMA [59]. While these machine learning methods are very powerful and accurate, they are not without their drawbacks. These methods work in a black box-like manner and do not clearly indicate the relationship between the various parameters used and are also prone to overfitting if the training is not done with care or is not varied enough. While requiring more work, a mathematical model allows for more generalisation to be made with regard to traffic pattern behaviours while also indicating the relationship between the various parameters.

Having a mathematical model that describes how certain traffic parameters affect one another would be helpful in obtaining a deeper understanding of traffic behaviour, especially for developing countries like Malaysia, where urban development is always ongoing. This is in line with the research objective mentioned above, which is to investigate and model Malaysia's developing cities' travel behaviours using online network information. TABLE 2.1 summarises the comparison between traffic models obtained via machine learning and mathematical modelling.

TABLE 2.1

Table of Comparison between machine learning and mathematical models

Traffic Models	State-of-the-Art	Research Gap
Machine Learning	Very flexible and able to include various parameters in order to achieve a highly accurate prediction model of behavioural traffic patterns	Essentially works as a black box whereby it is uncertain how some parameters are related to one another. Machine learning methods may also result in overfitting to occur, resulting in a non-generalised model. This makes the resulting model unsuitable to be used for any location other than the one on which the training dataset is based on.
Mathematical	Shows a clear relationship between different parameters and how each parameter affects the other with regard to traffic pattern behaviour.	It is less flexible than machine learning in that deeper analysis has to be done to incorporate a new parameter properly into the model.

2.2) Missing Data Imputations[1]

Missing data can occur due to various reasons, such as random error, sensor malfunction, or network error during transmission, and it has a detrimental effect when performing data analytics. This problem is also not uncommon in the ITS, necessitating research to be conducted to address the problem.

There are various ways to handle missing data, as mentioned by [60], such as deletion-based methods, learning methods utilising complete and incomplete data, as well as imputation methods. This thesis focuses on data imputation methods as deletion-based methods would result in a lack of data, and machine learning methods might not provide an adequate understanding of the dataset depending on the methods used.

One of the most widely used methods in recent years for missing data imputation is the use of tensor factorization [15], [61]–[65]. There are also studies that utilise other time-series prediction methods, such as Prophet [14] or cluster-wise linear regression [60]. However, the studies utilising the methods listed above require the use of a dataset that has already been collected (e.g., the data points before and after the missing data points are known) and are not suitable for real-time applications. This is because missing data occurring in real-time should ideally be imputed as soon as it is registered. There are also neural network and deep-learning methods such as [66], [67]. Although it is possible to utilise these methods for real-time situations, it may not be ideal due to the computation time required for updating and training the models.

[1] Part of the content of this subsection appears as "R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban, "Missing Traffic Data Imputation for Artificial Intelligence in Intelligent Transportation Systems: Review of Methods, Limitations, and Challenges," IEEE Access, vol. 11, pp. 34080–34093, 2023, doi: 10.1109/ACCESS.2023.3264216."

On the focus of real-time traffic missing data imputation, [68] has proposed a PPCA-based minimum data imputation optimization method that ignores certain missing data points that it deems not required to be imputed, along with simplification of the spatial correlation between road segments on the map. However, not every country has a well-built traffic infrastructure that would provide clear road segment data, thus hampering the effectiveness of data imputation methods requires the use of spatial data. Besides that, although the effects may be small, missing data should be imputed to ensure the completion of the data set and to prevent possible bias in prediction results down the line.

Other concepts to be aware of in missing data imputation research are the types of missing data and how these missing data may occur in a real-life scenario. The subsections below will define the classification of missing data used in this research, as well as review popular methods used in recent years to conduct missing traffic data imputation. Additionally, a more comprehensive literature review was conducted on the subject and published in IEEE Access [1].

2.2.1) Types of Missing Data

Existing missing data imputation research has differing classifications for similar types of missing data. For example, [50] describes random missing data, along with two other types, namely univariate missing data and multivariate missing data. In other papers, such as [4], [51], and [52], univariate and multivariate would be named fibre and block or panel missing data, respectively. Other papers might have also given overlapping or different names for similar kinds of missing data, such as continuous missing data to represent fibre missing data [53].

For the sake of unification, these missing data types should be defined and generalized in order to help simplify the direction of future research. The general idea of the three categories is as mentioned below, visualized in Figure 2.1:

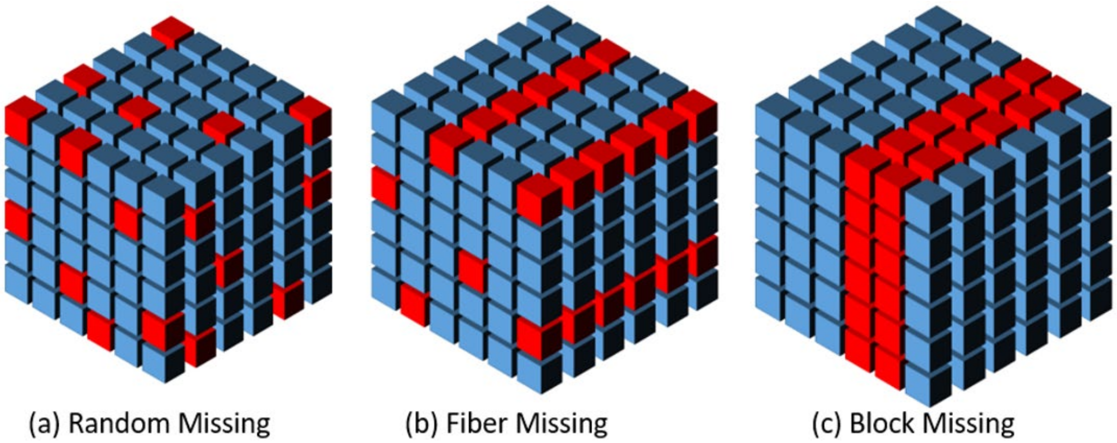


Figure 2.1. Visualization of Different Categories of Missing Data where the dark cubes represent the missing data. (a) Random Missing Data, (b) Fibre Missing Data, and (c) Block Missing Data

i. **Random Missing Data**: Missing Data is caused by sporadic errors in the transmission of which there is little to no correlation known between the data loss and other variables. Results in missing data at random points in the dataset.

ii. **Fibre Missing Data**: Missing data is caused by a sudden, temporary failure in connectivity or in the data-capturing device, resulting in long periods of missing data. Results in missing data for a length of time.

iii. **Block Missing Data**: Missing data caused by the absence of a detector in the area of interest (i.e., A rarely used arterial road that does not justify the installation of a loop detector [4] or all sensors are not in operation for some reason). Results in complete missing data for the entire length of time over a long period or complete missing data from all sources of information for the same time horizon. This is seen in datasets with multiple sources of data.

While random missing data can be further broken down into three more types, namely, Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR), simulations are usually done in an MCAR situation, such as [54]. [55] has also stated that MNAR is generally not considered as well. Hence, for most research, MCAR is the general test case, followed by fibre and block missing.

It is important to note that block missing data imputation is not researched much, probably due to the significant lack of data as well as some research deeming that the areas with these levels of missing data do not contribute much to the overall traffic state.

2.2.2) Missing Data During Data Collection

When it comes to monitoring systems such as sensors and cameras, the main reason missing data occurs tends to be equipment malfunction and electrical breakdowns, which leads to loss or damaged data [55]. Early detection leads to this being a case of fibre missing data, and failure to do so causes it to devolve into block missing data. Public or private datasets which use similar monitoring systems would also be subjected to similar issues. However, as the data has already been collected in the past, it is trivial to ignore datasets with missing data and select the ones for which the dataset is complete. Online traffic APIs might face similar issues, but applications like HERE Traffic API [23] or Google Maps [21] would have more than one source of data to ensure the integrity of their data, such as floating car data (FCD) or probe vehicle data from a fleet of connected vehicles via GPS services or applications [25], although even then, there are times where missing data can occur with an online service.

Random missing data can be caused by sporadic errors due to aged electrical parts or packet drops during the transmission of data, causing data loss or corruption for an element in the dataset. It tends to be spread out and is not obviously affected by the environment.

2.2.3) Missing Data Imputation Methods

Past literature reviews a specific aspect of missing traffic data imputation, such as [69]–[72], usually focusing on the results but largely ignoring other aspects, such as the road networks or missing data types involved. This section reviews the popularly used methods, broadly categorized into two methods, along with looking into the type of road networks and missing data scenarios used in various literature.

There are generally two categories of missing data imputation methods: statistical and machine learning. Statistical methods refer to the more classical methods of utilising mathematical models and statistical theories to impute the data, whereas machine learning makes use of modern computational power and big data to better learn the non-linear, latent features and patterns in a dataset and attempt to learn and output the most likely result based on an input from a similar dataset.

(A) Statistical Methods

Statistical methods analyse the available data and aim to develop a model that best represents the original dataset. Unlike machine learning, which makes use of big data to learn, it is less necessary for statistical methods to need such a large amount of data at the cost of being less robust in general.

There are various ways to handle missing data, as mentioned by [60], such as deletion-based methods, learning methods utilising complete and incomplete data, as well as imputation methods. Mean smoothing has also been used in studies such as [73]. On the other hand, deletion-based methods tend to be avoided as deleting data may result in bias in the estimates and decrease the quality of the dataset itself [74]. Note that deletion-based and mean smoothing represents the simplest methods and are usually not used in missing traffic imputation studies.

Regarding learning methods, predictive mean matching (PMM) based on multiple imputations by chained equations (MICE) has been looked into in [75]. A study done later on has then proceeded to compare variations of PMM methods, including MICE, Classification and Regression Trees (CART), Least Absolute Shrinkage and Selection Operator (LASSO), and random forest, with the result being the MissForest implementation of Random Forest being the best performer [76]. It is noteworthy that random forest is considered a machine learning algorithm, which shows why machine learning tends to be researched more compared to statistical methods, especially in recent years.

Instead, two of the most popular methods for missing data imputation would be Probability Principal Component Analysis (PPCA) and tensor decomposition. These methods are explained below:

(I) Probability Principal Component Analysis (PPCA)

The most commonly used statistical method when it comes to data imputation is the PPCA-based (Probability Principal Component Analysis) model. PPCA is an extension of the Principal Component Analysis (PCA) method through the use of the expectation-maximization algorithm [77]. The resulting probability model results in the ability to better deal with missing data by treating the missing data as not-yet-observed missing data [78].

Recently, [69] has excellently reviewed spatiotemporal PPCA-based data imputation methods in an urban network setting for traffic flow data. As expected, the accuracy of the PPCA-based model changes depending on its field of view, i.e., whether it is a network, sub-network, or single-point imputation. Interestingly, if the view is too large, the result would drop, resulting in more inaccurate results. It was found that for a more realistic use case, the sub-network PPCA-based model worked the best for an urban road network as it is within a reasonable range of detectors.

Focusing on real-time missing traffic data imputation, [68] has proposed a PPCA-based minimum data imputation optimization method that ignores certain missing data points that it deems not required to be imputed, along with simplification of the spatial correlation between road segments on the map. However, not every country has a well-built traffic infrastructure that would provide clear road segment data, thus hampering the effectiveness of data imputation methods that requires the use of spatial data. Furthermore, although the effects may be small, missing data should be imputed to ensure the completion of the data set and to prevent possible bias in prediction results down the line.

[78] also conducted a case study on the PPCA model for traffic analysis, data imputation, and flow prediction, and while the missing data rates tested were not large (1.4%, 4%, and 33% missing data rates), it was found that the PPCA did not show a large degradation in performance when comparing the Weighted Mean Absolute Percentage Error (WMAPE) between the 1.4% and 33% missing rates, which showed around a 1% drop in accuracy from 1.4% to 33%, meaning it is overall robust. However, the initial WMAPE itself is rather high at around 14.75%. Despite that, the case study also exhibits the strength of statistical methods, namely the ability to conduct traffic analysis via a breakdown of its principal component scores. While it seems that PPCA was not used much for missing traffic data imputation, it should not be ignored due to its analytical ability, which could contribute to the advancements of itself as well as other techniques.

Additionally, a comparison between MICE and PPCA was made for missing data imputation in the healthcare sector [79], and PPCA was found to have performed better as well, further explaining why this method is one of the more popular statistical methods.

(II) Tensor Decomposition and Factorization

Tensor factorization and its derivatives have seen a significant rise in popularity when it comes to the field of missing data imputation, and missing traffic data is not an exception. This can be seen when comparing the reviewed literature between [80] and [81], noting the tensor factorization methods have shown a spike in use in [81] compared to [80]. In fact, tensor factorization can be considered both a statistical model and a machine learning model. However, tensor factorization is more interpretable compared to other machine learning models because it enables the extraction of a dataset's latent variables via tensor decomposition. Even papers that focus on traffic forecasts, such as [82], make use of tensor decomposition to deal with their missing data before moving on to their proposed model.

Papers such as [63], [83], [84], [85], [86], [87], [15], [65], [88], [64], [89], [90] and [91] are some of the recent state-of-the-art missing data imputation methods that have been proposed in the past three years that have utilised tensor factorization as a core part of their model. These tensor-based models performed well due to their being able to extract latent features from a traffic dataset and, through decomposition and completion, can fill in the missing blanks in an accurate manner. Via Bayesian Statistics ([69], [71], [74]), extending or modifying the existing tensor factorization methods ([83], [85], [86], [65], [88]), and even adding an additional pre-processing method ([64], [91]), the base tensor factorization method has shown a significant improvement in the field of missing traffic data imputation. This can also be seen as a majority of these models have been tested for robustness in imputing missing traffic data of rates ranging from 1% to 90% while still retaining a high level of accuracy when compared to their respective benchmarks. Besides that, out of the 13 papers mentioned, 11 of them ([63], [83], [86], [87], [15], [65], [88], [64], [89], [90], [91]) have also been tested on urban traffic networks, raising the evaluation on their robustness as urban traffic tends to be significantly more complicated than freeways/highways/expressways.

However, tensor factorization methods are largely dependent on their dataset and would be unable to perform similarly if the same trained model is tested in another location without retraining [92]. Besides that, tensor decomposition tends not to scale well with larger datasets [93].

(B) Machine Learning

Data-driven models in machine learning methods utilise the availability of data and learn the best weights to obtain the most optimum result for a certain model, as compared to classical statistical methods, which require prior knowledge to derive an appropriate mathematical expression from a given data trend. In general, the model is trained via a training set to 'learn' the optimum values to output given a certain set of inputs. This is a very powerful tool as it requires little to no supervision from the user, but at the same time, a certain understanding of the model may be lost. However, it could be understood that the underlying features of the dataset have been, in theory, mined via the model, allowing it to be more robust and accurate compared to traditional statistical methods.

Neural networks are the models which are the most synonymous with the term machine learning despite just being a subset of it. Regardless, the idea of neural network was introduced in [94] back in 1943 and only began gaining traction in recent years due to the improvement in computation technology. Now, it is being used in various fields, from classification, prediction, and identification to missing data imputation, among others. The following subsections cover the popular methods used in missing data imputation.

(I) Generative Adversarial Networks

Generative Adversarial Networks (GAN) is a new model proposed in 2014 [95] utilising a Generator and Discriminator model to train a network. To summarize, the generator continuously attempts to 'trick' the discriminator into that the generated data is the same as the trained dataset. This results in both being trained to generate better, more realistic data as well as more discriminatory testing, allowing the overall model to impute missing data more accurately or realistically, in theory. While not as popular as tensor methods, GAN is a fairly popular method in missing data imputation applications due to its nature of constantly training to create a better dataset to trick the discriminator. This can be seen by the recent papers focusing on GAN methods such as [93], [96], [97], [98], [99], and [100]. As with other methods, this research tends to focus on the Spatiotemporal features of the traffic data ([93], [97], [100]) when conducting traffic data imputation. Some utilise the Attention mechanism ([96], [97]). Besides that, [98] makes use of additional external factors such as weather and holiday factor. Interestingly enough, that research found that external factors excluding holidays do not influence the data imputation much for missing rates less than 40%. While researchers tend to test for high levels of missing data, it could be said that missing traffic data would not be that high. In this case, future researchers can focus on methods that improve the missing traffic data imputation at low missing rates with minimal concern that external factors might cause a large discrepancy in their performance. Another interesting GAN model was proposed by [99], whereby the generated result is once again used as an input into another generator, and the discriminator tries to discriminate between the first generated data and the double-generated data.

Despite the potential of GAN models, there are still issues with the evaluation of the model's performance. As mentioned by [101], GAN models have to be critically evaluated in terms of fairness as well as understanding the degree to which the model memorizes the training data. Improper evaluation of the model may lead to skewed results and data imputations.

(II) Graph Neural Networks

Various real-world datasets are represented as graphs, such as from a social network or the internet itself, and traffic data is not an exception. Traffic networks are naturally represented as a graph, as it is a suitable form in which to visualize road connections and their related information. Realizing this, researchers have proposed the use of Graph Neural Networks (GNN).

Recently, [102] has done a comprehensive survey regarding GNN and has classified various GNN models into four categories: Recurrent GNN, convolutional GNN, graph autoencoders, and spatial-temporal GNN. Among these, we have found that convolutional GNNs are the more popular choice in recent times when it comes to traffic research, as shown by [103]**,** [104]**,** [92], [105]. Convolutional GNNs, or Graph Convolutional Networks (GCN), utilise convolutional neural networks to embed graph information into a tensor, resulting in a uniform framework from irregular datasets [104].

While GNN and GCN are popular methods used in traffic studies, most recent research focuses on traffic forecasting, and not as many focus on missing data imputation. Some research, such as [103]**,** treats missing data as part of the traffic prediction process instead of the focus of the problem. This could be useful as traffic actions tend to require real-time analyses and predictions. While it is good to design a robust traffic prediction model towards missing data, having a missing data imputation model should not be neglected as it can further enhance the already robust traffic prediction model. On the other hand, [92] is more focused on missing data imputation, proposing a model that uses a bidirectional recurrent neural network (RNN) to capture temporal patterns and GCN to capture spatial patterns. Meanwhile, [105] proposed a Graph neural network that makes use of the attention mechanism, as well as a temporal convolutional network instead of RNN as standard RNN, which suffers from various drawbacks such as being unable to hold memory for long, prone to vanishing or exploding gradients, and having low efficiency in parallel training and inference. While not exactly imputing missing traffic data itself, [104] combined GCN with a mapping function to impute missing spatial flow data. This is another important aspect of traffic data that the authors believe should be highlighted and received attention, as origin-destination flow data can be a vital addition to other traffic-related models that could use additional traffic features.

Besides GCN, there are also pieces of literature, such as [106] and [107], that make use of spatial-temporal GNN instead. In other words, instead of utilising convolution for feature extraction and graph embedding, the research proposes other methods, such as the fusion of multiple data sources ([106], [107]) or attention mechanism, as well as multitask learning [107].

As the concept of GNN was introduced relatively early in 2005 [108], and GCN itself was only introduced even more recently in 2017 [109], there is still plenty of room for improvement, as can be seen by the recent literature mentioned above. As road networks differ depending on the location, it is imperative to find a model that is robust towards various forms of missing data and the structure of road networks. GNN may have a strong potential in this due to its deep-learning structure as compared to tensor decomposition methods, which might be more transductive.

(C) Ensemble Model

In certain cases, when a statistical method fails to impute missing data accurately, it is normal to fall back to traditional traffic models or the less interpretable machine learning models. Both the traditional and machine learning models have their advantages, with one being more grounded in principles while being cheaper regarding resources. At the same time, another may be more accurate but would require more resources. However, cases like these show that a single model tends to have some forms of shortcomings along with its advantages. In this case, researchers have come up with the idea to combine multiple models to resolve each model's weaknesses and enhance their strengths further.

For example, [110] uses the very popular tensor decomposition but utilises a Fuzzy Neural Network to further enhance the imputation accuracy by optimizing the weights of the tensor resolvers. Besides that, [111] combines GCN and tensor decomposition using graph Laplace for tensor completion.

Meanwhile, [112] designs a framework combining matrix modelling and factorization and conducting matrix decomposition before using a dendrite neural network to fuse the information to obtain the final imputed data. Besides being another ensemble model, the proposed neural network model was recently proposed by [113], of which the code is provided in their paper. This could be another good avenue for researchers to look into as it expands upon the existing neuron structure to further resemble the human nervous system.

As shown, these ensemble models make use of already established methods while modifying them to work together to obtain an even greater result. However, it should be noted that using more models would inadvertently increase computation time, which may result in the inability to function in a real-time scenario.

2.2.4) Missing Data Imputations Review Summary

TABLE 2.2 summarises the state-of-the-art methods discussed in the subsections above and their related research gap. In general, it can be said that while tensor decomposition is strong, it suffers from scalability problems, while machine learning methods tend to require a large number of data and features which may not be available. Ensemble models essentially make use of the strengths of existing models and remove their weaknesses, making them a flexible choice if the model is designed and executed properly. Through analyses of the research gaps, a missing traffic data imputation ensemble model is proposed in order to achieve the research objective mentioned above, which is to design an accurate missing data imputation method utilising machine learning.

TABLE 2.2

Table comparing the state-of-the-art missing data imputation methods of recent years and their research gaps

Methods	State-of-the-Art	Research Gap
Principal Component Analysis	Allows traffic analysis to be done while also providing a good level of missing data imputation	The performance of missing traffic data falls behind the other popular methods.
Tensor Decomposition	Able to impute missing traffic data well. Allows some traffic analysis to be done by decomposing traffic data into latent variables	Faces scalability issues. High computational costs when conducting missing data imputation due to requiring multiple iterations. A large number of data is required to perform optimally.
Generative Adversarial Networks	Able to impute accurate and realistic traffic data due to the adversarial network design.	GAN methods are difficult to evaluate in terms of their bias and fairness, which may lead to skewed data imputation. Fine-tuning the parameters for the network can be exhaustive. A large number of data is required to perform optimally.
Graph Neural Networks	The structure of the network suits that of a traffic network, making certain operations more intuitive could be easier to visualize compared to other neural network-based methods. The graph structure makes GNN more scalable for traffic networks	A large number of data is required to perform optimally. The structure of GNN limits the types of datasets it can work on, although traffic data should be fine due to the similar structure.
Ensemble Models	Makes use of the strengths of multiple models and minimizes or eliminates their weaknesses.	An increase in the number of models may inadvertently result in an increase in computational complexity, which may be detrimental to a real-time system such as ITS. Depending on the type of models involved, it may require a large number of data to perform optimally.

2.3) Traffic Speed Forecasting

Among the various ITS-related studies, traffic forecasting is likely the one that has garnered the most interest. While traffic modelling and missing data imputation are both also important, traffic forecasting directly influences how traffic planners or rerouting algorithms would recommend routes to the drivers. Hence, there are quite a number of studies done on this particular topic, along with their respective surveys.

Surveys have been done that properly categorize existing traffic prediction methods into two broad categories, namely Classical and Deep Learning-based methods [114]. Statistical methods generally require domain-based [115] or apriori knowledge of the model's parameters to perform optimally. On the other hand, machine learning methods do not require such information and are capable of learning on their own such as Support Vector Machines used by [116]. Deep learning takes this one step further and creates multiple layers of networks in order to better extract and learn features from datasets.

While [114] briefly talked about classical methods, it is obvious that the focus of the survey as well as future research, would involve deep learning-based models more due to consistently achieving better performance for data with high complexity, such as traffic data, while also requiring little to no domain knowledge or unreliable assumptions regarding the dynamic and complex traffic data. This is further confirmed by [117] as another survey was done specializing in the implementation of deep neural networks by summarizing the data representation methods as well as the development of deep neural networks from the past to present times.

Surveys like [118] have also been conducted, which properly review work in the traffic prediction life cycle, such as the data categorization and preprocessing, identification of traffic prediction problems and their proposed solutions. While traffic prediction for ITS generally gives the idea of predicting future traffic flow or status, [118] also interestingly mentioned that other forms of traffic prediction include the prediction of future traffic demand for companies providing taxi services.

Recently, researchers have also started looking into hybrid models, whereby different models are integrated in order to strengthen their overall performance and robustness while minimising or eliminating the models' individual weaknesses.

2.3.1) Deep Learning Methods

Thanks to the improvements in technologies as well as more data being made available and accessible, it is now possible to utilise the more computationally heavy deep learning methods with the large quantity of data collected throughout the years. Thanks to these big data, researchers are able to obtain various traffic data in order to design and propose more efficient data-driven methods, which generally make use of machine learning at their core.

Deep learning models are essentially neural networks designed to emulate how a human brain works by utilising many neurons and stacking them in separate layers for separate degrees of feature extraction and learning. The Deep Neural Network (DNN) architecture used by [6] would be the best fundamental representation of such a system, which is an Artificial Neural Network (ANN). Despite that, the more layers or, the deeper the neural network is, the larger the computational power required, and care must be taken to ensure that the model can be run in a time-sensitive environment such as for an ITS. Instead of increasing the number of layers, more novel models have been developed to obtain better results for more specific use cases while using fewer layers, as shown below.

2.3.2) Recurrent Neural Networks

As traffic prediction is reliant on historical data along with the previous traffic trend, there have been plenty of traffic prediction models proposed which make use of Recurrent Neural Networks (RNN) in recent years. These models are especially popular due to how they have the ability to remember older information and are specifically designed for processing sequential data. However, the basic RNN only has short-term memory, which might not be good enough for traffic forecasting. This is where Long Short-Term Memory (LSTM) is introduced, which is an extension of the RNN architecture, featuring an extended memory while also resolving various issues of RNN, such as the vanishing gradient or exploding gradient problems[119] through the introduction of a set of gates [120]. These gates are known as the Input gate and Forget gate.

Examples of the use of LSTM in traffic forecasting are [66] which stacked two custom bidirectional LSTMs (bi-LSTM) or [121] which designed an encoder-decoder model with bi-LSTM as the encoder and decoder layer.

When it comes to time series predictions, Long Short-Term Memory (LSTM) is a very popular choice. This can be seen in the many traffic prediction models that make use of LSTM, such as [122] and [123]. However, although it is a popular choice, it may not necessarily be the best choice. [124] conducted a traffic flow prediction experiment comparing ARIMA, LSTM, and Gated Recurrent Units (GRU), whereby both LSTM and GRU outperform ARIMA while GRU outperforms LSTM on 84% of the total time series. Not to mention, GRU's design is less computationally complex than LSTM [125], requiring fewer parameters, and in fact, an electric load forecasting was done for [126] between LSTM and GRU networks, with GRU performing significantly better in terms of performance as well as computational time. While further experiments should be done, future research should consider looking further into GRU as another method of implementation besides just LSTM. This is even more so since GRU and LSTM are very similar in function.

2.3.3) Convolutional Networks

One of the most used features other than the traffic speed or traffic flow would be the spatial data, as it tends to accompany the traffic data itself. By making use of this feature, many studies have proposed the use of Graph Convolutional Networks (GCN) such as [127] and [128]. Like GCN, Convolutional Neural Networks (CNN) have also been proposed in studies such as the one by [23] in which CNN was used to capture the spatial relationships among links before passing over to the LSTM for traffic prediction. However, cases where spatial data is unavailable or inaccurate, would likely result in a reduction in the performance of the model.

2.3.4) Other Models

Based on the subsections above, there are many studies that make use of the spatiotemporal features of traffic data through the combination of CNN and LSTM or other similar machine learning algorithms, such as [21] which decomposes the traffic data using wavelet decomposition before passing it through a CNN-LSTM for traffic prediction or [129] that makes use of the feature extraction ability of CNN to capture daily and weekly periodicity of traffic patterns across multiple days.

It goes without saying that the majority of studies would preferably make use of more than just the traffic speed as an input to their model. When data features are limited, they would process the data using methods such as wavelet decomposition or improve the training via alternative training methods. One such effective method is using GAN, which was previously mentioned, such as [130], in which the generator aims to generate a prediction that the discriminator would be unable to differentiate between the real and the generated future traffic.

Speed interval prediction is also an option as traffic managers could use these intervals to make decisions as well instead of a single speed value. A study done by [7] did just that by utilising a combined model of fuzzy information granulation and the grey autoregressive model.

Many newly proposed models tend to utilise various models in a combined manner, such as the ones mentioned above, to obtain richer data which results in better training or utilising the different machine learning models to facilitate training and prediction of different aspects of the data before the final value is outputted, future researchers should break down the various steps and consider whether there are individual models that might work better for each step while also keeping computational complexity in mind.

2.3.5) Traffic Speed Forecasting Review

TABLE 2.3 summarises the different methods used in traffic speed forecasting studies and their corresponding research gap. LSTM has shown to be a very popular and reliable method, but with the GRU being developed, it is a good idea to monitor the performance of both of them, especially since they can be used somewhat interchangeably. Meanwhile, it is important to keep in mind the real-time requirement of ITS and the possible lack of traffic data features in certain areas when developing a traffic speed forecasting model. The objective of the proposed model in this research is to develop an accurate traffic prediction model by implementing machine learning models, and it does so by combining LSTM and K-means clustering in a novel traffic data segmentation and clustering method, which will be described in the relevant chapter below.

TABLE 2.3

Summary of the state-of-the-art traffic forecasting methods and their research gaps

Methods	State-of-the-Art	Research Gap
Deep Learning Models	Able to analyse input traffic data in-depth and provide an accurate prediction by making use of the deep learning framework.	Deep learning involves a large computational burden and is not suitable for real-time use.
Recurrent Neural Networks	Ability to remember previous inputs allows for more reliable time-series predictions to be made, making it a popular choice to be used for traffic speed prediction.	The performance of GRU against LSTM should be looked into.
Convolutional Networks	Able to make use of spatial data together with time-series data in its prediction in order to better predict the traffic speed.	In a limited feature data environment, where spatial data may not be available, this method may not perform as well by itself and would require some form of augmentation to perform well.
Generative Adversarial Networks	Able to predict accurate and realistic traffic data due to the adversarial network design.	Fine-tuning the parameters for the network can be exhaustive. A large number of data is required to perform optimally.
Ensemble Models	Makes use of the strengths of multiple models and minimizes or eliminates their weaknesses.	High number of models may result in an increase in computational complexity, which is detrimental to a real-time system such as ITS. Depending on the type of models involved, the resulting model may require a large number of data to perform optimally.

2.4) Traffic Rerouting

Traffic rerouting is the final step in the process of providing drivers with an optimal driving experience, and this usually translates to short travel time and a smooth drive with minimal or no traffic congestion. As urban development continues to evolve, new roads are built, and old ones are replaced or modified, resulting in a bigger and newer road network. Traffic routing is thus becoming an increasingly important tool to have on hand in order to navigate through the ever-changing concrete jungle.

There are many traffic studies conducted with regard to vehicle routing, which by itself has many different scenarios besides typical traffic situations, such as for distribution logistics [131] which focuses on a multi-depot vehicle routing problem or routing, which focuses on the reduction in greenhouse gas emissions [9]. The common end goal of these studies is always to reduce the overall costs required for an agent (i.e., vehicles) to reach its destination. These costs are attached to the path taken and are calculated according to each study's focus. The methods such rerouting models take are usually called dynamic traffic assignments (DTA).

2.4.1) Decentralized Models for Traffic Rerouting

As vehicle routing affects many vehicles, it can be too computationally heavy to rely on a single, centralized network, which has led to many traffic routing studies focusing on decentralized methods instead. A very popular implementation of this is the agent-based model, whereby vehicles act as agents to transport information to other vehicles or infrastructures.

An example of an agent-based method is Ant-Colony Optimization, as employed in [132] which makes use of information in the form of pheromones deposited by the ants, or in this case, the vehicles. Meanwhile, [133] makes use of the multi-agent reinforcement learning model and implements Markov's game paradigm in the way the agents are to decide their routes. In a different vein, [134] utilises a decentralized cluster of connected and automated vehicles (CAV), of which the CAV share information only within the cluster to reduce computational overhead and has shown improved routing performance in terms of travel time when compared to other benchmarks. It is important to note that for decentralized models, coordination and communication between vehicles are very important. Hence, many of these studies assumed that the vehicles used are connected vehicles, or at the very least, are vehicles that are capable of communicating with other vehicles or road infrastructures like roadside units (RSU), such as in [132], [134]. As shown in [134], the fewer vehicles communicate in a network, the less efficient the overall routing model, and this scenario is the more realistic one for the foreseeable future as CAV have yet to fully penetrate the market, and even then, there will likely be many non-CAV active on the road.

2.4.2) Centralized Models and Other Traffic Rerouting Models

There are also centralized approaches as well, despite their shortcomings, such as [135] which uses a centralized system to manage various intersection managers and will compute incoming vehicles in batches and determine their optimum route. While the batching of vehicles is done to reduce computational burden and decrease the delay, it can still be seen that for a 9-by-9 grid intersection network, the average delay is still rather high, although it is much faster compared to the benchmarked alternatives. This will only get larger as the network is scaled up.

Besides these solutions, there are other studies that make use of existing traffic monitoring tools, such as [136] which makes use of Uppaal Stratego, which is an integrated "tool environments for modelling, validation and verifying real-time systems models as networks of timed automata". Through these suites of tools, along with machine learning, the study has shown a more optimized traffic flow. While this has shown the suitability of the software to provide routing optimization services, the user will eventually find themselves restricted to the software's routing performance and unable to improve upon it as the software is developed in-house by said company, limiting their potential. Meanwhile, [137] has made use of quantum annealing technology developed by D-Wave Systems [138] to determine the most optimal route for the vehicles to take while also providing three possible route choices for each vehicle in order to minimize overlapping road segments between vehicles. However, this model faces a scalability problem, as the existing quantum annealer technology is limited in the size of the logical problem it can compute [138].

Recent advancements in machine learning have also seen the use of deep learning models such as deep reinforcement learning [139] or inverse deep reinforcement learning [140] which makes use of the reward system for the traffic controller to route vehicles in the most efficient manner based on multiple parameters. Some, like [140] even have a certain level of interpretability as they can display the impact the variables have towards the optimum result.

2.4.3) Research Gaps in Traffic Rerouting Studies

Despite the benefits of deep learning models, they can be computationally heavy and may not scale well for a big network, especially a centralized controller which has to handle all the vehicles in the network. In the case of [139], while fast, the model was trained only for a specific destination which makes it impractical in its current state, but it has potential via modern technologies.

Another issue is that the common setup among many of the traffic routing studies is the use of a simplified network graph, designed in a grid-like or similar manner, such as [134], [137], which is not representative of a real urban road network. In short, there is a lack of studies utilising real maps as simulation environments. This is important as many urban roads may not have been ideally planned or were implemented at an earlier time, which resulted in it failing to keep up with the constant growth trend of vehicles on the road over the years.

Additionally, many recent studies on vehicle routing have shifted their routing model's focus on these CAV in preparation for the future, such as [134], [141], [142] as these vehicles give a significant level of flexibility and control. However, a certain level of cooperation or inclusion of non-CAV vehicles should also be included since, as mentioned before, these vehicles will very likely still be active on the road for the foreseeable future, and their presence will influence the effectiveness of these CAV-focused routing models.

Finally, the main thing that seems to have been overlooked is that many routing algorithms look only at travel time or pollution levels when routing vehicles, and they tend to ignore that there are some roads that should be given a higher priority to vehicles that needs it, instead of being a choice simply because it is the road with the shortest travel time. Figure 2.2 below illustrates a simple example.

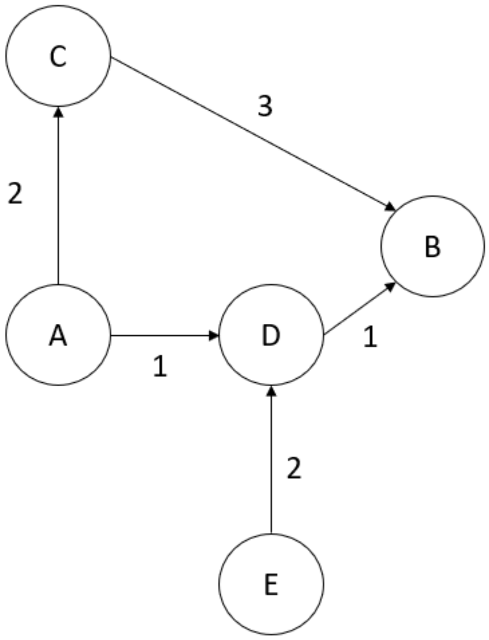


Figure 2.2. Simple example illustrating the problem with simply taking the path with the shortest travel time first.

In the example shown in Figure 2.2, node A is the origin, and node B is the destination. For node A to get to node B, a driver can take either the route A-C-B or A-D-B; however, getting from node E to node B, there is only one route, which is E-D-B. However, due to the focus on the shortest travel time, D would likely be congested before node C is even considered, and drivers getting from node E to node B will have no choice but to get stuck in traffic congestion. Conversely, if drivers coming from node A prioritizes route A-C-B first, then while the individual travel time for drivers from node A may increase, the overall travel time of the road network would decrease, and the overall congestion rate with it. The reduced congestion rate will also benefit future drivers coming from node A as by the time node C is congested enough to warrant using node D, most vehicles from node E would already have made it to their destination, making any duration of congestion that may occur to be shortened as a whole. Of course, taking too long a path would also not be fair to drivers coming from node A, but a certain level of consideration should be taken, nonetheless, which the proposed model has done, and the methodology is explained in the corresponding Chapter 6.2.

TABLE 2.4 summarises the analyses of the state-of-the-art rerouting methods, such as centralized, decentralized, and commercial methods. The analyses also include CAV and reinforcement learning models. Their individual research gaps and common research gaps, which are shared by all the traffic studies, are described in the table. As the objective of the research for this topic is to formulate an efficient traffic routing model in order to reduce travel time and air pollution, it is believed that the proposed model has achieved just that by implementing a road weighing model that includes the number of intersections a road has as one of the criteria, while also simulating the whole traffic activity in an actual urban network map.

TABLE 2.4

Summary of research gaps for traffic rerouting methods

Topics	State-of-the-Art	Research Gaps	Common Research Gaps
Centralized Routing	Has the most complete information, being able to make the most optimum decisions	Faces scalability issues with the number of vehicles and locations. Computational burden makes it unsuitable for real-time applications.	Many studies make use of simplified grid-like networks, which are not properly representative of urban network topology. Route costing methods only look at travel time and pollution or aim to integrate travel information between vehicles. None, as far as the authors are aware, takes into account the importance of a road based on the number of its intersections and how it would impact vehicles whose route does not have any alternatives.
Decentralized Routing	Segmented or Agent-based approach overcomes computational overhead burdens.	Communication between vehicles must be maintained for maximum performance. Non-connected vehicles will cause performance to suffer.	
Routing using Commercial Technologies	Established technologies allow for more flexibility in processing and analysing datasets, allowing for cleaner and more efficient optimization.	Certain technologies are restricted to the creator's prerogative, and performance is tied to the current update. Models developed using these technologies would also be restricted to the technology's limitations until it is addressed.	
Connected and Automated Vehicles	Uses vehicle communication protocols such as the Vehicle Ad-hoc Network (VANET), which allows communication with vehicles and traffic infrastructures. Automated vehicles allow more flexible control of vehicles' trajectory and traffic control, allowing for more traffic routing models to be implemented.	Not all vehicles on the road are connected or automated. In fact, many are not, as CAVs can be expensive or just not available yet. For the foreseeable future, non-CAVs will still be on the road, which would impact or even invalidate proposed fully-CAV routing models.	
Reinforcement Learning Models	Deep learning models allow various parameters to be used to generate a more accurate route suggestion.	The computational burden would be heavy, especially for centralized controllers that are handling the communication with all the vehicles in the network. Existing models have certain limitations in their current state, such as only being trained on one destination [139].	

2.5) Summary

An appropriate analysis must be done in the form of a literature review to justify the problem statements mentioned in Section 1.2. The section above has clearly highlighted the research gaps between existing models and has shown that each of the four problem statements addresses the relevant research gap.

Firstly, to the best of the author's knowledge, there are no attempts at mathematically modelling urban traffic patterns against the time of the day with the input of local traffic parameters alone and without requiring the train a neural network.

Secondly, there are few attempts at a real-time missing data imputation model as it is generally done after data collection, and existing tensor factorization methods suffer from scalability issues.

Thirdly, traffic speed prediction is affected by the heteroscedasticity of data, which could result in erroneous predictions.

Finally, existing traffic rerouting does not take into account the accessibility of the roads in terms of the number of intersections a road has, which would, in turn, affect the efficiency of traffic rerouting.

The following chapters go into more depth into the methods proposed in this thesis.

Chapter 3: Preliminaries

As there are pieces of information that are reused throughout the research, such as the datasets, the traffic data calculations, as well as other jargons and definitions, this chapter is meant to collate all this information in one place in order to reduce repeated mentions and redundancies in the thesis to facilitate a more clear and concise reading experience. Later research will refer to this chapter as needed, especially with regard to the data set used, shown in Section 3.2 below.

3.1) Introduction

The research conducted in this thesis has made use of certain APIs and tools extensively, namely HERE Traffic API, MATLAB, and Simulation of Urban MObility (SUMO). Furthermore, the dataset collected would be used repeatedly throughout the research. The following subsections will explain the data collection method, traffic data calculations, as well as the evaluation criteria used throughout the thesis.

3.2) Data Collection Method

HERE Traffic API[143], an online traffic data API, was used to obtain online traffic information such as jam factor (i.e., traffic congestion level), mean speed, and free flow speed. By using the obtained data, it is possible to plot the traffic activity of an area at a given time. The main area of interest is Bukit Bintang, located in Kuala Lumpur, Malaysia, as shown in Figure 3.1.

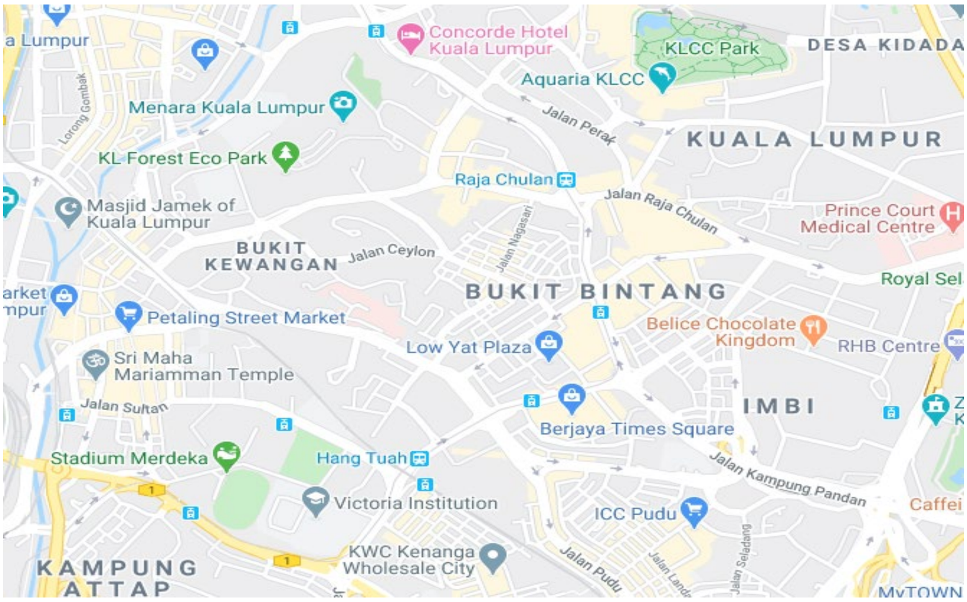


Figure 3.1. Map of Bukit Bintang for the area of interest of the traffic API. The traffic API would return the traffic data for the roads on the map given the appropriate coordinates.

Data from 23 October to 17 December 2019 were recorded at 5-minute intervals, where one day's worth of data is 288 points. Days with erroneous data due to errors in the API calls were removed, leaving 37 days of data intact.

Bukit Bintang is a popular shopping and entertainment district in Kuala Lumpur, in which many popular shopping malls are located, such as Pavilion and Lot 10. Due to this, the traffic activities experienced every day at that location are deemed to be a suitable representation of urban traffic in Malaysia.

Besides Bukit Bintang, two other locations were chosen: Bandar Sunway and Damansara Utama, both urban and suburban cities in Malaysia. Sunway represents an urban area with a focus on educational facilities such as two universities (Monash University Malaysia and Sunway University) along with a popular mall called Sunway Pyramid. Damansara Utama focuses on an urban area with more office blocks and shops, along with a residential condominium in its vicinity.

Note that the traffic data were recorded in 2019, prior to the Movement Control Order (MCO) measures implemented in 2020-2021 by the Malaysia Federal Government, and hence would reflect the natural traffic patterns for the urban locations mentioned above. As MCO was in place during those times, it was impossible to gather new traffic data that would skew the existing dataset and not represent a normal traffic pattern that would re-emerge after the MCO was lifted.

3.3) Traffic Data Calculations

Although the Traffic API returns the traffic information for all roads in a grid based on the Mercator Projection [144], the calculations used in this thesis utilise the mean of all the roads' traffic information. The relevant calculations for the traffic data are shown below:

3.3.1) Traffic Speed

In order to have a better grasp on the traffic pattern of the area, the traffic speed data of the roads were aggregated and calculated as shown in Equation (1) below:

$$V_{Avg} = \frac{1}{N}\sum_{n=1}^{n=N} V_n \qquad (3.1)$$

Where N is the number of road segments returned by the API, and V is the traffic speed data. The averaged traffic data is used as a simplification as there are many low-frequency roads that are too sparse in data due to the limited traffic data collected. Hence, the averaged traffic is used as it is a representation of the overall traffic pattern of the region.

3.3.2) Traffic Jam Factor

[145] has explained that the Traffic Jam Factor returned by the HERE Traffic API describes the congestion of the road using a value in the range of 0 to 10, unlike that of the Greenshields' traffic model, which describes traffic density, k as a value between 0 and 1. By assuming a linear relationship, Equation 3.2 below is derived:

$$JF = \frac{k}{kj} * 10 \qquad (3.2)$$

Using the time when the API is called as the time stamp, it is possible to plot a graph of jam factor (JF), which represents the congestion rate of the traffic along the road, to time, which describes the traffic activity of the area. This is shown in Figure 3.2.

For example, the jam factor for the roads used to plot the graph in Figure 3.2 is the average of the jam factor of every road. Mathematically speaking, for N number of roads, the average JF is shown in equation 3.3 below:

$$JF = \frac{1}{N}\sum_{n=1}^{n=N} JF_n \tag{3.3}$$

Based on Figure 3.2, it can be seen that the general traffic trend for the three urban and suburban areas in Malaysia, namely, Bukit Bintang, Bandar Sunway, and Damansara Utama is essentially the same, with traffic picking up in the early morning, increasing further in the evening, and eventually reaching a peak around night time before subsiding as the day comes to a close. A traffic study in the city of Winnipeg [146] also shows a similar traffic pattern. Although the traffic patterns may show some variations, the general pattern is ultimately the same, which means that a generalised mathematical model is feasible.

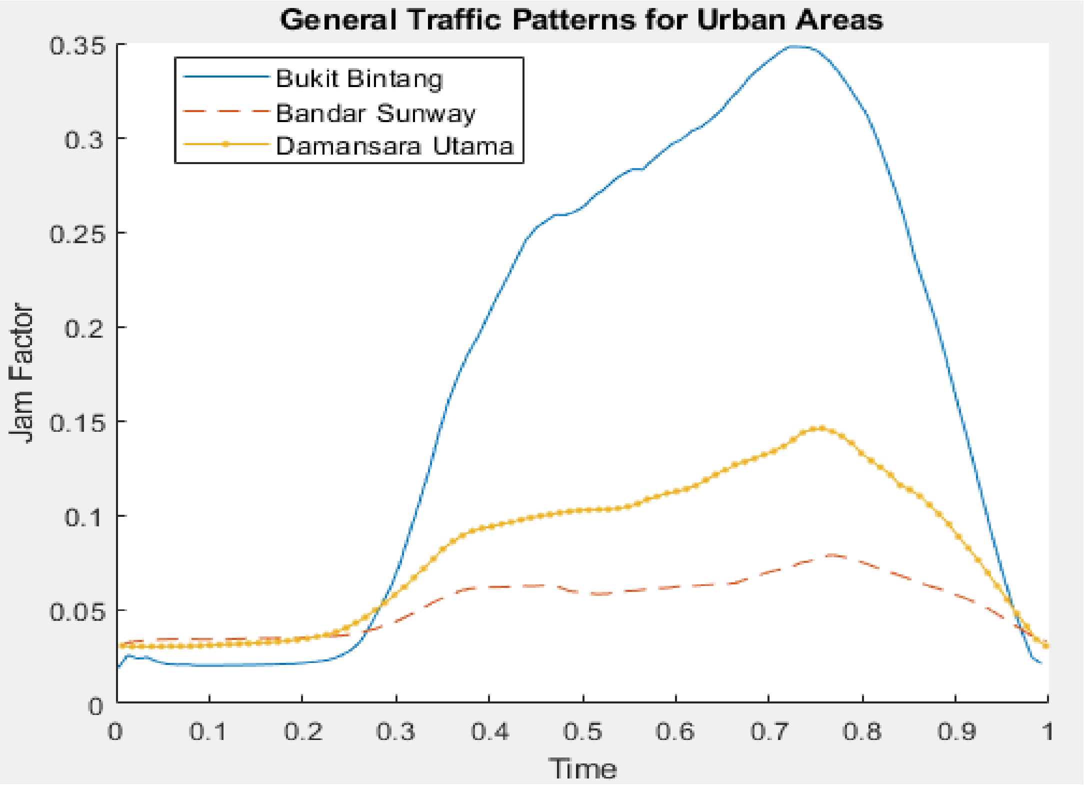


Figure 3.2. Plot of General Traffic Congestion for Bukit Bintang, Bandar Sunway, and Damansara Utama, showing the typical traffic activities of the areas obtained from the traffic API

The research conducted in this thesis aims to use an urban area's general traffic trend to provide a more high-level and robust view of urban traffic in Malaysia. Hence, it makes use of the aggregated traffic patterns instead of the individual roads in each urban area. While that may help in making certain simulations more accurate, it would result in an overly specialised model and would not apply to other areas in the map unless more data features were included, which the research conducted in this thesis does not have at the time of conducting the simulations. Hence, it is of note that the research conducted in this thesis all work on the assumption that minimal traffic data features are available, as that is true for many developing countries such as Malaysia.

3.4) Evaluation Criteria

There are a few commonly used evaluation criteria used in evaluating the performance of the proposed models in this research. These evaluation criteria that were used in multiple objectives are

Root-Mean-Squared-Error (RMSE) and Mean Absolute Percentage Error (MAPE) are the two main error calculations. The formulas are listed in Equation (3.3) and Equation (3.4) below:

$$RMSE = \sqrt{\frac{1}{D_{pred}}\sum_{d=1}^{D_{pred}}\left(x_d - \widehat{x_{pred\_d}}\right)^2} \tag{3.3}$$

$$MAPE = \frac{1}{D_{pred}}\sum_{d=1}^{D_{pred}}\left|\frac{x_d - \widehat{x_{pred\_d}}}{x_d}\right| * 100\% \tag{3.4}$$

Where D_{pred} represents the total number of data points that had been worked on, whether it be missing data imputation or traffic data prediction.

RMSE gives a relatively high weight to large errors as the errors are squared before they are averaged. This means the RMSE would show if the performing method has a more frequent occurrence of larger errors compared to other methods or scenarios (e.g., Different rates of missing data).

Contrarily, MAPE displays non-biased relative effectiveness of a particular method and scenario on average. Note that for both RMSE and MAPE, the lower the value, the better the model's performance.

3.5) Chapter Summary – Preliminaries

In the interest of conciseness, this chapter aimed to collate the general information used throughout the research explained in the rest of the thesis. The dataset used and the related traffic data collection method, notable traffic parameters and their calculations and derivations, as well as the common performance metrics used, were all explained in this chapter in order to remove the need to repeat the explanation in the following chapters and avoid repetitiveness.

Chapter 4: Traffic Data Modelling and Missing Data Imputation

Traffic data analyses have been conducted many times in the past, bringing forth traffic theorems such as Greenshields' Model [52] and other car-following models. These models aimed to deduce the pattern of traffic given certain traffic parameters and have contributed greatly to the field of traffic engineering. In recent years however, the focus has shifted more towards utilising the computational power afforded by modern machines to conduct machine learning in order to obtain the final results, which, while efficient, has taken out some of the traffic analysis portions of the study. This chapter looks into the importance of traffic analysis and traffic modelling and proposes a mathematical model that generalises urban traffic patterns, named MOD3D-PAT. During the course of the research, the proposed model has been published in [2].

Besides that, it also investigates another important aspect of traffic studies, which has seen a popular number of studies, which is missing traffic data imputation. Traffic data can be said to be the lifeblood that all other traffic-related studies rely upon, as without traffic data, nothing can be referenced or studied in the first place. Additionally, erroneous data will negatively affect the related studies and their corresponding model as well, leading to inaccurate or unreliable results. The importance of data, in general, cannot be stressed enough, and this, of course, makes missing data imputation a very important aspect in ensuring that traffic studies or the ITS itself can be executed without worries. ITS, in particular, requires a missing data imputation model that can match the real-time aspect of the system, and this chapter introduces the ensemble model proposed in this research that makes use of the real-time computational capabilities of selected machine learning models and leverages the strengths of computationally heavier models for offline training, while also introducing a novel clustering and quantile regression mechanism to overcome another problematic nature of traffic data, which is Heteroscedasticity. [2]

4.1) Introduction

Concerns around traffic congestion have been growing in recent years. This growing concern has led to an increase in studies related to the improvement of existing traffic systems. Intelligent Transportation Systems (ITS) is a traffic system that utilises these studies to maximize the efficiency of traffic on the road by using various state-of-the-art statistical and machine learning techniques. These studies mainly include traffic management ([3], [4]), traffic forecasting [6], [57], [147], and vehicle rerouting [148]. In an effort to further improve the accuracies of the studies, various machine learning and mathematical models have been proposed to understand further the mechanics that affect traffic behaviour.

In the early days of traffic studies, there were many studies that researched traffic behaviour and tried to model these traffic patterns in a mathematical manner, such as the Greenshield, Pipes, or Van Aerde models [149] to name a few. Needless to say that being able to describe traffic behaviour mathematically gave future researchers a solid manner in which to simulate and test their various proposed traffic models. Even now, traffic simulators would use various car-following models in order to achieve a level of realism in the way their simulators emulate vehicle behaviour.

[2] Part of the content of this chapter appear as R. K. C. Chan, J. M. Y. Lim, and R. Parthiban, 'MOD3D-PAT - A novel modified 3rd degree polynomial approximation for modelling traffic congestion in urban areas," International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2021, no. October, pp. 7–8, 2021, doi: 10.1109/ICECCME52200.2021.9591081.' and 'R. K. C. Chan, J. M. Y. Lim, and R. Parthiban, "Ensemble ARIMA-QRFC with Tensor Factorization Training for Real-Time Missing Data Imputation for Traffic Speed"' which has been submitted to Future Generation Computer Systems for review

In recent years, although there have been studies done in order to analyse traffic congestion in certain areas, such as [12], there are none that propose a mathematical model to approximate the related traffic behaviour to the best of the author's knowledge. The model proposed in this research, MOD3D-PAT, would give further insight as to what parameters may affect the traffic behaviour and how the traffic would turn out.

It is believed that having a mathematical model to represent the traffic congestion behaviour would turn out to be a useful tool in assisting further studies in understanding traffic behaviour and, ultimately, overcoming traffic congestion.

Furthermore, by using easily accessible online data via traffic APIs, it is hoped that further studies could be done without worries of having to invest in too many resources. However, this also brings up a separate problem that traffic studies, as well as many other data-related research, face, missing data.

The occurrence of missing data is seen to be a disruption to various data analysis attempts, where mishandling of it would result in a lack of reliable data and incorrect analysis and prediction.

Hence, missing data imputation has been a constant field of study in various fields of data analytics due to the challenge it poses. This is true for various fields, including ITS. In the field of ITS, however, real-time data imputation is especially important as traffic management is done in real-time, and a slow data imputation would end up redundant. Although there are missing data imputation methods that impute missing data after the collection is done, such as the ones in [14], [15], there are fewer studies that predict and impute missing traffic data in real-time, and those that do, may suffer from high computation times.

In the field of missing data, tensor factorization has seen great success, and missing traffic data is no exception. There are also neural network models which are popular in recent years, such as Generative Adversarial Networks (GAN) and Graph Convolutional Networks (GCN) that have been introduced fairly recently, with GAN being proposed in 2014 [95] and GCN in 2017 [109] respectively. This means that these two methods still have the potential for further improvements. However, these models can be computationally expensive and would not be ideal for real-time applications without significant training data.

While the simpler Ordinary Least Squares (OLS) regression could also be used to impute traffic quickly, these methods are more heavily affected by the presence of outliers and would require data pre-processing to detect and remove outliers, which in turn could cause further inaccuracy due to false positives. Instead, an inherently robust data imputation method against outliers should be developed, removing the need for outlier detection and removal.

Another issue faced is the outlier data problem and the heteroscedasticity of traffic data. Outliers can affect the accuracy of trained models, as shown by [150], and heteroscedasticity is also acknowledged as a problem and has research aimed at resolving the issue [151]. Besides these two, the ITS environment tends to require live updates to provide real-time support, which means that missing traffic data imputation should operate well enough in a real-time setting.

Traffic modelling and analyses, together with missing traffic data imputation, are important knowledge and pre-processing steps that can be used to further improve the predictive and routing capabilities of an ITS, making them important background steps to be taken before implementing the other functions. With that in mind, this research has proposed a novel Modified 3rd Degree Polynomial Approximation for Modelling Traffic Congestion in Urban Areas, MOD3D-PAT, which is a generalised mathematical model for urban traffic patterns, coupled with additional traffic analysis on the types of traffic patterns observed in local urban areas in Malaysia to grasp Malaysian drivers behaviours better. Besides that, a missing traffic data imputation framework was also proposed, aiming to resolve the issue of real-time imputation by using ARIMA and Quantile Regression Forest (QRF) for imputation, combining the QRF with Clustering (QRFC) to resolve the outlier and heteroscedasticity problem, and tensor factorization for imputation enhancement via training the overall model. The proposed model missing traffic data imputation model was named **A**RIMA-**Q**RFC with **T**ensor Factorization (AQT). This model provides a data imputation system that is robust to varying levels of missing data and is suitable for real-time applications.

To summarise, this chapter of the research covers two of the main objectives, which are also the backend functions upon which traffic predictions and routing are built, namely:

i. To investigate and model Malaysia's developing cities' travel behaviours using online network information.

ii. Design an accurate real-time missing data imputation method utilising machine learning.

4.2) Derivation Of Proposed Mathematical Model - MOD3D-PAT

This section goes over the derivation process of the mathematical model for approximating traffic congestion throughout the day.

Using the available online traffic information, it is possible for a mathematical model to be derived in order to fit the resulting traffic patterns shown in datasets like Figure 3.2. Through the use of a 3rd degree polynomial as the base, a MOdified 3rd Degree Polynomial Approximation for Traffic, or MOD3D-PAT for short, is derived.

Unlike conventional traffic models, this mathematical approximation allows quantitative analyses for an area based on the maximum speed/speed limit, the time when traffic begins to rise, the average maximum speed, etc.

Regarding the obtained traffic data, the jam factor explained in Section 3.3 is used in the derivations in the subsections below. Time-related parameters are represented using a range of 0 to 1, where 0 represents 12:00 am for the current day, and 1 represents 12:00 am for the next day. These traffic data are part of the parameters that will be used when trying to derive some of the coefficients in the following derivations.

It is also important to note that this model is done for an area or a cluster of roads in an area, giving insight into the overall traffic of the area itself. Another point to note is that since this model provides a generalized traffic model, it is not tuned to any specific day. Hence, there is no need to specify what kind of day it is, such as weekdays, weekends, holidays, events, etc. Instead, this model is controlled based on the parameters mentioned previously, such as the time when traffic begins to rise, and the resulting model will change according to those. These parameters themselves could be linked to the type of day, but it does not directly affect the model.

4.2.1) Overall Proposed Derived Equation

A 3rd degree polynomial, or cubic function, was chosen to act as the base of the mathematical approximation for jam factor (JF) against the time of the day where the value [0,1) represents [0000, 2400) hours. Note that a cubic function has the standard form:

$$\alpha x^3 + \beta x^2 + \gamma x + \delta \tag{4.1}$$

Where α, β, γ, and δ are the coefficients of each power of x in decreasing order. Based on the polynomial regression done on the JF plots, it is found that the cubic function used has the form:

$$-\alpha x^3 + \beta x^2 - \gamma x + \delta \tag{4.2}$$

As shown in Figure 3.2, the jam factor for the early part of the day is more or less constant. Based on this understanding, the cubic function is modified by multiplying its higher powers with a sigmoid activation function to have it act in a similar manner during the early part of the day.

The sigmoid function used is of the form:

$$S(x, j, k) = \frac{1}{1 + e^{-k(x-j)}} \tag{4.3}$$

Where j is the point x at which $S(x, j, k) = 0.5$, and k is the slope of the sigmoid function transitioning from 0 to 1, where a larger magnitude of k means a steeper slope and a negative k flips the sigmoid function.

By multiplying the sigmoid function with the higher powers of the cubic function, the resulting equation would be:

$$S(x, Ts, 10) * (-\alpha x^3 + \beta x^2 - \gamma x) + \delta \tag{4.4}$$

Where Ts represents the approximate time when the traffic begins to increase, and by referencing Figure 3.2, its value would be around 0.2 or 4:45 am. The value of Ts is generally around 0.15 to 0.25, or around 3:30 am to 6 am. Values outside this range would be classified as an outlier and may not be compatible with the proposed model.

4.2.2) Verification of Model and Proposed Values for Coefficients

This section covers the proposed derivation of the coefficients. These coefficients are first found using polynomial regression, and a mathematical solution is then derived and verified to provide further clarity as to what affects the JF.

(A) Polynomial Regression of Data as Reference

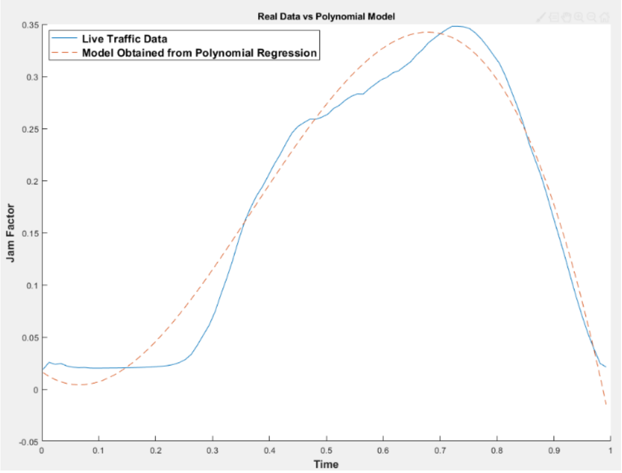


Figure 4.1. Comparison between the real data and a polynomial model utilising polynomial regression.

In order to obtain a reference to what the coefficients of the model should correspond to, polynomial regression is first conducted on the dataset, as shown in Figure 4.1.

In order to obtain a close enough model, the tests have been conducted using the values of the coefficients returned by the polynomial regression in mind, which acts as a form of verification towards the proposed values of the coefficients mentioned in the later subsections.

(B) Proposed Adjusted Peak Jam Factor, α

α is found to be related to the average peak jam factor, which is the jam factor during the busy period of the day. It is found that α is equal to the average peak jam factor multiply 10.

The time when the peak jam factor starts is determined by the mean of the ratio of the average speed on the road and the maximum speed on the road, specifically:

$$t_{peak\_start} = \frac{V_{Avg\_D_{day}}}{Vf_{Avg\_D_{day}}} - 0.5 \tag{4.5}$$

Where $VD_{Avg} = \frac{1}{D_{day}} \sum_{d=1}^{d=N} V_{Avg\_d_{day}}$ and $VfD_{Avg} = \frac{1}{D_{day}} \sum_{d=1}^{d=N} Vf_{Avg\_d_{day}}$ and D_{day} represents the number of data points in one day's worth of traffic dataset. As the traffic data is collected in 5-minute intervals, $D_{day} = 288$ for this particular study. Each parameter of $v_{Avg\_d_{day}}$ and $vf_{Avg\_d_{day}}$ represents the average speed and maximum speed of all the roads for each data point. Regarding Vf_{Avg}, which used the same calculation as V_{Avg} shown in equation 3.1, note that while Vf for each road tends to equal the indicated speed limit, this research looks at the maximum speed of the road instead, which means that it takes into account speed readings which breaks the indicated speed limit, leading to times where $Vf = v \; when \; v > V_{speedLimit}$.

The average peak jam factor, JF_{peak} is then found by first noting the JF at t_{peak_start} and taking the average across the values after that until the JF falls to a similar value again, an indication of when the peak hour is ending.

α is also less sensitive to the average peak jam factor at lower JF values. At lower JF values, such as around 0.1, the value of α is less than expected. In order to simulate this behaviour, JF_{min} is introduced. JF_{min} represents the average JF from the start of the day until Ts. By taking the difference between JF_{peak} and JF_{min} and then dividing the result by JF_{min}, α is able to scale well for both low and high values of JF.

Hence, the resulting equation defining the adjusted peak jam factor, α would be:

$$\alpha = JF_{peak} * 10 \; * (\frac{JF_{peak} - JF_{min}}{\mathrm{JF}_{min}}) \tag{4.6}$$

Note that $JF_{peak} * 10$ is used at the input to the sigmoid function to ensure a greater change in the sigmoid function, as too small an input value would then require changes to the slope instead.

(C) Proposed Traffic Start Ratio

It is determined that the ratio of adjusted peak jam factor, a to the coefficient β corresponds to Ts in accordance with some function f:

$$\frac{\beta}{\alpha} = f(Ts) \tag{4.7}$$

The values of Ts is found by observing the traffic behaviours from the graph, and the traffic start ratio is taken from the resulting polynomial regression for the overall equation $S(x, Ts, 10) * (-\alpha x^3 + \beta x^2 - \gamma x) + \delta$ and taking the ratio $\frac{\beta}{\alpha}$.

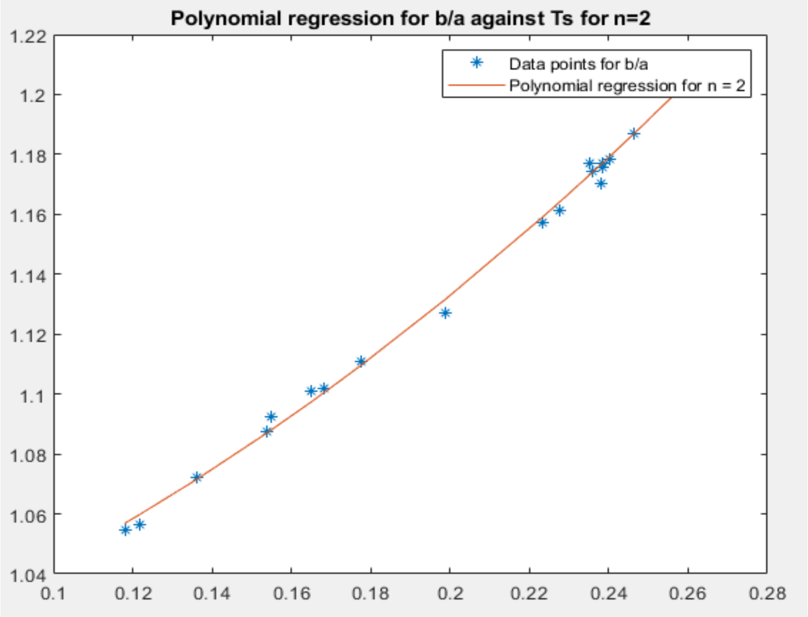


Figure 4.2. Linear regression of Ts against the traffic start ratio, β/α

By plotting the value of Ts to the traffic start ratio, $\frac{\beta}{\alpha}$ and conducting a polynomial regression of order $n = 2$ as shown in Figure 4.2, it is determined that the relationship between the two can be approximated using the following function:

$$f(Ts) = 1.8571x^2 + 0.3367x + 0.9913 \tag{4.8}$$

The coefficient β can be determined by using this function. A higher traffic start ratio would result in a higher rate of increase of JF and maximum JF for the polynomial approximation, as β would be greater. Based on Figure 4.2, it can be seen that the later the traffic starts to increase, the greater the rate of increase of traffic, and this would, in turn, affect the maximum JF for the day as well.

(D) Proposed Traffic Start Difference, γ

Similar to the Traffic Start Ratio, the Traffic Start Difference is the difference between the coefficients b and a, namely:

$$\gamma = \beta - \alpha \tag{4.9}$$

This coefficient is important in that it ensures that the model remains positive throughout the duration of the day for a reasonable value of Ts.

(E) Proposed Average Low Traffic, δ

The coefficient δ represents the early part of the traffic, which means δ is the mean of the JF before Ts. In other words:

$$\delta = \frac{(JF_1 + JF_2 + JF_3 + \cdots JF_{d\_ts})}{d\_ts} \tag{4.10}$$

Where d_ts represents the number of data points taken for the pre-Ts period, of which each data point represents a snapshot of the traffic situation at that time.

As the early traffic tends to be constant, it is not affected by a range of values like α is and, as such, does not require adjustments such as using the sigmoid function.

4.3) Results and Analysis - MOD3D-PAT

This section displays the results of the accuracy of the proposed MOD3D-PAT in modelling traffic congestion for the day.

4.3.1) Validation of Proposed Derived Model MOD3D-PAT

The main area chosen is Bukit Bintang, Malaysia, as it is one of the main urban areas in Malaysia and hence, is suitable for testing.

Figure 4.3 displays the results of the final derived mathematical model for one of the data sets for Bukit Bintang.

Following a similar pattern, TABLE 4.1 below shows the average R-squared (Goodness of fit) of the model and the Root-Mean-Squared-Error (RMSE) of the mathematical models across different datasets representing different days.

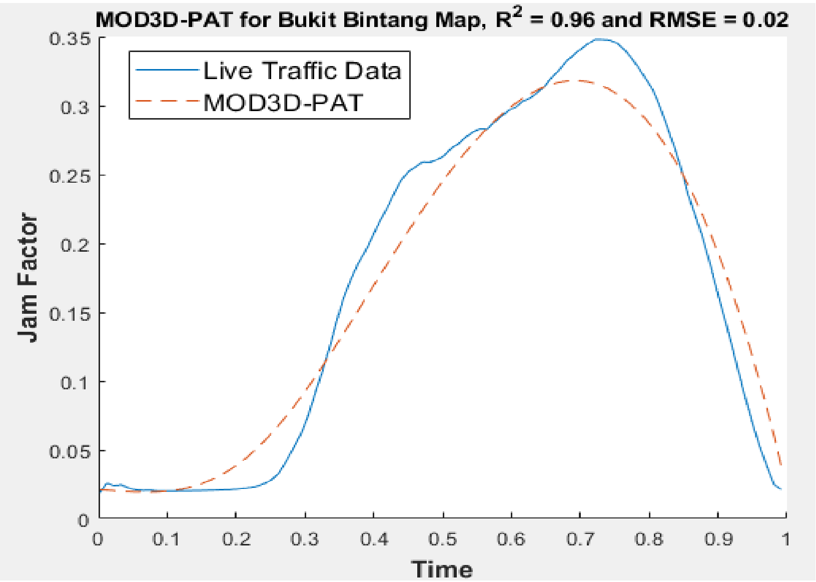


Figure 4.3. Plot comparing the accuracy of the proposed model compared to the original live traffic data obtained via the traffic API.

As shown in Figure 4.3 and TABLE 4.1, the model represents a good fit for all the datasets, with the least good fit having an R-squared value of 0.93 and an RMSE of 0.035 and the best results having an R-Squared value of 0.97 and RMSE of 0.024. These two results do not differ much, proving that the model is consistent and works well enough as an approximate and that it is possible to analyse traffic patterns using online traffic data. This can also be seen from the averaged R-squared value of 0.9543 and RMSE of 0.0279.

4.3.2) Robustness of MOD3D-PAT Towards Other Maps

In order to ascertain the reliability of the model, two other places are selected: Bandar Sunway and Damansara Utama, which are urban and suburban areas, respectively, both in Malaysia. This section covers the performance of the proposed derived model, MOD3D-PAT, for these two areas.

(A) Bandar Sunway

From Figure 3.2, it can be seen that the traffic patterns in Sunway tend to experience a sharp rise in traffic during the late afternoon or early evenings, at around 4:30 pm onwards.

This is likely due to Bandar Sunway having two university Campuses: Sunway University and Monash University Malaysia. This results in a larger number of vehicles on the road during the evening rush hour due to both the students as well as employees going home as opposed to Bukit Bintang and Damansara Utama, where there are mostly only people from the workforce.

TABLE 4.1

VERIFICATION RESULTS OF THE MODEL COMPARED TO THE ACTUAL DATA

Dataset	R-Squared	RMSE
1	0.96	0.026
2	0.96	0.027
3	0.96	0.035
4	0.94	0.030
5	0.93	0.025
6	0.97	0.024
7	0.96	0.028
Average	**0.9543**	**0.0279**

As the proposed model is aimed to be a generalisation of traffic behaviour, it aims to provide a more general traffic pattern rather than follow a specific trend. Instead of following the sharp increase, the model provides a more gradual increase and reaches its peak at an earlier time, resulting in a more averaged-out result. This can be seen in Figure 4.4 below.

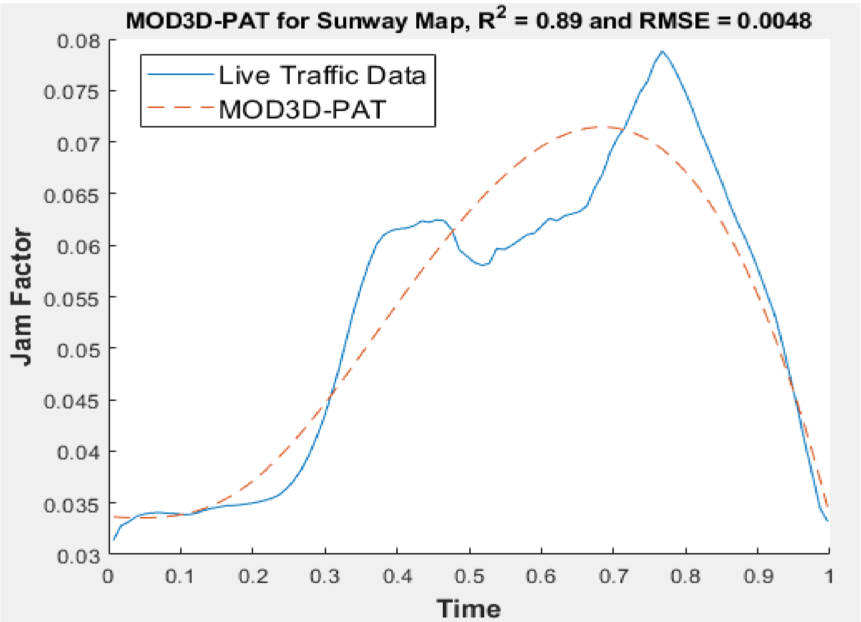


Figure 4.4. Traffic Model Comparison between the original data and the data produced by the proposed model for Bandar Sunway

(B) Damansara Utama

Damansara Utama shows a trend that is between Bukit Bintang and Bandar Sunway, displaying a sharper increase than Bukit Bintang but more gradual than Bandar Sunway. This is likely due to the fewer roads in the area compared to the other urban areas, as Damansara Utama is considered a suburb, meaning drivers do not have many choices in their route selection.

Taking this situation into consideration, it is expected that Figure 4.5 would show similar behaviour, albeit with a better R-squared and RMSE. It can also be seen that, similar to the other locations, it is possible for the proposed MOD3D-PAT to provide a reliable 3rd degree polynomial approximation for the traffic no matter the location.

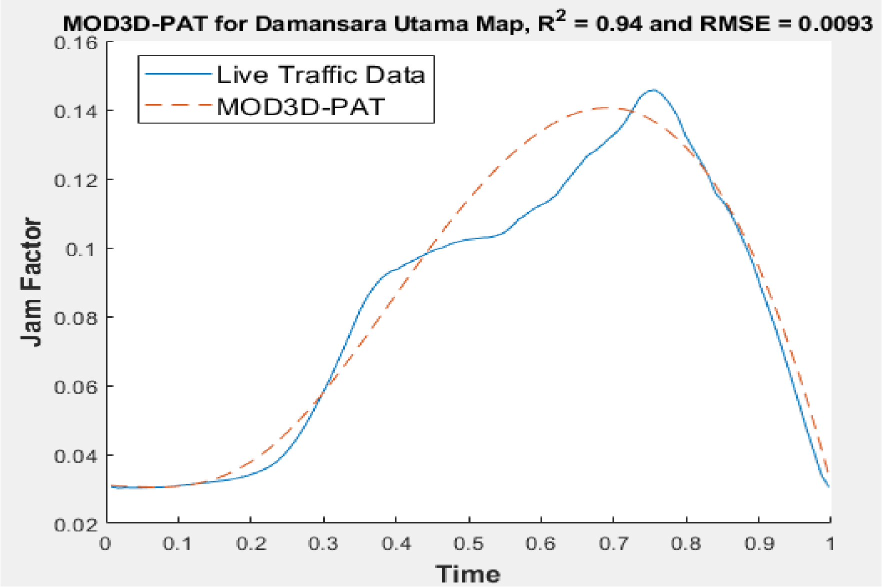


Figure 4.5. Traffic Model Comparison between the original data and the data produced by the proposed model for Damansara Utama

4.4) Methodologies - ARIMA-QRFC with Tensor Factorization (AQT)

This section details the proposed missing data imputation method, ARIMA-QRFC with Tensor Factorization (AQT). The proposed model utilizes the computation speed of ARIMA for low rates of missing data and relies on a quantile regression forest that is robust towards outliers for higher rates of missing data. Tensor factorization is used for training, which would resolve its longer computation time as the model would be further trained separately from the actual data imputation. To further enhance the model's capabilities, a novel k-means traffic clustering method and quantile regression forest are proposed to optimize the data set used for quantile regression. The clustering algorithm clusters the traffic data into smaller clusters of similar patterns, improving the accuracy as well as the computational efficiency of the model.

The following subsection explains the framework of the proposed AQT model. It covers the overall algorithm, as summarized in Figure 4.6, as well as the proposed Quantile Forest Regression Clustering

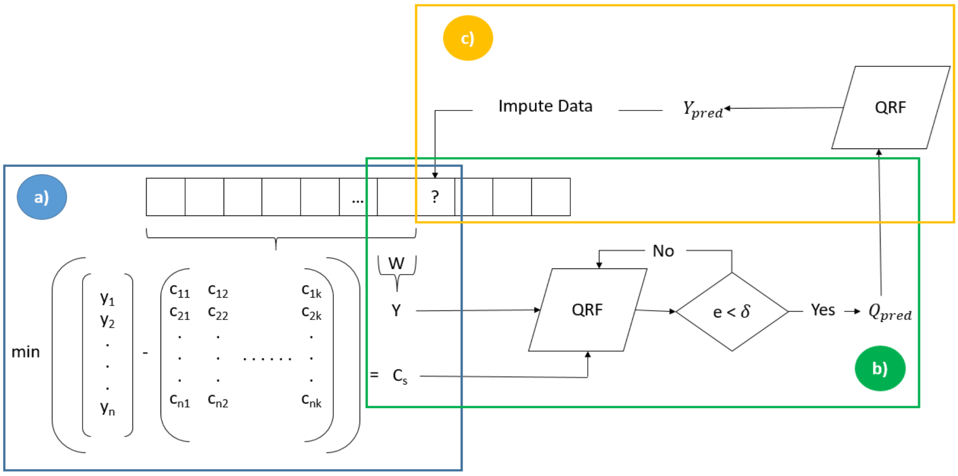


Figure 4.6. Overall process Flow of Proposed Model, (a) Represents the cluster selection, (b) represents estimation of the quantile, and (c) represents the data imputation process

algorithm. The methods for determining the number of clusters for the K-means clustering are also covered.

4.4.1) ARIMA Set-Up

ARIMA requires three parameters p, d, and q, which represent its three models, which are Auto-regression, Integration, and Moving Average, to be used effectively.

The Augmented Dickey Fuller Test (ADFT) [152] was used to test for stationarity of the traffic data at different levels of p. Meanwhile, MATLAB's autocorrelation function, autocorr, and partial correlation function, parcorr, were used to determine the values of d and q.

The above tests determined that an ARIMA system with parameters (1, 2, 6) would be the most suitable.

4.4.2) Algorithms

The following subsections explain the various algorithms used by the proposed method, namely the training of the Quantile Regression Forest Clusters (QRFC), the imputation algorithm for the QRFC, as well as the overall algorithm for the simulation conducted for the proposed ensemble method, **A**RIMA-**Q**RFC with **T**ensor Factorization, or AQT.

(A) QRFC Algorithm

The proposed QRFC algorithm consists of 2 parts: i) Generation and ii) Run-time or imputation. Generation refers to creating traffic data clusters via k-means clustering to group traffic data of similar patterns and thus create a random forest for each cluster.

Run-time or imputation refers to the steps taken during data imputation when missing data is detected. The Euclidean distance of the traffic data recorded for the day is taken, and the traffic pattern is compared to the clusters created during the generation phase. The forest of the cluster closest to the current traffic pattern is selected, and the last W observed data is extracted to estimate the quantile to predict, Q_{pred} using quantile regression forest (QRF), where W represents the window size of the observed data to be extracted.

The bisection method is used to estimate Q_{pred}. This is done by selecting the median and comparing the value obtained using QRF with the observed data. Once the error is below the set threshold, that quantile is used as $Q_{pred,}$ and QRF is once again used to obtain the traffic data for the detected missing data.

The following subsections cover the generation of the clustered forest, the imputation algorithm, as well as the selection of the window size, W, used for the simulation.

(I) Generation of Clusters and Forest Algorithm

The traffic training data are used to obtain the K number of clusters using K-means clustering. Figure 4.7 shows the clustering mechanism whereby the collection of traffic data is split into K clusters.

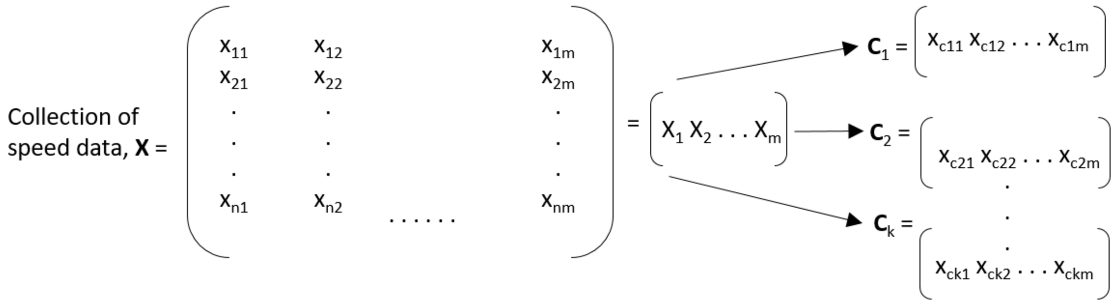

Collection of speed data, X =
x_{11} x_{12} x_{1m}
x_{21} x_{22} x_{2m}
x_{n1} x_{n2} x_{nm}
= $X_1 X_2 ... X_m$
C_1 = $X_{c11} X_{c12} ... X_{c1m}$
C_2 = $X_{c21} X_{c22} ... X_{c2m}$
C_k = $X_{ck1} X_{ck2} ... X_{ckm}$

Figure 4.7. Clustering of Traffic Data. Traffic Data is split up into groups consisting of traffic data with the same cluster labels.

After each traffic data is labelled with its corresponding cluster, Quantile Regression Forest is then used to obtain a model for each cluster by training the traffic data with the same label. This is shown in Algorithm 4.1:

Algorithm 4.1 Clustering And Forest Generation Algorithm

```
Inputs:
    Training Traffic Data, X;
    Number of Clusters, K;
Outputs:
    Cluster Models Collection, C;
Conduct K-means clustering on training data X for K number of clusters
For k = 1:K
Obtain daily traffic data with label k from K-means clustering
Use the TreeBagger function to create the ensemble of decision trees for cluster k.
Save the corresponding 'Forest' cluster model in the cluster models collection, C
end
```

(II) Cluster Selection

When missing data is detected, the past observable traffic data for the day is compared to each cluster's mean traffic data via the Euclidean Distance. The cluster that is 'closest' to the observed traffic data is then chosen as the cluster to perform Quantile Forest Regression (QRF) on, as shown in Equation (4.11). In the process flow diagram shown in block (a) of Figure 4.6.

$$Selected\ Cluster, C_s = (X - C_n) \qquad (4.11)$$

(III) Estimating the Quantile

Quantile regression returns the expected value at a designated quantile. Hence, to increase the likelihood of an accurate prediction, the quantile to predict, Q_{pred} needs to be estimated. To achieve this, the bisection method is used [153]. Among other methods, the bisection method is the simplest to use as it does not require a predetermined equation in order to estimate the optimum value.

The most recent observable data of size W is taken and compared with the value returned for those time steps by using the QRF on the selected cluster. The error, e is calculated as shown in Equation (4.12).

$$e = mean(x - \hat{x}) \qquad (4.12)$$

Where x is the observed data and $\hat{x}$ is the value returned from the QRF algorithm. If the error is positive, it means that Q_{pred} is higher than the previous guess and lower if it is negative. This is repeated until e is less than the error threshold set prior to the simulation. Block (b) in the process flow diagram in Figure 4.6 visualizes this part of the algorithm.

(IV) Determining the Window Size

The QRF Algorithm uses the last W observations to estimate the expected quantile for the bisection method. As such, simulations were conducted for a window size of 1 to 5 to determine what the optimum size is. Too big a window size was deemed not to be suitable as the traffic patterns would be largely different due to the time difference, i.e., traffic patterns in the evening would differ from traffic patterns in the morning.

TABLE 4.2 and TABLE 4.3 shows the result taken from the simulation for each window size and missing rate.

TABLE 4.2

RMSE of Simulation for Different Window Sizes

	RMSE				
Missing Rate (%)	Window Size				
	1	2	3	4	5
0.1	**0.3983**	0.3988	0.3988	0.3988	0.3988
0.3	**0.473**	0.473	0.473	0.473	0.473
0.5	**0.5306**	0.5315	0.5315	0.5315	0.5313
0.7	**0.6617**	0.6771	0.7084	0.7285	0.764
0.9	**0.837**	0.8789	0.9752	1.0108	1.0491

TABLE 4.3

MAPE of Simulation for Different Window Sizes

	MAPE				
Missing Rate (%)	Window Size				
	1	2	3	4	5
0.1	**0.8263**	0.8263	0.8263	0.8263	0.8263
0.3	**0.9476**	0.9476	0.9476	0.9476	0.9476
0.5	**1.0711**	1.0735	1.0736	1.0736	1.0735
0.7	**1.365**	1.4115	1.4651	1.5022	1.5848
0.9	**1.7319**	1.7995	1.9818	2.0779	2.1494

As seen from the tables, a window size of 1 is equal to or better than the other window sizes. Along with the resulting shorter computation time, a window size of 1 was chosen for the simulation.

(V) Overall Imputation Algorithm

Algorithm 4.2 shows the summarized steps taken for the QRFC imputation. The process flow for a single imputation is also visualized and shown in Figure 4.6, where C_s is the selected cluster, ∂ is the error threshold, and x_{pred} is the value that is returned by the proposed model for data imputation. Block (a) and (b) represent the process of cluster selection and estimating the quantile, respectively, whereas block (c) is the data imputation itself by taking Q_{pred} and conducting the QRF algorithm to obtain the data to impute, x_{pred}.

Algorithm 4.2 QRFC Imputation Algorithm

```
Inputs:
    Time step, t
    Observed Road data, X
    Window Size, W;
    QRF Clusters, C;
Outputs:
    Imputed Data, $x_{pred}$;
1:  Calculate Euclidean distance between observed data, X with QRF Clusters, C
2:  Select the closest cluster, c
3:  Use bisection method to obtain expected quantile to predict, $Q_{pred}$ using X from time step t-1 to t-W until less than error threshold.
4:  Obtain $x_{pred}$ using MATLAB's forecast function using $Q_{pred}$ and QRF cluster, c
```

(B) Real-Time Simulation of AQT Model

As the algorithm was designed to work in real-time, the simulation was designed in such a way whereby traffic data is obtained sequentially. The observed traffic data is then recorded into a list of previously observed data for the day if the data is present and is imputed if it is missing.

In the event of the obtained value being missing data, the missing rate is calculated and will utilise the appropriate imputation method depending on the apriori switching threshold.

In the case of QRFC, the previously observed traffic data are used to compare with the reference clusters obtained during the training phase by comparing their Euclidean distance, and the closest matching cluster is selected.

The last W number of observed data is used to estimate the expected quantile to predict using the QRF algorithm. In order to estimate the quantile, the bisection method is employed and compares the value produced by the predicted quantile, Q_{pred}, to the observed W values.

In the event that the ARIMA method is chosen, the previously imputed daily data is used instead of just using the observable values and the data is imputed.

Finally, at the end of each day, the tensor factorization method is used (The referenced BATF method [15] is used in this thesis) to re-impute the whole day's data for training.

The following subsections show the overall algorithm as well as analyses done to determine the optimum switching threshold.

(I) Proposed AQT Algorithm

The overall simulation is summarized as shown in Algorithm 4.3. The simulation is run with a completely missing random case as the scenario.

Note that the algorithm is run on a per-day basis. This means that the model is run at the beginning of each day and is trained at the end of the day.

Algorithm 4.3 Proposed AQT Algorithm

```
Inputs:
    Time step, t
    Max Time Step, T;
    Traffic Data, X;
    QRF Clusters, C;
    Number of Clusters, K;
    Trained ARIMA Model;
    Switch Threshold, SW;
Outputs:
    Imputed Traffic Data, $x_{pred}$;
While t != 0 /* t=0 represents 12am or end of day */
    Obtain road data for the current time step, $x_t$
    If isMissing($y_t$)
        Determine the current missing rate, MR
        If MR < SW
            Use trained ARIMA model for imputation
        Else
            Use a trained QRF model for imputation, as shown in Algorithm 1.
        End
        Update $\mathbf{x}_{pred}$ (day, end+1) = imputed Value for ARIMA Model.
    Else
        Update $\mathbf{x}_{pred}$ (day, end+1) = xt for ARIMA Model.
    End
    t = mod(t+1, T) /*Next time step */
end
Using the Tensor Factorization algorithm, refill recorded missing data for the current day.
```

(II)Determining the Switching Threshold

Based on TABLE 4.9 and TABLE 4.10 in Section 4.6, the ARIMA method's accuracy drops sharply when at 70% missing data. Hence, further analyses were done around that value to determine if there is a better point to be used as the switching threshold. This is shown in TABLE 4.4 and TABLE 4.5:

TABLE 4.4

RMSE Comparison of Proposed Model with Different Switching Thresholds at Different Missing Rates

	RMSE			
	Switching Threshold			
Missing Rate (%)	0.5	0.6	0.7	0.8
0.1	0.3983	0.3973	0.3988	0.3869
0.3	0.4692	0.4708	0.473	0.4647
0.5	0.5546	0.5256	0.5306	0.5432
0.7	0.6862	0.6889	0.6617	0.7009
0.9	0.837	0.837	0.837	0.8571

TABLE 4.5

MAPE Comparison of Proposed Model with Different Switching Thresholds at Different Missing Rates

	MAPE			
	Switching Threshold			
Missing Rate (%)	0.5	0.6	0.7	0.8
0.1	0.8267	0.8248	0.8263	0.8238
0.3	0.9446	0.9464	0.9476	0.953
0.5	1.1353	1.0688	1.0711	1.1102
0.7	1.4199	1.4218	1.365	1.4103
0.9	1.7314	1.7313	1.7319	1.8083

Based on the comparison of RMSE and MAPE, it shows that, on average, a switching threshold of 0.7 displays consistently better results at higher percentages of missing data, although not necessarily the best. It can also be seen that for lower percentages of missing data, a switching threshold of 0.8 may be preferable. However, as a larger priority would be given to a higher percentage of missing data, a switching threshold of 0.7 was chosen for the remainder of this thesis.

4.4.3) Determining the Optimum Number of Clusters, K

There are a number of heuristic methods to determine an appropriate number of clusters. Two widely used methods will be discussed in the following subsections, namely the Elbow method and the Silhouette method.

(A) Elbow Method

The elbow method calculated the square of error, the Euclidean distance between the points in each cluster with their respective centroids. These sum of squared errors (SSE) are used as a performance indicator. Hence, the K-value is iterated for a range of values up to a maximum of the number of observations, their SSE is plotted on a figure, and the inflexion point, or 'elbow', is chosen as the optimum number of clusters. This is shown in Figure 4.8, of which the inflexion point is at K=4, making the assumed optimum number of clusters to be 4.

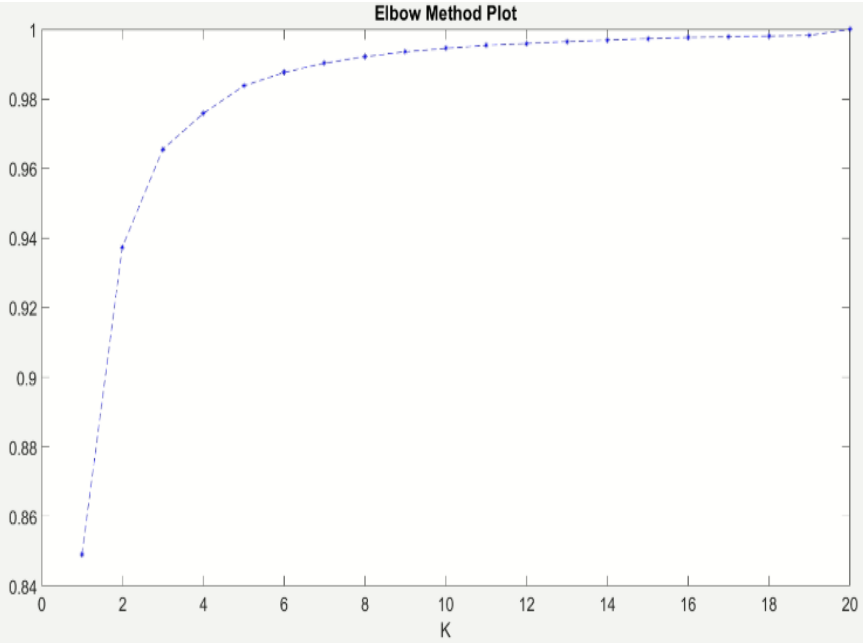


Figure 4.8. Elbow Plot of the data set, the inflection point

(B) Silhouette Method

Proposed by Peter J. Rousseeuw [154], the silhouette method measures the similarity of a sample point to its own cluster (known as cohesion) compared to its similarity to other clusters (known as separation).

The range of the value is between -1 to +1, and the higher the value, the better, as it indicates that the point is most similar to its own cluster. Low silhouette values would mean too few or too many clusters. The silhouette value is calculated as shown in Equation (4) below:

$$Silhouette\ Value, s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} = \begin{cases} 1-\frac{a(i)}{b(i)}, & a(i) < b(i) \\ 0, & a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & a(i) > b(i) \end{cases} \qquad (4.13)$$

The silhouette values were calculated using MATLAB's evalclusters function, and the values are shown in TABLE 4.6.

TABLE 4.6

Silhouette values for K=2 to K=10, the highest valued cluster is highlighted in bold

K	Silhouette Value
2	0.4719
3	**0.5347**
4	0.4774
5	0.3837
6	0.3545
7	0.3602
8	0.3581
9	0.3690
10	0.4225

TABLE 4.6 shows the average value of each number of clusters after running the evalclusters function ten times to ascertain the optimum number of clusters, as it is possible for the same evaluation to return a different optimum number of clusters. Pairing this with the knowledge obtained from the elbow method would provide a better insight into the cluster number selection, which in the case of the silhouette method, would be K=3, with the second highest being K=4.

(C) Cluster Selection Decision

The two heuristic methods described above have shown that the assumed optimal number of clusters is either K=3 or K=4. During the simulation, K=4 is found to perform a little better than K=3, while higher values of K do not seem to produce any significant improvement in performance. Hence, heuristic methods used are good at providing an estimation of the number of clusters to use. Due to the better results, K=4 is used for the remainder of the thesis.

4.5) Time Complexity Comparison - ARIMA-QRFC with Tensor Factorization (AQT)

When it comes to real-time systems, it is important that the system respond quickly while still retaining its accuracy.

The comparison between the complexity of the tensor factorization method, ARIMA model, as well as Quantile Regression Forest (QRF) methods are shown in TABLE 4.7 below. Due to the many tensor factorization methods available, a simple matrix multiplication complexity is used as the reference (with tensor factorization obviously being more complex), whereas QRF is based on the complexity of the random forest algorithm. TABLE 4.8 describes the notations used in TABLE 4.7.

Meanwhile, the complexity plots for ARIMA and tensor factorization are shown in Figure 4.9, and the run-time complexity of QRF is shown in Figure 4.10. It can be seen that tensor factorization is slower than ARIMA. Although a single iteration of tensor factorization would likely be faster than QRF, the case might not be true for a larger number of iterations required for the tensor factorization methods to reach convergence. The run-time of QRF is also decided prior to the simulation due to the parameters used, whereas the complexity of tensor factorization as well as ARIMA increases along with the amount of data.

It can be seen from the tables and figures that QRF's computational time does not change much during run-time, whereas ARIMA increases at a slower rate compared to tensor factorization, which makes these two methods more robust and consistent during the data imputation.

The results show that the less computationally complex and consistent ARIMA and QRF are more suitable for real-time applications than the tensor factorization method. Hence, those two would be used for real-time imputation, while the tensor factorization would be used for training purposes.

TABLE 4.7

Big O Notation estimates for Tensor Factorization, Quantile Regression Forest, and ARIMA

Tensor Factorization (Per Iteration)	**QRF (Tree-Building)**	**QRF (Run-time)**	**ARIMA**
O(G * log (G))	O(ntree*w*p*n)	O(w * ntree)	O(n)

TABLE 4.8

Description of Notations Used in TABLE 4.7

Symbols	Description
G	MNF, where M, N, and F are the x, y, and z dimensions of the tensor
R	The rank of the tensor
J	$J = \prod_{n=1}^{N} I_n$
w	Number of Variables
p	Depth of tree
n	Number of data

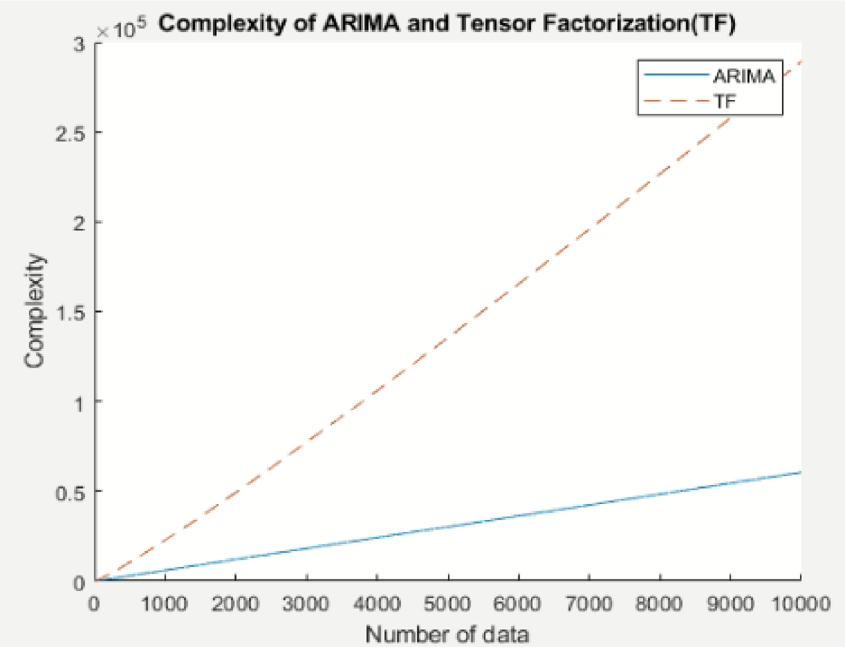


Figure 4.9. Complexity Plot of ARIMA vs Tensor Factorization

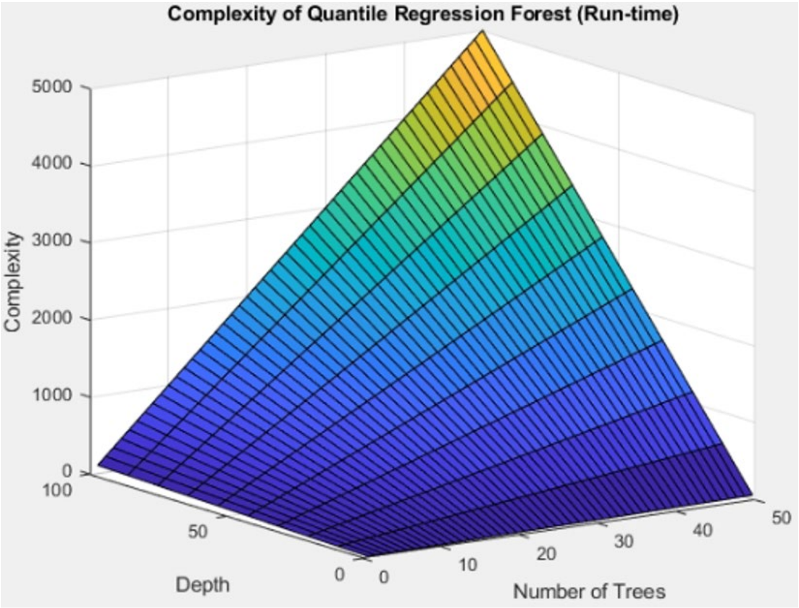


Figure 4.10. Complexity Plot of Quantile Regression Forest for run-time scenario

4.6) Results - ARIMA-QRFC with Tensor Factorization (AQT)

The performance of the tested models is compared using the RMSE and MAPE calculations shown in Equation 3.3 and 3.4 and are tested on various sets of random missing cases of different levels of missing rates, ranging from 10% missing rate to 90% missing rate. TABLE 4.9 and TABLE 4.10 showed the RMSE and MAPE performance between various models, with the first column using a referenced tensor factorization method, BATF [15], as the base. Note that any state-of-the-art tensor factorization can be used; BATF is used for the sake of comparison. Following that, the performance of the ARIMA model, the ensemble between QRFC and BATF, and the ensemble between ARIMA and BATF are checked on top of the proposed model. The simulation is run a number of times with different RNG seeds, and the mean values are recorded in the tables below.

It can be seen that the ARIMA method is good for a low rate of missing data (<70%) when compared to other methods besides the proposed method but performs badly for a higher rate of missing data (>= 70%). Meanwhile, the QRFC method coupled with BATF training shows higher performance at 70% and 90% missing data.

Finally, it can be seen that the proposed AQT method that combines and takes advantage of when best to use each model has shown the best performance throughout all the test cases. This shows that the proposed AQT method can adapt to the fluctuation and missing data and choose the appropriate method accordingly, producing results better than any single system for all cases of missing data.

Being able to perform better than the referenced models at 90% missing rate shows that the proposed model is capable of handling various missing data scenarios, such as times when a sensor is unavailable for a period of time.

TABLE 4.9

RMSE table of the compared models for missing traffic data imputation

	RMSE				
Models	**Missing Rate = 10%**	**Missing Rate = 30%**	**Missing Rate = 50%**	**Missing Rate = 70%**	**Missing Rate = 90%**
BATF	0.6025	0.6076	0.6199	0.6718	0.9647
QRFC-BATF	0.5042	0.5642	0.6289	0.672	0.866
ARIMA	0.403	0.4747	0.5519	0.7319	1.4607
ARIMA-BATF	0.4035	0.4751	0.5578	0.7485	1.4919
Proposed - AQT	**0.3988**	**0.473**	**0.5306**	**0.6617**	**0.837**

TABLE 4.10

MAPE table of the compared models for missing traffic data imputation

	MAPE				
Models	**Missing Rate = 10%**	**Missing Rate = 30%**	**Missing Rate = 50%**	**Missing Rate = 70%**	**Missing Rate = 90%**
BATF	1.3081	1.3216	1.3525	1.4617	1.9044
QRFC-BATF	1.0522	1.1699	1.2863	1.3923	1.8182
ARIMA	0.8277	0.9482	1.1086	1.4385	2.7724
ARIMA-BATF	0.8315	0.9651	1.1352	1.4705	2.8042
Proposed - AQT	**0.8263**	**0.9476**	**1.0711**	**1.365**	**1.7319**

4.7) Chapter Summary - MOD3D-PAT and AQT

In order to better understand traffic patterns and behaviour, a mathematical model for approximating traffic congestion utilising basic traffic data was proposed in this research project, namely MOD3D-PAT, or Modified 3rd Degree Polynomial Approximation for Modelling Traffic Congestion. This model provides a high-level generalization of a location's expected urban traffic pattern based on minimal local traffic parameters and is easily interpretable compared to needing to train a machine learning model that may not be easily interpretable. The goal of the model prioritizes understanding of the traffic parameters and their effects towards the urban traffic pattern as a whole and less on attempting to accurately match the non-linearity of the urban traffic for a particular location.

The provision of a reliable traffic congestion model would help to incorporate intelligence in traffic systems as the number of drivers on the road continues to increase. It is possible to leverage already existing infrastructures provided by existing organisations to obtain real-time road traffic data by utilising online traffic API, allowing one to conduct traffic studies without utilising too many resources. A robustness test has been conducted for two other areas that further strengthen the proposed MOD3D-PAT derivation. The proposed model provides a good approximation for urban traffic. Comparing them to the main map, Bukit Bintang, it showed a very good approximation with the worst R-Squared of 0.93 and an RMSE of 0.035 and the best being an R-Squared value of 0.97 and RMSE of 0.024, proving that the model is valid based on the historical traffic data collected. Even when testing it against Damansara Utama and Bandar Sunway, the proposed model has been shown to be robust in providing a high-level generalization of urban traffic, proving that the proposed model is robust and dynamic. The proposed model differs from traditional traffic models in that it adds the uniqueness of its local parameters, such as traffic start time and peak start and end time, as well as estimating the traffic pattern based on the time of the day. Traditional traffic models tend to focus on the relationships between multiple traffic parameters but less towards time. In addition, the model can also be used to provide an estimate for uncertainty quantification and analysis regarding traffic patterns by being used as an input or even a benchmark and working together with historical data to improve the quantification of the aforementioned analysis further.

However, another issue faced by all data-related studies is the occurrence of missing data. It goes without saying that missing traffic data imputation is also an essential part of a fully functioning Intelligent Transportation System (ITS). Since without good data to work with, the system would be unable to provide an accurate idea of the traffic behaviour, which could lead to wrong traffic analysis being done, resulting in inaccurate traffic prediction and routing. In addition, traffic data was found to be heteroscedastic, and standard regression-based machine learning was determined to be negatively affected by outliers. This research has proposed an **A**RIMA-**Q**RFC with **T**ensor Factorization (AQT) method to overcome these problems while ensuring that the real-time operations are not affected,

The proposed adaptive model, AQT, utilised various strengths of each model to work in tandem to impute the detected missing data. The imputation was done while resolving the outlier and heteroscedastic problems and keeping the computational complexity low during the imputation operation. The results showed that the proposed adaptive model outperformed any single imputation method due to its adaptive nature. The proposed model used the switching imputation method for varying levels of missing data with ARIMA, Quantile Regression Forest, and K-means Clustering.

As seen in Figure 4.9, ARIMA has a much lower complexity compared to tensor factorization. At the same time, TABLE 4.9 and TABLE 4.10 showed that ARIMA also has better accuracy than tensor factorization when the missing data rate is 50% and below. The proposed model, AQT, further improved this by utilising tensor factorization in the training process when training the QRF model, increasing its accuracy. At the same time, Quantile Regression Forest (QRF) was paired with K-Means clustering to provide a quicker and more accurate data imputation for higher levels of missing data compared to the referenced models. For both the ARIMA and QRF, tensor factorization has been utilised on the side to train the network, making use of its accuracy and utilising both past and future data while not worrying about the longer computational time. As computation time is important for a real-time system, the proposed model was shown to be more suitable for real-time applications due to its lower complexity while maintaining a high accuracy level for data imputation.

Results have shown that the proposed AQT model is robust even for a high rate of missing data when compared to the state-of-the-art tensor factorization method as well as varying 2-model implementations between ARIMA, QRFC, and tensor factorization, with an RMSE of 0.3988, 0.473, 0.5306, 0.6617, and 0.837 and MAPE of 0.8263%, 0.9476%, 1.0711%, 1.365%, and 1.7319% for missing data rate of 10%, 30%, 50%, 70%, and 90% respectively. The dataset used for the simulation is the traffic data obtained via the HERE online traffic API [143] for the Bukit Bintang area of Malaysia.

Comparing the worst case of 90% missing data, the proposed model performed better than the referenced tensor factorization method, BATF [15], showing an improvement of 0.1725%, which is a 9% improvement from the referenced model's MAPE of 1.9044%. A similar improvement can be seen for RMSE, whereby the proposed model showed an improvement of 0.1277, which is a 13% improvement from the referenced model's RMSE of 0.9647. While the lower MAPE values showed that the proposed AQT model is more accurate on average, the lower RMSE values also showed that the proposed model is more robust towards the various sets of traffic data, indicating that the errors are smaller on average compared to the other models. This also proves that the model can handle outliers better than the other models, as although the RMSE has a heavier penalty towards larger errors, the final RMSE value remains small. This performance clearly proves the efficiency of the proposed method.

To summarise, this research has proposed a novel 3rd degree polynomial mathematical model to generalise traffic patterns in an urban setting, as well as proposed a novel, robust, and real-time missing traffic data imputation model where a K-means data clustering algorithm is combined with QRF to achieve better imputation and training results and minimize errors due to the heteroscedastic nature of the dataset.

Chapter 5: Traffic Speed Forecasting

One of the most significant traffic studies is the prediction of future traffic state, be it traffic speed, congestion levels, or even accidents. The most common among these, however, is the prediction of traffic speed, as it is also the most relevant when deciding the rerouting plans for drivers on the road. As mentioned in Chapter 1.2, studies into long-term traffic forecasting are lacking compared to short-term traffic forecasting, leading to a research gap. This chapter introduces the issues faced in traffic forecasting with regards to lacking feature data, as well as proposes a traffic speed prediction model utilising data augmentation via a novel segmented time frame clustering mechanism named Clustered Augmented LSTM (CAL). [3]

5.1) Introduction

Whether long-term or short-term, traffic prediction always faces a few common issues, one of which is the lack of feature data. There are traffic studies that utilise events [155], the weather [156], or other contextual factors [6]. These data aren't always available, possibly due to the inability of the service provider to install their sensors or establish the required tools and human resources at the respective location. Despite the data being available, there are times when such data's accuracy must be scrutinized as it may not be entirely accurate. For example, a weather API could determine whether a particular location is raining while the actual weather at the precise area of interest is clear due to the location's size or the API's inaccuracy.

Due to this, there is a need for data augmentation to overcome the limitation in time series data [157]. [158] has done an empirical survey on various methods of data augmentation, most of which are transformation-based data augmentation, which makes sense as data augmentation is used because there is a lack of data in the first place. However, it can be noted that such data augmentation methods tend to show somewhat inconsistent or ineffective results at times and require much handling and care to ensure that the generated data via transformation still conforms to the original data's pattern and behaviour. Failing which, the model would end up performing even worse than it should or show no improvement at all. As such, it is recommended to find other ways to supplement the lack of data, such as decomposing the data into its frequency response via FFT [159] or wavelet decomposition [21].

The proposed model, Clustered Augmented LSTM (CAL), presented in this chapter aims to resolve these issues by augmenting the input with an additional feature which is cluster mean obtained via K-Means clustering. While there is the choice of using exemplars - which are actual data points in a cluster - instead of the mean for clustering algorithms, the mean of the cluster was chosen for its computational efficiency and robustness to outliers when compared to using exemplars, which are more sensitive to outlier data compared to the mean as they represent actual data points. Also, by splitting one day's traffic data, it is possible to improve the results of existing prediction models for long-term traffic prediction.

To summarise, the proposed model CAL model introduced in this part of the research focuses on the objective of developing an accurate long-term traffic prediction model by implementing machine learning models but also includes the objective of doing so in a feature-limited setting. Additionally, seeing as GRU is becoming more popular in recent years and is similar in design to LSTM, a comparison between the two was made by changing the implemented LSTM to GRU for the proposed model as well.

[3] Part of the content of this chapter appears as 'R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban "Long-Term Traffic Speed Prediction Utilising Data Augmentation Via Segmented Time Frame Clustering,"' which has been submitted to Expert Systems with Applications for review

5.2) Proposed Model Set-up - Clustered Augmented LSTM (CAL)

The proposed CAL model utilises a K-Means clustering algorithm in order to cluster the training dataset, clearly grouping similar traffic patterns, which have been split and segmented, and labelling them. A Convolutional Neural Network (CNN) is then trained to identify these traffic patterns with and output their corresponding label. These labels are then used to extract the cluster mean, representing each cluster which will then be used as an additional feature vector for the Long Short-Term Memory (LSTM) model to overcome the challenge of lack of additional feature vector.

To reiterate, there are three main steps to the proposed model:

1. Pre-processing of the traffic data via splitting and segmentations.
2. Clustering of the traffic data for cluster recognition and data augmentation.
3. Utilising the LSTM for training and outputting the necessary predictions.

The following subsections below will go through each of these steps in a more detailed manner.

5.2.1) Pre-processing of Data

The traffic data Y is normalized using min-max normalization to reduce the possibility of dealing with other data of different ranges. After that, every two consecutive days' worth of traffic data are grouped, and a day's worth of time frame is taken and stored in a separate matrix. For example, the 12:00 am to 11:55 pm worth of traffic is stored, followed by the 12:05 am to 12:00 am, and so on. This process is repeated through all the traffic data collected while concatenating the matrices with the same time frame, resulting in as many matrices as there are time steps in a single day (e.g., For a 5-minute interval traffic data, there are 288 observations of traffic, resulting in 288 matrices of different time frames). Figure 5.1 shows an example of splitting the traffic data for $t = 1$(12:00am) and $t = 144$(12:00pm), where t is the time step of the first data taken by the time frame, and traffic speed represents the normalized traffic speed.

These matrices are then further split based on the daySplit parameter, which determines how many equal segments should a day's worth of traffic be split into. For example, a daySplit of 4 would split one day's worth of traffic data into four segments, each containing 6 hours' worth of traffic data. Figure 5.2 displays the split that would be made for t=1 when daySplit=4. The traffic data would be similarly split equally for other values of daySplit.

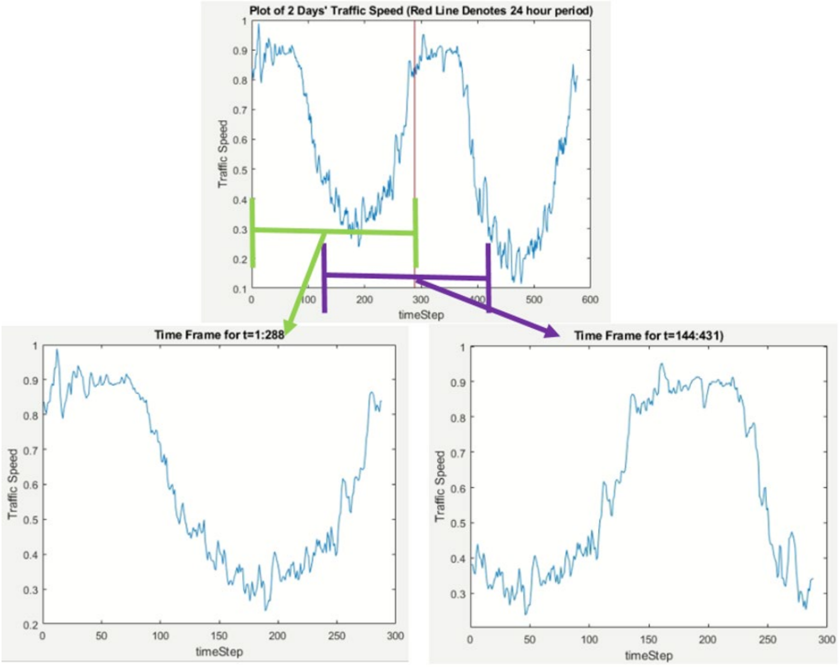


Figure 5.1. Example of Splitting Traffic Data into Time Frames for t=1 and t=144

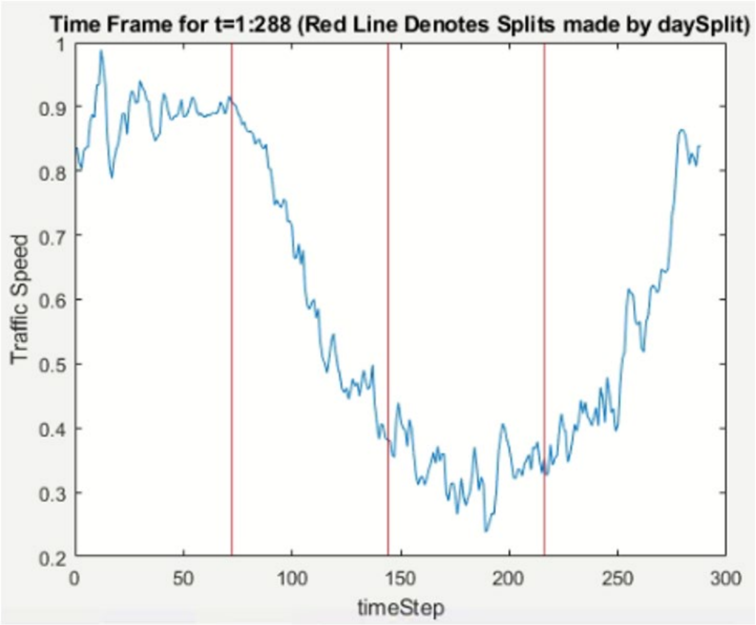


Figure 5.2. Time Frame of t=1 Split into 4 parts for daySplit=4

5.2.2) CNN for Cluster Recognition and Data Augmentation

Utilising the K-Means clustering algorithm, the historical traffic data are clustered and labelled into $K = 4$ clusters for each segment. i.e., For each segment of each time frame, there are 4 clusters. $K = 4$ was used as a study similar to [5] was conducted on the traffic speed pattern in Bukit Bintang using K-Means Clustering, and 4 clusters were deemed to be the most optimal in order to obtain distinct traffic clusters.

Traffic Speed Forecasting

A CNN was then trained to identify traffic patterns that are closest to each cluster and assign the most appropriate label to a particular traffic pattern. The total number of clusters would be $K * daySplit * (number\ of\ data\ points\ per\ day)$.

The appropriate cluster mean is then utilised as an additional input to the following LSTM model, which is used for the actual traffic prediction.

CNN model was used thanks to its strong classification ability to classify future traffic speed patterns according to the labels trained. Note that in the event new traffic patterns are required to be trained (i.e., non-recurring traffic patterns such as accidents), the CNN would have to be retrained again to include the new labels.

5.2.3) LSTM for Traffic Speed Prediction

Long Short-Term Memory (LSTM) is widely used for various time series prediction tasks due to the nature of capturing long-term correlations due to having an internal memory which is the main characteristic of the Recurrent Neural Network (RNN) architecture. However, unlike RNN, LSTM resolves the gradient issues, such as exploding gradient and vanishing gradient problems that the standard RNNs face [119], which further improves its robustness when training in time-series predictions.

The LSTM is trained using the current segment of traffic data along with the aforementioned cluster mean output by the CNN as the inputs and the future day-ahead segment as the output.

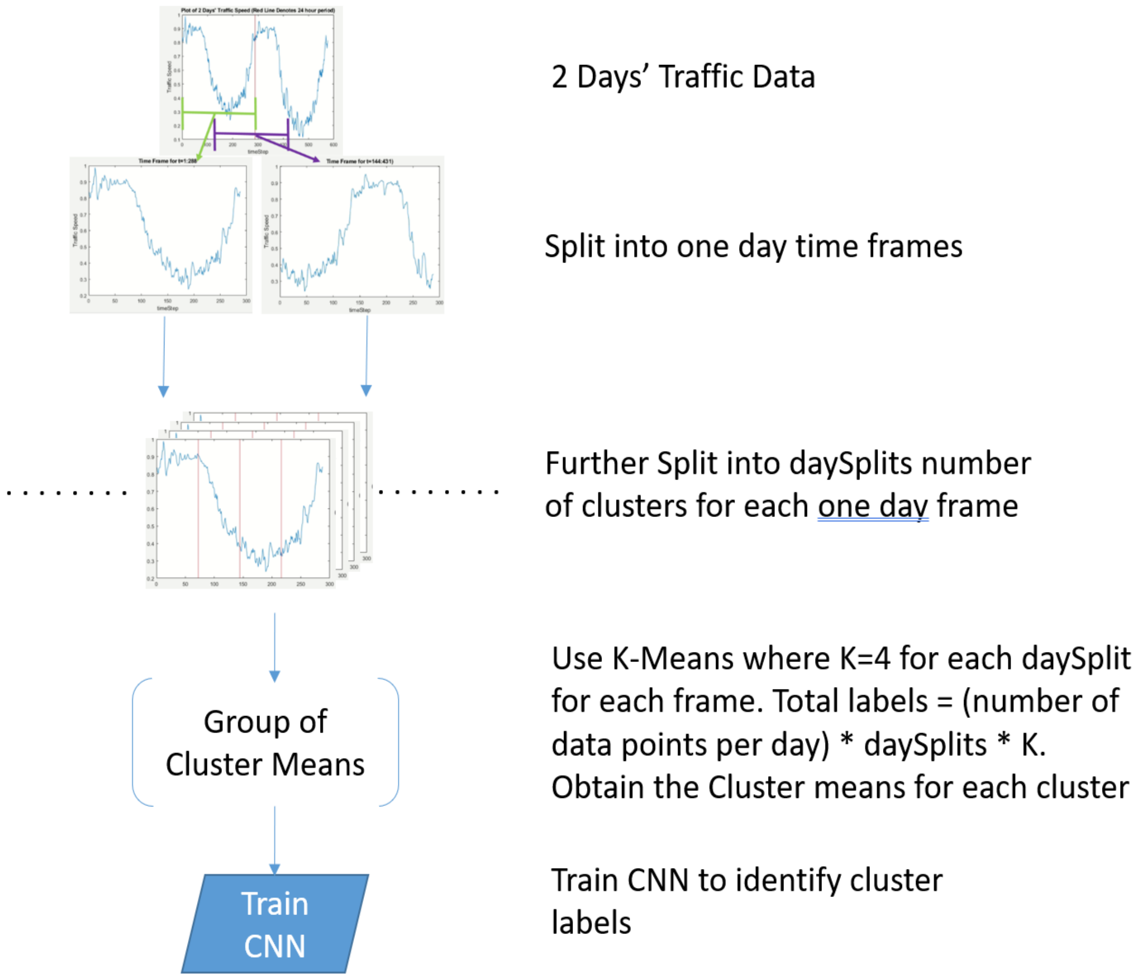
To capture the non-linearity of the traffic data, a two-stacked LSTM model is used to ensure a deeper level of feature abstraction. This also helps to reduce the issue of few input features being available, as the lack of feature data is one of the issues faced in many time-series prediction studies. It is for this reason that the extra feature via the CNN cluster recognition and the two-stacked LSTM models are used.

Training of the model is done via shuffling the dataset and using only 50% of the dataset to prevent overfitting, as the differences between each time frame could be too small, resulting in the model not being trained well.

5.2.4) Summary of Overall Algorithm

This subsection aims to clarify the training as well as forecasting steps that the proposed CAL model took by displaying the steps in a pseudocode form, as shown by Algorithm 5.1 and Algorithm 5.2, respectively. These Steps are further highlighted in Figure 5.3 and Figure 5.4, both representing the training of the CNN for cluster recognition and the corresponding training, as well as prediction using LSTM, respectively. To further highlight the key points of the model, the uniqueness of the proposed model is in the data pre-processing and clustering shown in Figure 5.3 and explained in Algorithm 5.1, as it breaks down 24-hour frames into smaller frames before using K-means clustering to label the different frames, grouping the ones with similar frames in order to provide a more accurate prediction for each section. The CNN is trained to recognize and label future segments based on the trained data which will help in determining the cluster mean data used as input into the LSTM as augmented data.

Note that the training and validation traffic data, y, is a vector with $288 * N$ elements, where N is the number of days, and 288 represents a day's worth of data points for a 5-minutes interval dataset. Meanwhile, t represents the first point at which a time frame is taken, ranging from 1 to 288, to ensure the correct slices are grouped together.



2 Days' Traffic Data

Split into one day time frames

Further Split into daySplits number of clusters for each one day frame

Use K-Means where K=4 for each daySplit for each frame. Total labels = (number of data points per day) * daySplits * K. Obtain the Cluster means for each cluster

Group of Cluster Means

Train CNN

Train CNN to identify cluster labels

Figure 5.3. Flowchart for the training of the cluster recognition via CNN

Algorithm 5.1 Training Algorithm

```
Input:
    Training Traffic Data, y
    daySplit, ds
Output:
    Trained CNN Network, CNNnet
    Trained LSTM Network, LSTMnet
for i = 1:y-1
            TF{t}(:, end+1) = vector y(i:i+287)_
      end
      for i = 1:288
            Split the time frames in TF{i} into ds segments, resulting in length(TF) = 288*ds
      end
      for i = 1: length(TF)
            [labels{i}, cMean{i}] = kmeans(TF{i})
            Ensure labels are continuous values from 1 to 288*ds*K
end
Shuffle both TF and labels
Train CNNnet with input=TF, output=labels
foreach x=288/ds segment of y(1:length(y-ds))
      X{end+1} = [x, cMean(CNNnet(x))]
end
foreach x=288/ds segment of y(2:length(y-ds+1))
      Y{end+1} = x
end
Remove 50% of X and Y to reduce the redundant training set
Train LSTMnet with input = X, output =Y
```

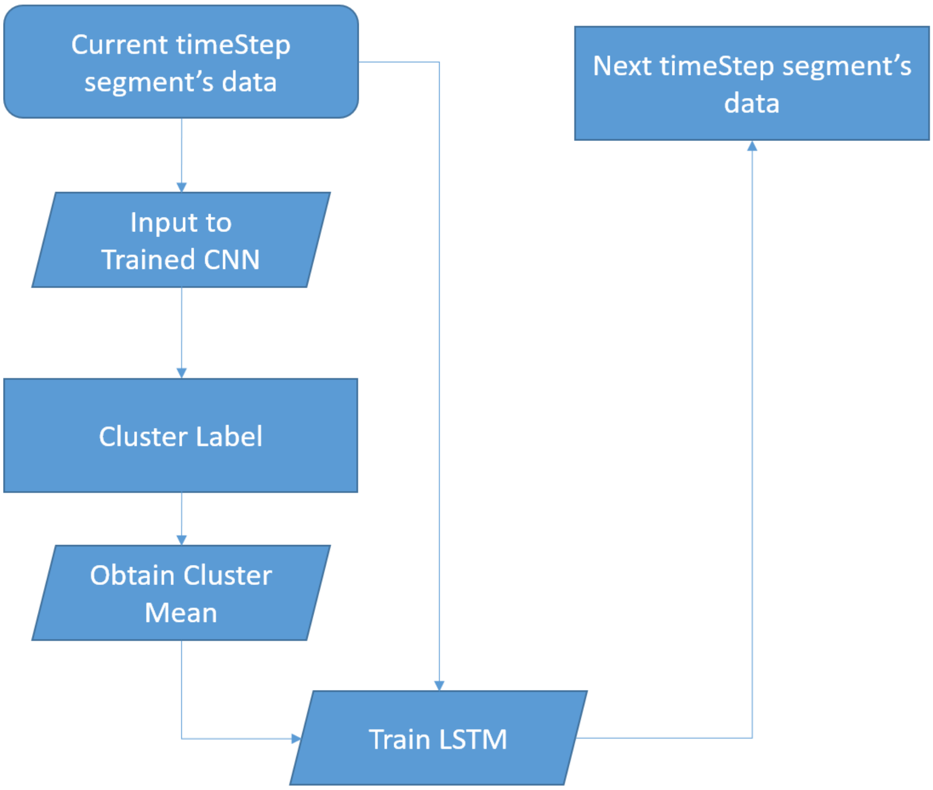


Figure 5.4. Flowchart for the steps used for prediction of the traffic data. Note that the same steps are used for training the LSTM

Algorithm 5.2 Forecasting Algorithm

```
Input:
    Validation Traffic Data, y
    daySplit, ds
    Cluster Mean, cMean
    Trained CNN Network, CNNnet
    Trained LSTM Network, LSTMnet
Output:
    Day Ahead Prediction, Y
    Actual Day Ahead Traffic Speed, $Y_{Actual}$
foreach x=288/ds segment of y(1:length(y-ds))
        X{end+1} = [x, cMean(CNNnet(x))]
    end
    Y = LSTMnet(X)
foreach x=288/ds segment of y(2:length(y-ds+1))
        $Y_{Actual}$ {end+1} = x
    end
Perform performance comparison using Y and $Y_{Actual}$
```

5.3) Results and Analysis - Clustered Augmented LSTM (CAL)

This section describes the implementation of models to be used as a benchmark, which are two basic models and one referenced model. The basic models are the standard CNN-LSTM and CNN-GRU models, while the referenced models are wavelet CNN-LSTM (W-CNN-LSTM). These models are then compared with the performance of the proposed CAL model. The performance analysis done uses the RMSE and MAPE calculations shown in Equation 3.3 and Equation 3.4.

5.3.1) Simulation

The referenced wavelet CNN-LSTM model (W-CNN-LSTM) [21] is implemented in MATLAB and simulated along with a CNN-LSTM and CNN-GRU models, whereby the architecture of the CNN-LSTM is used in the architecture of the W-CNN-LSTM model with the addition of the steps involving the wavelet transform, which is a 3-stacked CNN with a dilation factor of 1, 2, and 4 in order to obtain more information. A 2-stacked LSTM to capture any non-linearity in the traffic data. CNN-GRU is essentially the same but utilises GRU instead of LSTM. This is to cover any possible differences in performance between the two architectures. For this reason, the proposed CAL model is also compared with its GRU counterpart, named CAG instead.

Traffic Speed Forecasting

The simulation was conducted multiple times with different RNG seeds, and the averaged MAPE and RMSE were obtained and compared between the various models. The results are shown in the following subsection.

5.3.2) Results Discussion

This section displays the results obtained from the simulation conducted above. A summary of the averaged MAPE and RMSE is shown in TABLE 5.1 and TABLE 5.2 below, with the best results bolded:

TABLE 5.1

Averaged MAPE of Different Models for Day Ahead Prediction

Model	Averaged MAPE			
	DaySplit = 1	DaySplit = 2	DaySplit = 4	DaySplit = 6
W-CNN-LSTM	0.1286	0.1289	0.1274	0.1245
W-CNN-GRU	0.1214	0.1223	0.1223	0.1218
CNN-LSTM	0.1228	0.1221	0.1216	0.1211
CNN-GRU	0.1228	0.1224	0.1219	0.1213
CAL	0.1214	**0.1134**	**0.1069**	**0.1113**
CAG	**0.1192**	0.1138	0.1115	0.1116

TABLE 5.2

Averaged RMSE of Different Models for Day Ahead Prediction

Model	Averaged RMSE			
	DaySplit = 1	DaySplit = 2	DaySplit = 4	DaySplit = 6
W-CNN-LSTM	5.3348	5.3167	5.0789	4.825
W-CNN-GRU	5.0314	5.0347	4.861	4.6947
CNN-LSTM	5.0834	5.0236	4.8377	4.6698
CNN-GRU	5.0855	5.0324	4.8445	4.6777
CAL	5.0041	**4.7029**	**4.4194**	4.5259
CAG	**4.9354**	4.7556	4.5958	**4.5235**

Based on the tables above, it can already be seen that the proposed CAL, as well as the CAG model, has outperformed the rest of the referenced models. While CAG has a better performance than CAL for $daySplit$ = 1 and 6, when comparing both models' best case ($DaySplit$ = 4 for CAL and $DaySplit$ = 6 for CAG), CAL with $daySplit$ =4 was chosen instead as the model of choice. Besides that, a few more in-depth analyses can be done, which will be addressed in the subsections below.

(A) Effect of daySplit Parameter

As mentioned in Section 5.2.1), the daySplit parameter splits a single day's segment into n equal segments. This is to reduce the number of features in each data sequence, enhance each segment's features, as well as to increase the number of data that the network can use to learn.

Based on the results in TABLE 5.1 and TABLE 5.2, the improvements of the daySplit parameter showed an increase as the daySplit parameter value increased. In other words, the forecasting performance improves as the segments become smaller. This improvement is shown in the results of all the referenced models, showing that pre-processing the data into segments plays a vital part in the improvement of the prediction accuracy. The improvement is more prominent when looking at the RMSE performance index, which indicates that the daySplit parameter is especially good in reducing or minimizing the occurrences or magnitude of large errors during predictions.

However, unlike the other models, the proposed model shows a decrease in performance beyond the daySplit parameter of 4, which means that the optimum value for the daySplit requires some adjustments based on the models, although 4 or 6 segments would make sense considering the different phases of the day (i.e., midnight to dawn, morning rush hour, lunch period, after work hours). Note that regardless of the daySplit parameter value, the proposed CAL model still outperforms the other models, though it performs the best when daySplit = 4.

(B) LSTM vs GRU

Looking at the results from TABLE I and TABLE II, the GRU models' performance has consistently outperformed their LSTM counterparts, excluding the proposed model, albeit the difference in performance is relatively small. While not definitive proof, it can be thought that for day-ahead time series predictions, at least, GRU implementations should not be dismissed as it seems to perform better than their LSTM counterparts in general. Not to mention, for more time-sensitive systems that require a faster computational time, GRU would likely outperform LSTM as a whole due to lower computational complexity [125].

For the purpose of long-term traffic prediction, such as this thesis, however, it is not an urgent requirement. As such, maximum emphasis was placed on the accuracy of its forecasting capabilities. Hence, the LSTM implementation of the proposed model was ultimately chosen.

(C) Averaged MAPE and RMSE Per Time Frame

The averaged MAPE and RMSE shown in TABLE 5.1 and TABLE 5.2 represent the models' overall performance across the entire simulation. This subsection aims to break down the details of the results into Averaged MAPE and RMSE per time frame. As mentioned in section 5.2.1, the data is split into multiple time frames before being broken down into segments. The dataset used in this thesis is a 5-minute interval traffic speed data, resulting in 288 different time frames. This means that the performance for each timeframe group representing 12:00am to 11:55pm, 12:05am to 12:00am, 12:10am to 12:05am, and so on are looked at individually, and the average MAPE and RMSE of each group are plotted in order to visualise the performance of the models when predicting the day-ahead traffic for that particular timeframe. Each data point represents a group, with data point zero representing 12:00am to 11:55pm, data point one representing 12:05am to 12:00am, data point two representing 12:10am to 12:05am, and so on covering all 288 data points for the 288 time frames. The performance of the predictions for each time frame is obtained and plotted as shown in Figure 5.5 and Figure 5.6 for MAPE and RMSE, respectively.

Based on the results relating to daySplit=1, the performance of the models is shown to be somewhat similar on average. However, the proposed CAL model displays a more stable error as the variations per time frame are smaller than the other models. The error fluctuation for the proposed model is only somewhat similar to the other models in the case of daySplit=6, where its performance has started to decline, albeit still better than the other models.

For daySplit=4, the proposed CAL model performs exceptionally well for time frames relating to the early and late parts of the traffic, specifically the $t = 1$ to $t = 50$ and $t = 250$ to $t = 288$ time frames. For daySplit=4, this corresponds to time frames 12:00 am-5:55 am to 4:10 am-10:05 am and 8:50 pm-2:45 am to 11:55 am-5:50 am, respectively. This makes sense as these periods are when there are fewer traffic activities, resulting in smaller errors due to being more easily predictable. However, it is interesting to note that compared to the other models, only the proposed model with daySplit=4 displayed a noticeable change in performance between the time frames. In contrast, the other models show that their error fluctuates around a particular value in a similar pattern regardless of the value of the daySplit parameter. While it could be said that a similar situation occurs for the proposed CAL model for daySplit=2, it is significantly less than when daySplit=4. Perhaps further improvements could be made to reduce these peak errors in the future.

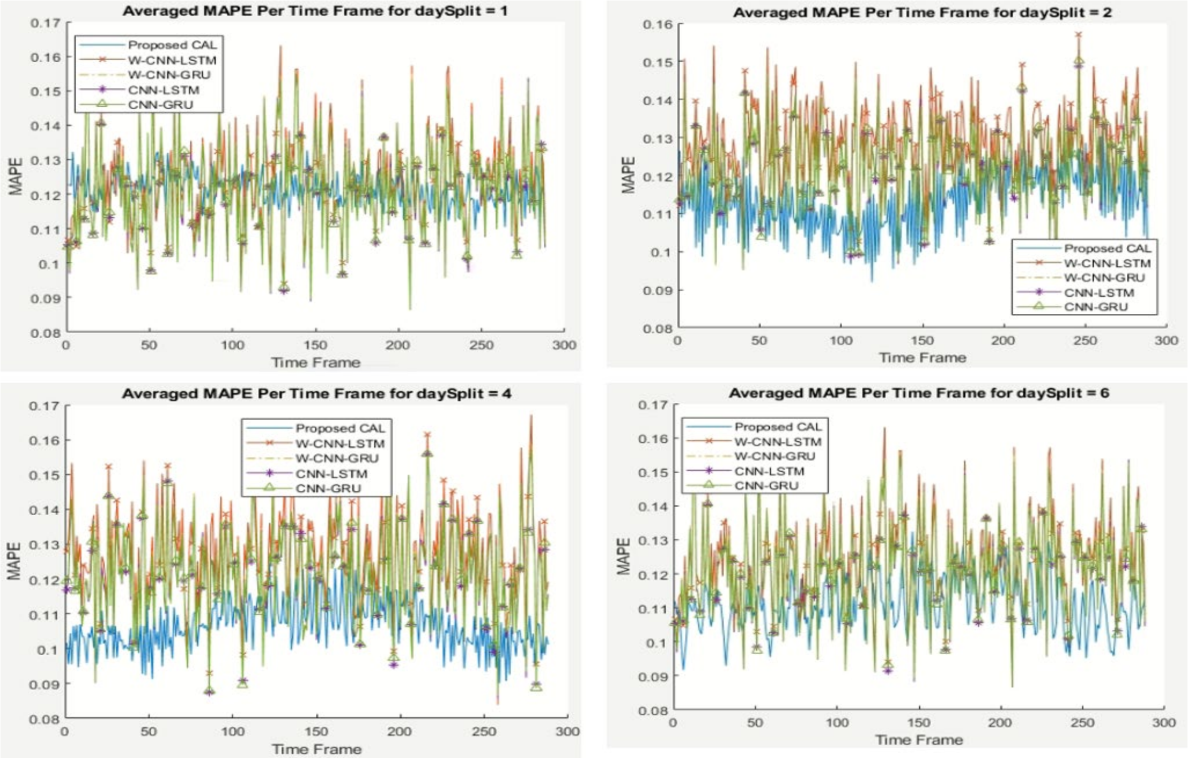


Figure 5.5. Averaged MAPE Per Time Frames for Top Left: daySplit=1, Top Right: daySplit=2, Bottom Left: daySplit=4, Bottom Right: daySplit=6

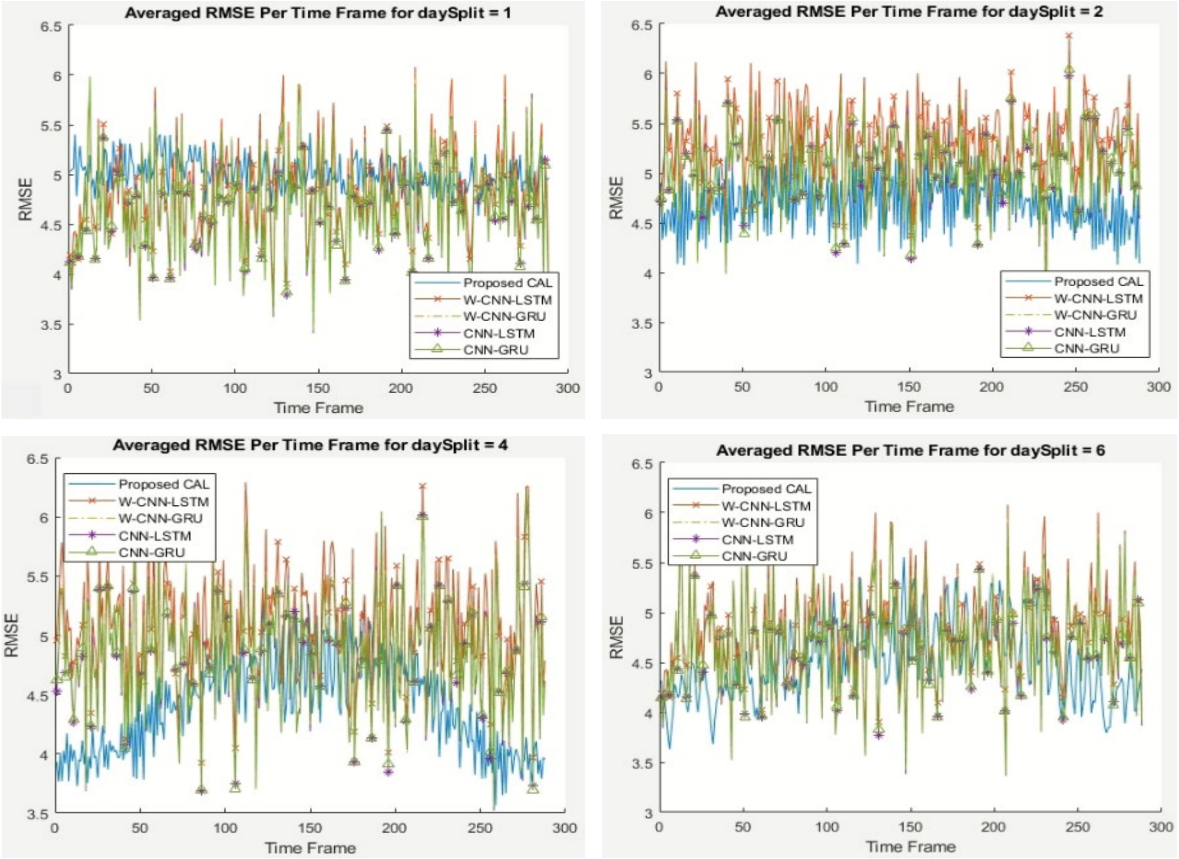


Figure 5.6. Averaged RMSE Per Time Frames for Top Left: daySplit=1, Top Right: daySplit=2, Bottom Left: daySplit=4, Bottom Right: daySplit=6

5.4) Chapter Summary - Clustered Augmented LSTM (CAL)

Long-term forecasting has not attracted much focus in research compared to short-term prediction since short-term predictions are considered more useful and manageable. However, long-term traffic predictions like day-ahead predictions can help the public plan their travels a day in advance and aid traffic departments in preparing for possible traffic congestion ahead of time, improving efficiency and productivity.

Much research proposed using various additional features that can help predict traffic. However, not all areas have easy access to their traffic data. Through an online Traffic API, research can be done on any particular location by leveraging the API company's existing infrastructure, albeit with relatively limited data features. This thesis aimed to overcome the limitations of the data features problem by proposing a Cluster Augmented LSTM (CAL) model for long-term traffic prediction. Clustering helps refine the traffic data to reduce the number of outliers that could cause errors in forecasting traffic speed, while the resulting cluster mean is also used as an extra set of features that the model can use to train, enhancing the accuracy of the data. Besides that, a $daySplit$ parameter was proposed, which would split a day's time frame into n equal segments, which helps to refine the data further and reduce deviations in predictions. It was found that the best performance for the proposed model is $daySplit = 4$, which might represent the different phases of traffic. However, that would require a separate analysis, as other referenced models show improvements even at $daySplit = 6$.

The proposed CAL model was trained and validated using traffic data obtained via the HERE Traffic API, and the RMSE and MAPE are both used as the performance indices of the model. When comparing their best results, the proposed CAL model has outperformed the referenced CNN-LSTM, W-CNN-

LSTM, and their GRU counterparts by 1.42%-1.76%, and the GRU implementation of the proposed model by 0.46% for MAPE and 0.25-0.41 and 0.1 for RMSE respectively. While the improvements may seem small, it should be noted that there is less than a 1% difference in performance when only comparing the other models. From this perspective, such an improvement is a testament to the performance of the proposed model. The results show that the pre-processing data methods and the proposed model have proven their ability for long-term traffic prediction.

Furthermore, while not conclusive, the performance of GRU has also been tested by replacing the LSTM cells in prevailing LSTM-based models with GRU instead and performing the same simulation. The GRU implementations have mostly shown an improvement compared to their LSTM counterparts, except for the proposed models, CAL and CAG, when comparing their best cases.

Chapter 6: Traffic Routing Optimisation

The fourth and final part of the research topic covers the last leg of the suite of processes that impacts drivers directly: Traffic rerouting. Referencing the research gaps mentioned in Chapter 1.2 and TABLE 2.4, it is noticed that existing literature has not looked into the impact of prioritising roads with more intersections compared to roads which have fewer intersections. Existing literature focuses their rerouting efforts based on a snapshot of the traffic state, either present or future state, and reacts accordingly. However, these models could very likely be improved by implementing a pre-emptive measure of prioritising less critical roads, which generally have fewer connections to other roads, instead of just prioritising the road with the shortest travel time.

This chapter has proposed an intersection-balancing rerouting model that would take this parameter into account and has proven that it does, in fact, has a significant impact on the number of successful arrivals for a given traffic simulation and map. [4]

6.1) Introduction

Nowadays, all kinds of navigational software such as Waze [160] or Google Maps Traffic [161] are being used daily by drivers to obtain the most optimum route to their destination, no matter how familiar they are with the area. Traffic routing has become such an important part of every driver's life due to the rise in congestion rates and the desire to get to their destination in the shortest amount of time. As previously mentioned, traffic routing is a key part of ITS and is built upon various processes, such as traffic prediction, to determine the state of the road and recommends the best path a vehicle should take, given its current position. Seeing how important traffic rerouting is, there have been many studies on the subject, and problems such as uncoordinated rerouting [10], [136] are seen to be the main concern in rerouting models, and just selecting the shortest route would not benefit the drivers as a whole. There are certainly drivers whose driving habits or context can be profiled, such as the research done in [162], but when using general traffic data via online Traffic API, it would be difficult to do. Even if the routing was coordinated, if the coordination was made inefficiently, the resulting routing would also not perform as well as intended. Various factors have to be taken into account, but recent studies have focused solely on the shortest travel time based on traffic speed predictions or distribution of harmful emissions [163] to improve air quality. In fact, there has been one factor which the authors have noticed to have been overlooked, and that is the number of intersections on the roads. Roads with more intersections tend to receive more traffic due to their connectivity. However, this leads to overutilisation of the road, leading to congestion between vehicles that has no choice but to take that particular road and vehicles which could have taken another, slightly longer road. To the best of the authors' knowledge, this situation has not yet been addressed in existing literature and is something that should be investigated.

To investigate the effect of rebalancing road weights based on their intersections and achieve the research's objective of formulating an efficient traffic routing algorithm in order to reduce travel time and air pollution, this part of the research has proposed a model called Decentralized Cluster with Intersection Balancing (DCIB), The key feature of the proposed DCIB rerouting model is that it considers the number of intersections each road has and balances the weights of each road accordingly. To be more specific, roads with more intersections will have a higher cost than roads with fewer intersections, making them a less desirable choice for the navigational system.

Furthermore, many traffic routing studies simulate their proposed work on a grid-like network and not a real urban network. This might impact the expected performance of the model due to the unrealistic

[4] Part of the content of this chapter appears as 'R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban "An Intersection-Balancing Decentralized Routing Model,"' which has been submitted to IEEE Intelligent Transportation Systems Magazine for review

grid road network. To better understand the impact and effect of the proposed and referenced traffic rerouting models, a real urban network should be used for the simulation.

Overall, the contribution of this work is 3-fold:

i. Simulate traffic routing models in a realistic urban network environment to ensure the reliability and accuracy of simulated performance.
ii. Proposed an intersection-focused road weight balancing algorithm that adjusts the weights of the road according to the number of intersections, allowing it to prioritize vehicles which has no other alternative roads to take it first.
iii. Developed an adaptive vehicle routing model capable of working well in an environment with connected and non-connected vehicles.

6.2) Proposed Methodology - DeCentralized Intersection-Balancing (DCIB) Routing Model

The proposed method utilises a cluster-based decentralized framework, similar to the referenced model, DEC-CTDSP [134] whereby vehicles within a certain radius are clustered, and their information is shared with one another. This is to take advantage of the future capabilities of CAV and their ability to communicate with one another in order to make more efficient routing decisions, as cooperation between agents has been shown to be effective in traffic routing such as [164]. However, this research focuses on not just that, but the coexistence of both CAV and non-CAV by incorporating another facet to the calculation of the route weights and how it would benefit both of them.

The base case, or ‘No Routing’ case, assumes all vehicles are routed according to the shortest path determined at the start of the simulation and this route does not change throughout the course of the simulation.

The proposed look ahead function works as follows: For a given planning horizon, the RSU will collect the vehicles’ route data and will then proceed to estimate the time each vehicle will take along each road based on their current speed and distance from the roads, as well as the state of the traffic signals. For example, for a planning horizon of 10 seconds, a vehicle requiring 5 seconds to pass through road A, followed by 10 seconds on road B, will have the RSU noting down the travel time along the road IDs A and B to be [1, 1, 1, 1, 1, 0, 0, 0, 0, 0] and [0, 0, 0, 0, 0, 1, 1, 1, 1, 1] for road A and B respectively. The additional time beyond the planning horizon is ignored and will be recalculated on the next calculation.

Through this, the RSU can estimate how many vehicles will be on the same road at certain times throughout the planning horizon and estimate the weights of the roads accordingly. However, the proposed model also introduces the concept of prioritizing the intersections with the fewer branches, in other words, the intersections with the fewer intersections. Hence, each RSU has an internal table of nearby intersections and their associated branches, and the RSU will modify the estimated weights of the roads by a factor of 1 to 2 in this simulation, as shown in Equation (7.1):

$$W = \frac{W_{lookAhead}}{B_{upper}} * IF \tag{7.1}$$

Where $W_{lookAhead}$ are the weights obtained using the look-ahead function described above, and W is a collection of the weights for all the roads in the network.

IF is the intersection factor by which to modify the weights along the road. In this research, the range of IF is set to [1, 2]. B_{upper} and B_{lower} represents the upper and lower bounds of the value, respectively, which are 2 and 1. The normalized IF follows the equation shown in Equation (7.2):

$$IF = B_{lower} + (intersections - B_{lower}) * \frac{B_{upper} - B_{lower}}{\max(\text{intersections}) - \min(intersections)} \qquad (7.2)$$

Where $intersections$ is an array consisting of the number of intersections each road has.

For example, if the maximum and minimum number of intersections is 4 and 1 (Which is essentially just a straight road), the IF of the 4-way intersection would be B_{upper} or 2, and the $intersectionFactor$ of a 2-way intersection would be 1.333, while the straight road has an IF of B_{lower} or 1. The range of [1, 2] was chosen as too large a value would skew the weights too much, while too small would result in a possibly smaller effect.

The RSU will then sequentially allocate the new route to each vehicle while also editing the route list it collected with the new route to ensure that the next vehicle gets an updated routing and is not based on older information. These steps are taken to ensure that within the limited information available in a decentralized model, the vehicles have as clear a piece of information as possible.

For non-connected vehicles, they are assumed to be using routing software in order to obtain the quickest route. This software is assumed to use a similar weighting function to the proposed one, except that there are no vehicles connected to it, which in turn means that the non-connected vehicles have no information on the state of the traffic aside from the distance and free-flow speed of the road. This is a conservative view, as modern navigational software does show the estimated state of the traffic. This scenario was chosen to demonstrate a completely non-connected vehicle.

It is important to note that all drivers are assumed to have the same priority in the traffic simulation, meaning there would not be any prioritised vehicles in the simulation, such as ambulance or police vehicles, just like the referenced model in [134].

6.3) Experimental Setup

The simulation is conducted using two different urban networks, Bukit Bintang and Sunway. Both are urban cities, with one focusing on office and shopping malls while the other focuses on educational facilities and shopping districts. As shown in [2], the traffic patterns for these two places are rather different.

HERE traffic API [143] is used to initialize the state of the traffic at the time of the simulation to mimic the current traffic pattern that is being experienced by the road network, as shown in [145].

6.3.1) Setting Up the Simulation

Map data was taken via Open Street Maps[165] which was subsequently processed and converted to the appropriate format for the traffic simulator SUMO [28] to run the simulation using SUMO’s netconvert tool. The routes were then created using the RandomTrips function while increasing the likelihood of the vehicles travelling between the fringes of the map, i.e., vehicles are more likely to have their origin and destinations be set on the fringes of the map to simulate entering and leaving the map, also to increase the minimum distance as well as the possible routes to be taken for each vehicle.

It is important to note that despite best efforts to increase the traffic network's realism, it is impossible to ensure vehicles follow the same route as those on the actual road network, and certain restrictions and considerations have been taken when running the simulation. However, the simulation is set such that most vehicles are to take the shortest distance to their destination as the default. The performance of this behaviour acts as the baseline against which the proposed model and the referenced model are benchmarked.

The following are the assumptions and restrictions made for the simulations:

1. Most vehicles, by default, take the shortest path to their destination.
2. The map is initialized based on congestion rates returned by the online Traffic API, HERE Traffic API [143].
3. Conversion of traffic congestion via the parameter returned by HERE, which is the jam factor, to the number of vehicles is listed in [145].
4. Connected vehicles are deemed to be able to properly communicate with other vehicles without any problems.
5. Non-connected vehicles may not share information with other vehicles; however, they would have a navigational system that would determine the best route based on the route cost calculation proposed by this thesis.

6.3.2) Performance Metric

The simulation outputs several data for performance analyses, namely average travel time, number of arrived vehicles, CO2 emissions, fuel emissions, and number of congested roads.

The definition of average travel time and the number of arrived vehicles are listed on the SUMO's website [166], while CO2 and fuel consumption are listed on [167] but are self-explanatory. However, the number of congested roads is taken by observing the average vehicle speed along the roads and determining obtaining the ratio of the average vehicle speed to the road's free-flow speed. If the ratio is below a certain threshold, it is deemed to be congested. For this simulation, a congestion threshold of 0.2 was used, meaning that if the ratio of average vehicle speed to free-flow speed is less than 0.2, the road is considered congested.

To summarize:

1. Average travel time: The mean travel time of all vehicles that have left the simulation within the previous and reported time.
2. Number of arrived vehicles: Number of vehicles that have reached their destination.
3. CO2 and fuel emissions represent the total CO2 emissions and fuel consumed during the simulation.
4. Number of congested roads: Number of roads whose average vehicle speed to free-flow speed is below the congestion threshold of 0.2 , where 1 is freeflow and 0 is standstill traffic.

6.3.3) Running the Simulation

The simulation is conducted through MATLAB using the TraCI [168] module via the TraCI4MATLAB [169] code. The relevant outputs were set up, and the simulation was run for both Bukit Bintang and Bandar Sunway for two separate cases, that there being 100% and 50% connected vehicles. As mentioned above, on top of the pre-generated vehicles and routes, there are also vehicles that are generated after processing the traffic data obtained via HERE online Traffic API [143]. The simulation was run for 3600 timesteps with a 200-second initialization phase to allow vehicles generated via the HERE Traffic API to populate the map.

Figure 6.1 shows the process flow of the simulation conducted. Additionally, an example of the simulation is shown for Bukit Bintang and Bandar Sunway maps in Figure 6.2 and Figure 6.3, respectively.

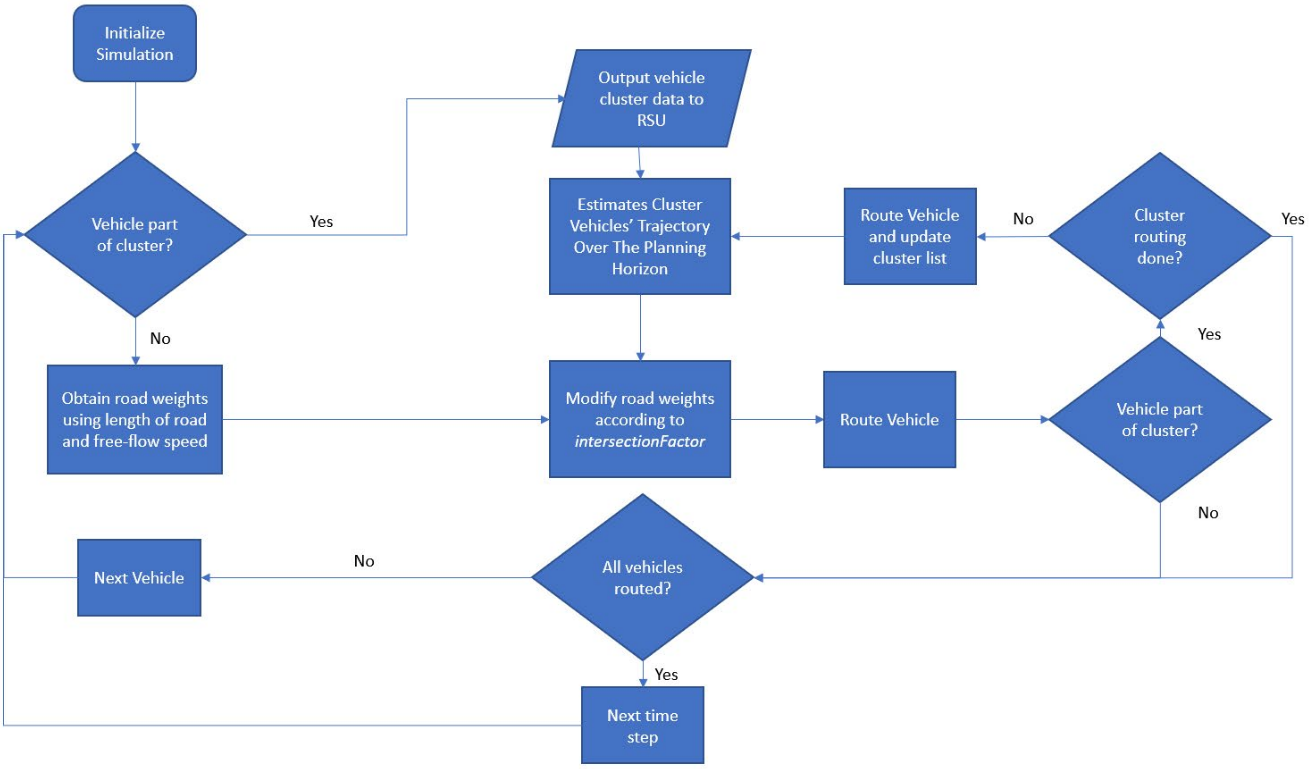

Initialize Simulation
Vehicle part of cluster?
Yes
No
Output vehicle cluster data to RSU
Estimates Cluster Vehicles' Trajectory Over The Planning Horizon
Route Vehicle and update cluster list
No
Cluster routing done?
Yes
Yes
Obtain road weights using length of road and free-flow speed
Modify road weights according to *intersectionFactor*
Route Vehicle
Vehicle part of cluster?
No
Next Vehicle
No
All vehicles routed?
Yes
Next time step

Figure 6.1. Flowchart of the simulation process

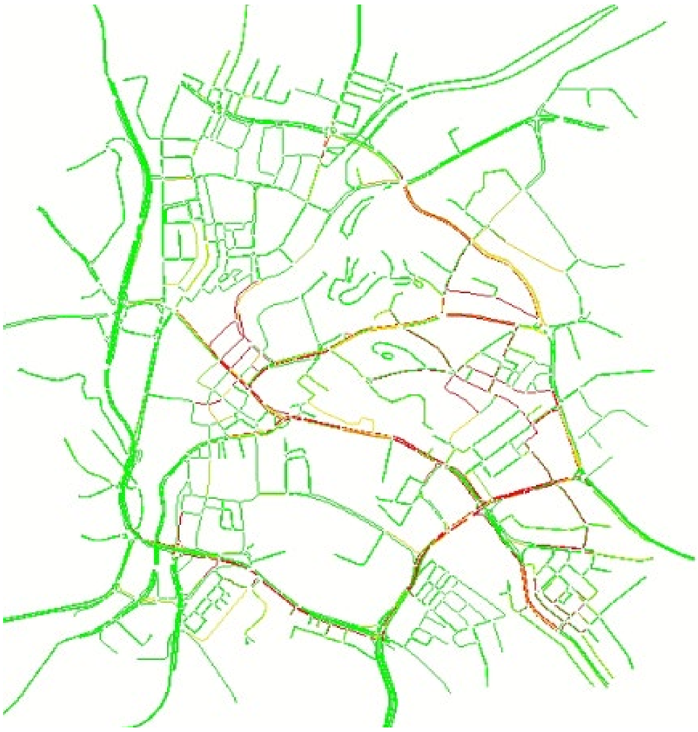


Figure 6.2. A snapshot of the SUMO simulation for Bukit Bintang

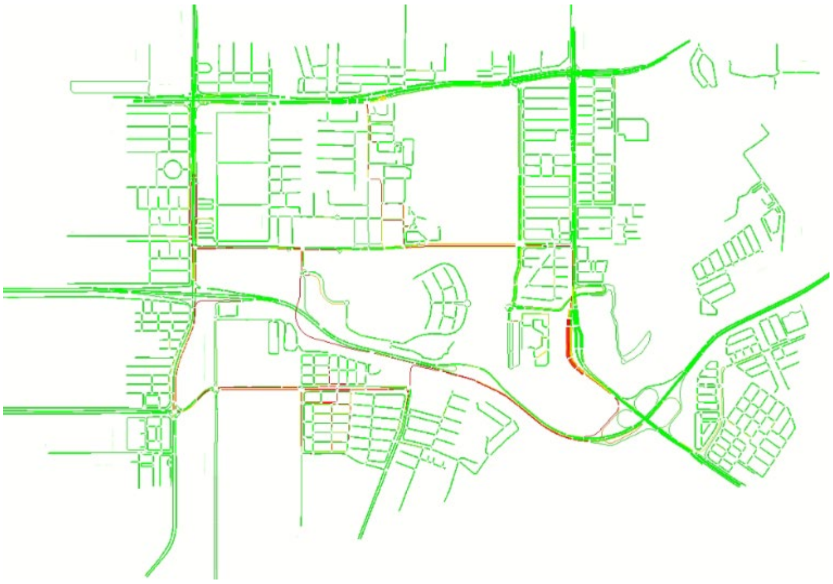


Figure 6.3. A snapshot of the SUMO simulation for Bandar Sunway

6.4) Results and Analysis - DeCentralized Intersection-Balancing (DCIB) Routing Model

The simulation was conducted and benchmarked against another decentralized routing model, DEC-CTDSP [134] and their performance at 50% and 100% connected vehicles were compared. The graphs for the number of congested roads, mean travel time, fuel consumption, and successful arrivals were plotted for both Bukit Bintang and Sunway maps. Note that CO2 emission has the same curves as fuel consumption and hence was not included to reduce redundancy.

6.4.1) Bukit Bintang

The graphs for Bukit Bintang are shown in Figure 6.4 to Figure 6.7. It can be seen that the proposed model consistently performed better than the referenced model at 100% connected vehicle for 100% connected vehicles, or in this case, connected and automated vehicles (CAV). In addition to that, the proposed model at 50% connected vehicles is actually performing competitively with the referenced model at 100% CAV initially, although it still loses out towards the end. However, a drastic improvement can be seen when comparing both 50% CAV cases between the proposed DCIB and referenced Dec-CTDSP model. Interestingly enough, the number of congested roads during most of the simulation showed that the proposed model at 50% connected vehicles is the lowest. However, this means that there are more vehicles taking a roundabout path compared to the 100% case where all cluster vehicles are aware of the other vehicle's route. This, in turn, resulted in increased mean travelling time and decreased successful arrivals, as shown in Figure 6.5 and Figure 6.7. Further calibration could likely improve this.

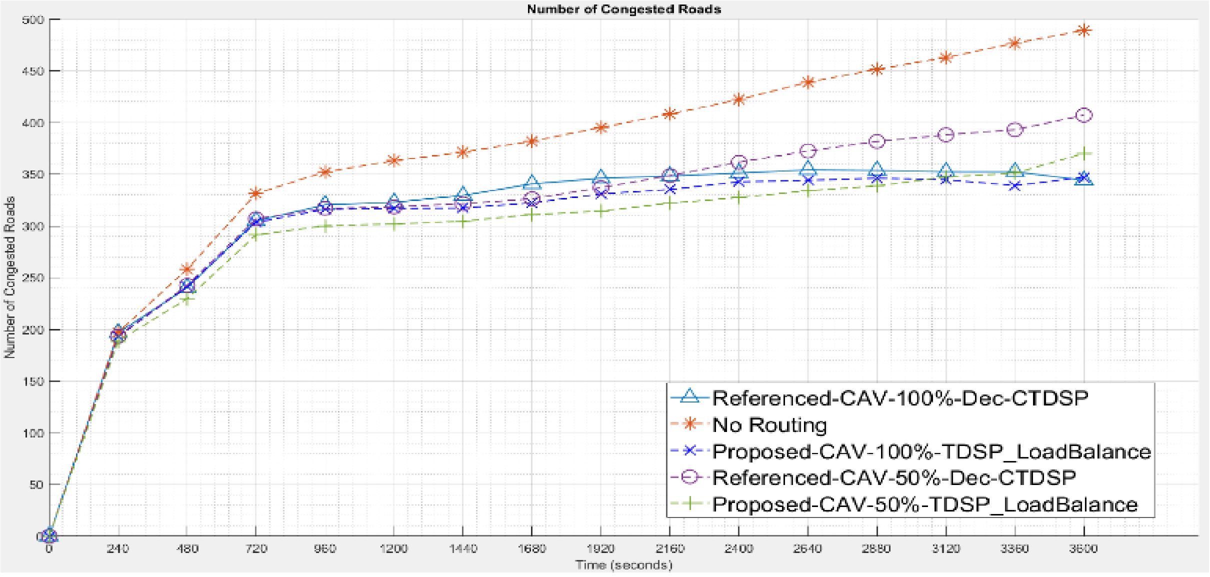


Figure 6.4. Number of Congested Roads for Bukit Bintang Simulation

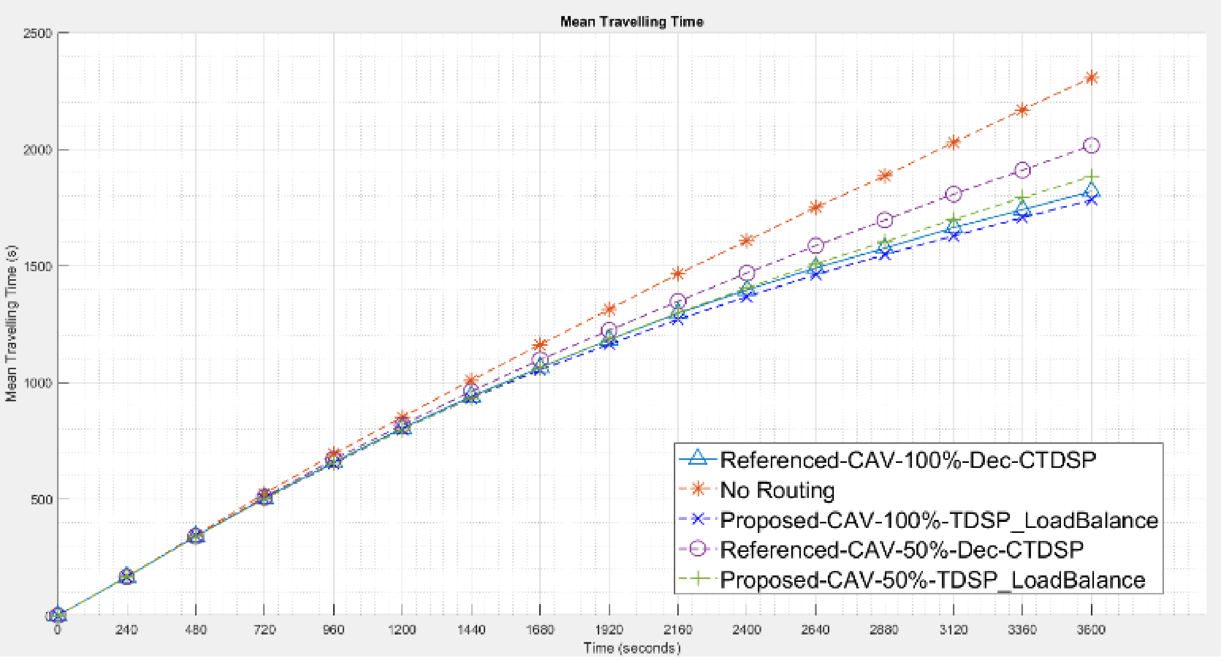


Figure 6.5. Mean Travelling Time for Bukit Bintang Simulation

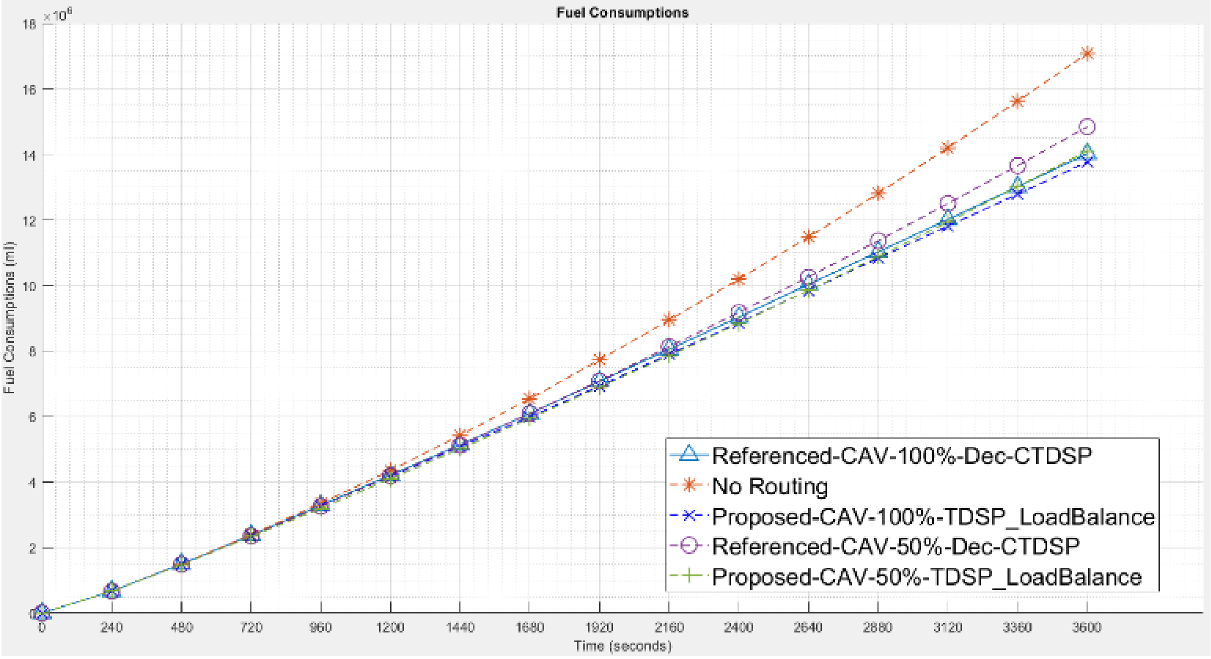


Figure 6.7. Fuel Consumption for Bukit Bintang Simulation

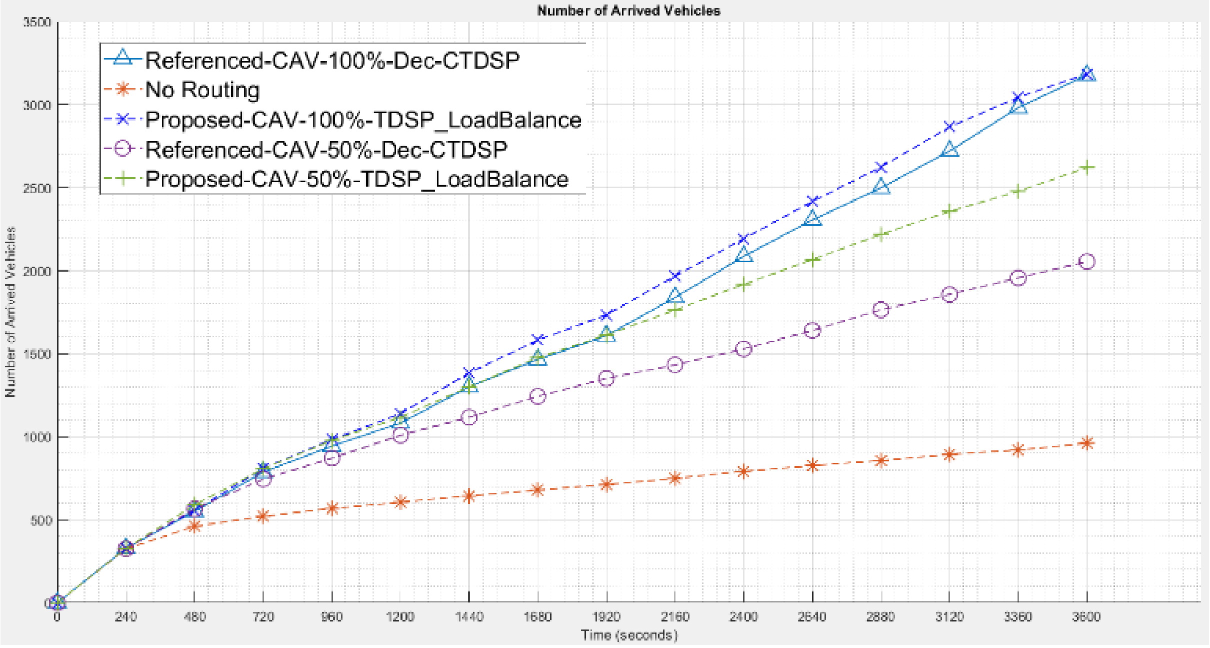


Figure 6.6. Number of Successful Arrivals for Bukit Bintang Simulation

6.4.2) Bandar Sunway

The graphs for Bandar Sunway are shown in Figure 6.8 to Figure 6.11. Looking at the number of arrived vehicles in Figure 6.11, the proposed model for both 100% and 50% connected vehicles has outperformed the referenced model by a significant margin. Interestingly enough, both the cases with 100% and 50% connected vehicles displayed very similar performance for this case, although the 100% connected vehicle version of the model still pulled ahead in the end. The same can be said for the mean travelling time in Figure 6.9 and fuel consumption in Figure 6.10. Similarly, for the number of congested roads, the proposed model has once again shown a consistent performance of having the least number of congested roads.

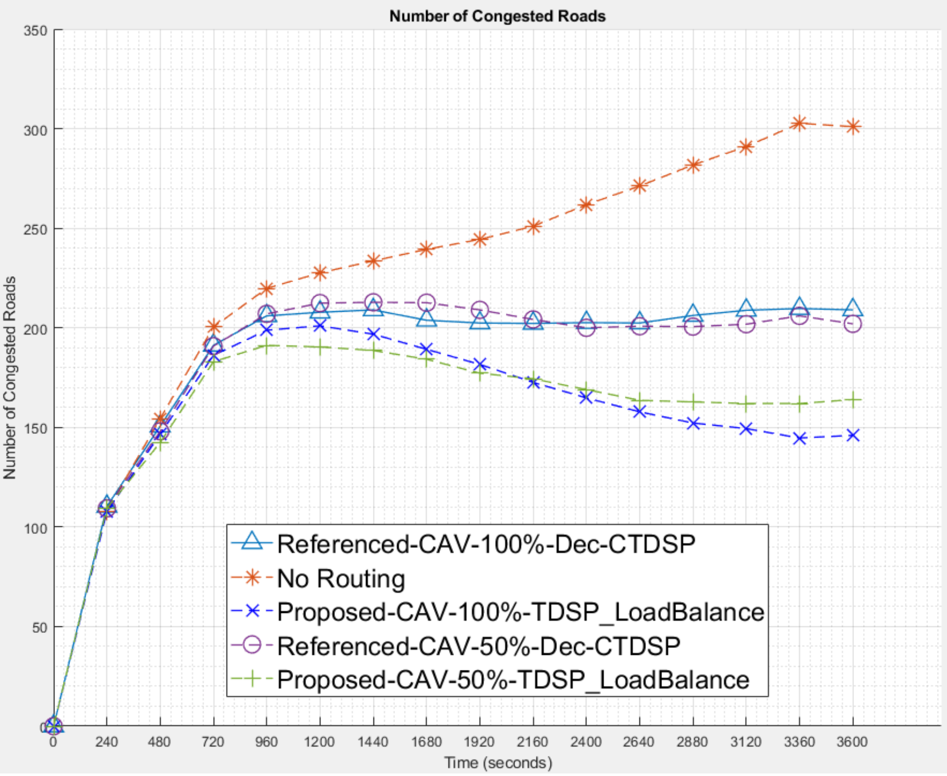


Figure 6.8. Number of Congested Roads for Bandar Sunway Simulation

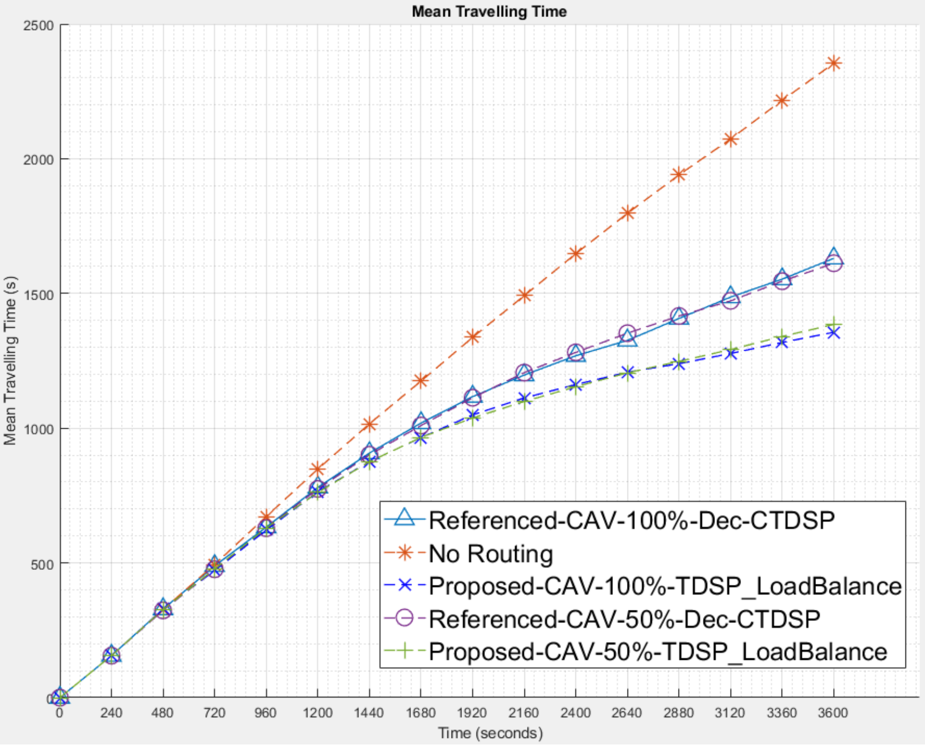


Figure 6.9. Mean Travelling Time for Bandar Sunway Simulation

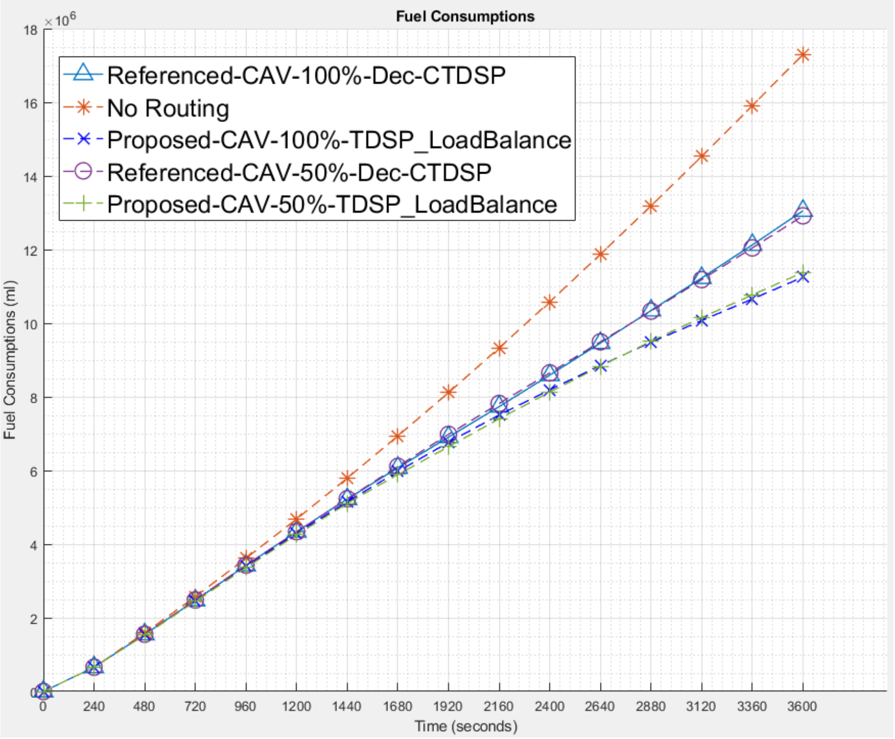


Figure 6.10. Fuel Consumption for Bandar Sunway Simulation

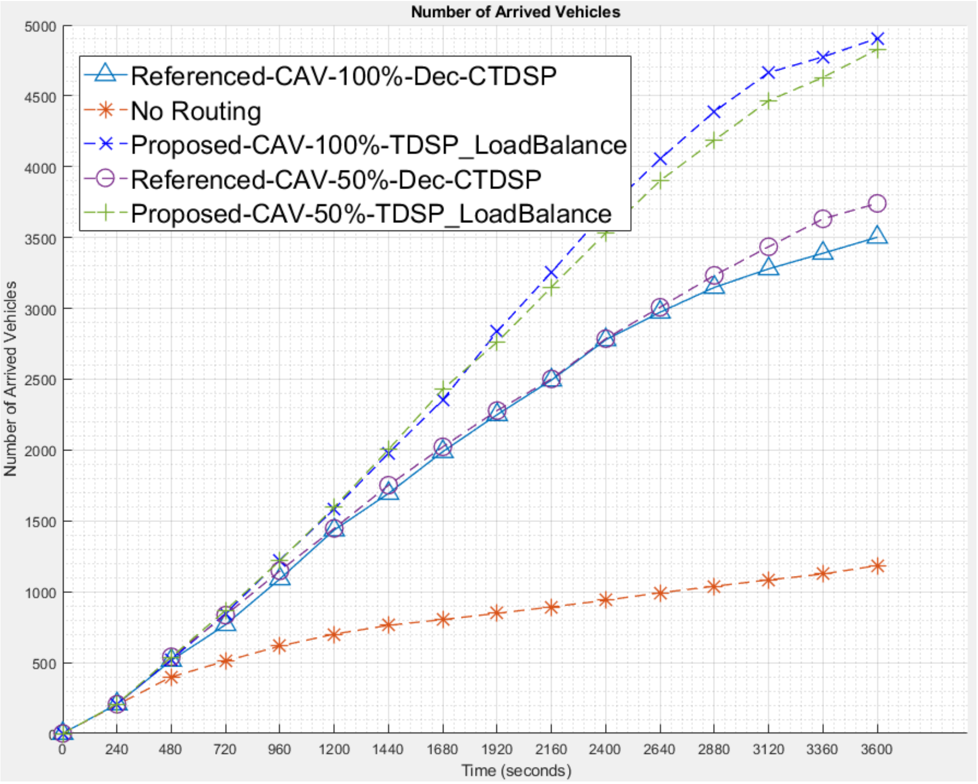


Figure 6.11. Number of Successful Arrivals for Bandar Sunway Simulation

6.4.3) Results Discussion

The values in the graphs are taken and averaged out across the entire simulation time, and TABLE 6.1 and

TABLE 6.2 shows the tabulated results for both Bukit Bintang and Sunway maps. Looking at the most important results, which is the number of successful arrivals, the proposed method shows an overall improvement of 4.4% and 31% in terms of successful arrivals for Bukit Bintang and Bandar Sunway, respectively, when comparing the 100% connected cases. Comparing the cases with 50% connected vehicles, we see a 21.4% and 23.8% improvement for Bukit Bintang and Bandar Sunway, respectively.

At the same time, looking at the number of congested roads, both cases of the proposed models are the lowest, with the proposed DCIB model at 100% connected vehicles is 2.4% and 14.1% lower for Bukit Bintang and Bandar Sunway, respectively. Comparing the case for 50% connected vehicles, the proposed model showed a 7.6% and 13.2% improvement for Bukit Bintang and Bandar Sunway, respectively.

This means that the proposed model has shown significant improvement in increasing the number of vehicles that has successfully arrived to their destination within the simulation time period, while also showing that it is more capable of controlling the number of roads to have a congestion threshold above 0.2 — in other words, minimized the number of roads which were deemed to be more than 80% congested — than the other models.

Looking at these two results, the proposed model overall performs significantly better for the Bandar Sunway map compared to the Bukit Bintang one, although both have shown that the proposed model has outperformed the referenced model by a significant margin, especially when taking into account the number of congested roads.

This discrepancy in performance between the two maps could be due to a few reasons.

i. The Bukit Bintang network was designed as a business district with an emphasis on shopping malls, hotels, and office blocks. Due to that, there may be fewer intersections connecting from one road to another compared to Bandar Sunway, which includes a few residential areas, allowing for more movement between areas.

ii. The Bukit Bintang map used is larger than the Bandar Sunway map, which may have led to longer trips and more congestion as a result.

iii. The parameters for the lower and upper bound of the $intersectionFactor$ for each map might require some additional calibration.

Looking at the overall improvement of the proposed model, especially when considering cases with 50% non-connected vehicles, as at the moment there are even fewer connected vehicles on the road in Malaysia, the proposed DCIB model has proven that reprioritizing the roads based on the number of intersections in an inverse manner provides an improvement to congested roads, successful arrivals, and travel time, and ultimately, reduction in fuel emissions and pollution.

TABLE 6.1

Tabulated Results for Bukit Bintang

Case	Congested Road	Mean Travel Time	CO2 Emission (E+10)	Fuel Emission (E+06)	Successful Arrivals
No Routing	371.63	1204.3	1.78	7.64	656.81
Reference: Dec-CTDSP 100% Connected	313.5	1039.5	1.56	6.72	1604.2
Reference: Dec-CTDSP 50% Connected	322.44	1100.5	1.60	6.89	1215.8
Proposed: DCIB 100% Connected	305.88	**1022.7**	**1.54**	**6.61**	**1675.3**
Proposed: DCIB 50% Connected	**297.81**	1051.3	1.54	6.63	1476.4

TABLE 6.2

Tabulated Results for Sunway

Case	Congested Road	Mean Travel Time	CO2 Emission (E+10)	Fuel Emission	Successful Arrivals
No Routing	229.94	1221.7	1.84	7.92	756.5
Reference: Dec-CTDSP 100% Connected	188.25	956.22	1.50	6.45	1969.7
Reference: Dec-CTDSP 50% Connected	188	954.37	1.50	6.45	2035.5
Proposed: DCIB 100% Connected	**161.63**	**869.1**	**1.40**	6.02	**2580.2**
Proposed: DCIB 50% Connected	163.19	872.15	1.40	**6.01**	2520.1

6.5) Chapter Summary - DeCentralized Intersection-Balancing (DCIB) Routing Model

Vehicle rerouting is the result of a line of ITS-related functions, such as data collection, data processing, and traffic prediction. Even so, this last step is just as crucial as any of the previous steps. Many studies focus only on the travel time and utilise various methods to efficiently reroute vehicles based on those criteria, with agent-based methods being the most widely used. However, by looking at only a snapshot of the road network, be it present or future traffic state, and prioritising the route with the shortest travel time, vehicles with no alternatives would have no choice but to clash with those that could have taken another road. This would result in a build-up of traffic on the selected road, which would end up even affecting other roads down the line.

Traffic Routing Optimisation

To investigate the significance of adjusting the routing priority of a navigational system based on the number of intersections along the roads, this chapter of the research has proposed a DeCentralized, Intersection-Balancing vehicle routing model, named DCIB, that considers the number of intersections on each road and assigns a weightage accordingly, with higher priority given to roads with minimal intersections and lower priority to those with more intersections. This would allow vehicles that need those intersections to use them instead of being blocked from using them due to other vehicles queuing up ahead of them. Not to mention, a similar rerouting model was used even for non-connected vehicles, which at the time of writing, represent the majority of vehicles on the road. The proposed model aimed to prove the significance of this parameter and, at the same time, introduce an efficient routing model for both connected and non-connected vehicles, thus providing a comprehensive routing model that is robust.

As traffic routing is an NP-hard problem, it is not feasible to determine the globally optimum solution, especially for a complex simulation, due to the limitation of time and resources. Hence, this thesis will compare the results with the referenced models. Looking at the results, whereby the proposed method shows an overall improvement of 4.4% and 31% in terms of successful arrivals for Bukit Bintang and Bandar Sunway, respectively, when comparing the 100% connected cases and a 2.4% and 14.1% improvement when comparing the number of congested roads for Bukit Bintang and Bandar Sunway respectively. For the case of 50% connected vehicles, there is a 21.4% and 23.8% increase in successful arrivals for the Bukit Bintang and Bandar Sunway maps, respectively, while also displaying a 7.6% and 13.2% improvement in their respective number of congested roads.

Through this, the proposed DCIB model has shown that by taking the number of intersections of a road into account, the performance of vehicle routing models can improve significantly. While obviously, the road network design plays a large part in how well the model does, as shown by the difference in performance between the Bandar Sunway and Bukit Bintang map, including this parameter in the model's routing consideration has been shown to improve the overall performance in a non-negligible manner. Another important point is that the proposed DCIB model works even when considering the existence of non-connected vehicles, which shows that this parameter, which considers the number of road intersections and works in a universal manner, can be incorporated in future traffic rerouting studies as well.

Chapter 7: Thesis Conclusion

Technological advancements and a more affluent economy have resulted in private vehicles being more accessible to the general public. This, coupled with the increasing popularity of e-hailing services, courier and logistics, and a growing population, has resulted in a surge of vehicles on the road. Urban development needs to take these factors into account when developing or upgrading urban districts. However, these efforts take time as well as will cause a disruption in traffic for a short period of time. Furthermore, urban development costs a lot of resources and is unable to be done in an adaptive manner. Instead, what is being done to manage traffic is what is known as Intelligent Transportation System (ITS).

ITS is a continuously evolving field whereby technologies and new research are constantly being studied in order to address the ever-growing issue of the increasing number of vehicles on the road, alongside the growth in the global population. This research has focused on four stages of the ITS functions leading to the rerouting stage for drivers utilising navigational software. These four stages include traffic analyses and modelling, missing traffic data imputation, traffic speed forecasting, and traffic rerouting.

An in-depth analysis of these topics revealed various shortcomings in existing literature, which then became the foundation for the research objective of this research. These problems were summarised in TABLE 1.1 and then addressed through the corresponding research objectives mentioned in Chapter 1.3, of which each research objective has its own sub-objectives. These objectives were split into chapters 4, 5, and 6. The aim of this research is to ultimately improve the existing ITS by developing new models and insights into traffic management that can be taken into account and incorporated into future studies.

The following subsections summarise the initial work done, followed by the key contributions of each of the research objectives and their respective results. The potential for future work is also discussed based on each objective's limitations and results.

7.1) Summary of Contributions

Throughout chapters 4 to 6, this research thesis has detailed the various proposed models and implementations to achieve the previously mentioned research objectives. Prior to that, some preliminary work was done in Chapter 3, which included traffic data collection, of which the traffic dataset used throughout the research was obtained via the HERE online traffic API [143] for the urban areas of Malaysia, namely Bukit Bintang, Bandar Sunway, and Damansara Utama. The calculation of the obtained traffic data and evaluation criteria were also explained.

Moving on to each research objective, first, the proposed MOD3D-PAT has proven to be a reliable and robust generalised model for traffic patterns based on the time of the day, allowing for further analyses to be done for urban traffic in Malaysia, with the traffic dataset obtained via an online traffic API, HERE Traffic API [143]. Comparing them to the main map, the Malaysian urban city, Bukit Bintang, the proposed model showed a very good approximation with the worst R-Squared of 0.93 and an RMSE of 0.035, and the best being an R-Squared value of 0.97 and RMSE of 0.024.

Secondly, an ensemble model combining the strengths of ARIMA, QRF and K-means clustering, and tensor factorization in order to quickly impute the missing data while maintaining a good model training method to overcome the heteroscedastic nature of the traffic data while being more accurate than the referenced model has shown that the second research objective has been achieved. The results of the proposed AQT model have shown to have the best performance with an RMSE of 0.3988, 0.473, 0.5306, 0.6617, and 0.837 and MAPE of 0.8263%, 0.9476%, 1.0711%, 1.365%, and 1.7319% for missing data rate of 10%, 30%, 50%, 70%, and 90% respectively. For the worst case of 90% missing data, the proposed model showed an improvement of 0.1725% MAPE than the referenced tensor factorization method, BATF [15], which is a 9% improvement from the referenced model's MAPE of 1.9044%. A similar improvement can be seen for RMSE, whereby the proposed model showed an improvement of 0.1277, which is a 13% improvement from the referenced model's RMSE of 0.9647.

Thirdly, by utilising a novel traffic data segmentation and clustering, an accurate long-term (day-long) traffic speed forecasting model has been proposed, named Clustered Augmented LSTM (CAL), and thus achieved the third research objective. The proposed CAL model has outperformed the referenced CNN-LSTM, W-CNN-LSTM, and their GRU counterparts by 1.42%-1.76%, and the GRU implementation of the proposed model by 0.46% for MAPE and 0.25-0.41 and 0.1 for RMSE respectively. This has proven that the pre-processing data methods and the proposed model have an impact towards an accurate long-term traffic prediction model. The potential of Gated Recurrent Units (GRU) was also looked into and has shown very promising results, although, for the case of the proposed CAL model, it seemed that the LSTM implementation performed slightly better on average, especially when considering their best cases.

Finally, the fourth research objective was achieved by implementing the intersection-based route weighting model, which considers the intersections available on each road in order to achieve a more efficient traffic routing model, which ultimately also leads to a reduction in harmful gas emissions. The proposed DCIB model shows an overall improvement of 4.4% and 31% in terms of successful arrivals for Bukit Bintang and Bandar Sunway, respectively, when comparing the 100% connected cases with the number of congested roads for both Bukit Bintang and Bandar Sunway showing and a 2.4% and 14.1% improvement respectively. For the case of 50% connected vehicles, there is a 21.4% and 23.8% increase in successful arrivals for the Bukit Bintang and Bandar Sunway maps, respectively, while also displaying a 7.6% and 13.2% improvement in their respective number of congested roads.

7.2) Future Work

Chapters 4 to 6 have introduced the different proposed traffic analyses and models, each showing encouraging results and interesting observations. However, throughout the course of the research, there are certain avenues for future work that have also been identified and is discussed below.

The proposed mathematical traffic model, MOD3D-PAT, takes into account the various time-based traffic parameters, such as the time of the peak or low traffic and the time of the day, to provide a good generalisation of traffic patterns in an urban setting. However, it was also shown that the fit of the model varies depending on the focus of the urban area, such as Sunway, with its focus on education. Future work could look into implementing additional parameters, such as the infrastructure surrounding the area of interest and observing how it affects the corresponding traffic patterns. This would in turn improve the

Currently, the research into long-term traffic predictions and short-term traffic predictions are done separately. However, a reliable long-term traffic prediction could possibly be used as another parameter for short-term traffic prediction, as mentioned by [24]. Future work could look into this and incorporate a long-term traffic prediction model as an additional feature to predict short-term traffic. This would also be useful for situations with limited traffic features, such as the case conducted in this research.

Finally, the rerouting model proposed has shown that the number of intersections along the road is a very important parameter that should be considered when rerouting traffic. Despite proving the significance of this parameter when compared to just looking at the route with the shortest travel time, further work can be done to look into implementing this parameter together with other traffic control implementations, such as a smart traffic light control system in place, to investigate the full extent of the contributions, this parameter has towards the overall performance towards reducing vehicles' travel times. Furthermore, the decentralized model used for the simulation can be improved upon by integrating more cooperation between vehicle clusters.

References

[1] R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban, “Missing Traffic Data Imputation for Artificial Intelligence in Intelligent Transportation Systems: Review of Methods, Limitations, and Challenges,” *IEEE Access*, vol. 11, pp. 34080–34093, 2023, doi: 10.1109/ACCESS.2023.3264216.

[2] R. K. C. Chan, J. M. Y. Lim, and R. Parthiban, “MOD3D-PAT - A novel modified 3rd degree polynomial approximation for modelling traffic congestion in urban areas,” *International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2021*, no. October, pp. 7–8, 2021, doi: 10.1109/ICECCME52200.2021.9591081.

[3] Z. Ning, J. Huang, and X. Wang, “Vehicular fog computing: Enabling real-time traffic management for smart cities,” *IEEE Wirel Commun*, vol. 26, no. 1, pp. 87–93, 2019, doi: 10.1109/MWC.2019.1700441.

[4] I. Rubin, A. Baiocchi, Y. Sunyoto, and I. Turcanu, “Traffic Management and Networking for Autonomous Vehicular Highway Systems,” *Ad Hoc Networks*, vol. 83, pp. 125–148, 2019, doi: 10.1016/j.adhoc.2018.08.018.

[5] P. Zhao and H. Hu, “Geographical patterns of traffic congestion in growing megacities: Big data analytics from Beijing,” *Cities*, vol. 92, no. March, pp. 164–174, 2019, doi: 10.1016/j.cities.2019.03.022.

[6] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, “Daily long-term traffic flow forecasting based on a deep neural network,” *Expert Syst Appl*, vol. 121, pp. 304–312, 2019, doi: 10.1016/j.eswa.2018.12.031.

[7] Z. Song, W. Feng, and W. Liu, “Interval prediction of short-term traffic speed with limited data input: Application of fuzzy-grey combined prediction model,” *Expert Syst Appl*, vol. 187, no. September 2021, p. 115878, 2022, doi: 10.1016/j.eswa.2021.115878.

[8] M. C. Ho, J. M. Y. Lim, K. L. Soon, and C. Y. Chong, “An improved pheromone-based vehicle rerouting system to reduce traffic congestion,” *Applied Soft Computing Journal*, vol. 84, p. 105702, 2019, doi: 10.1016/j.asoc.2019.105702.

[9] K. L. Soon, J. M. Y. Lim, and R. Parthiban, “Coordinated Traffic Light Control in Cooperative Green Vehicle Routing for Pheromone-based Multi-Agent Systems,” *Applied Soft Computing Journal*, vol. 81, p. 105486, 2019, doi: 10.1016/j.asoc.2019.105486.

[10] P. Dutta, S. Khatua, and S. Choudhury, “DB-Corouting: Density Based Coordinated Vehicle Rerouting in Smart Environment,” *International Journal of Intelligent Transportation Systems Research*, vol. 19, no. 3, pp. 539–556, 2021, doi: 10.1007/s13177-021-00261-6.

[11] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6. Springer Nature, pp. 422–440, Jun. 01, 2021. doi: 10.1038/s42254-021-00314-5.

[12] Md. T. Hossain and Md. K. Hasan, “Assessment of Traffic Congestion by Traffic Flow Analysis in Pabna Town,” *Http://Www.Sciencepublishinggroup.Com*, vol. 4, no. 3, p. 75, 2019, doi: 10.11648/J.AJTTE.20190403.11.

[13] D. Ma, X. Song, and P. Li, “Daily Traffic Flow Forecasting through a Contextual Convolutional Recurrent Neural Network Modeling Inter- And Intra-Day Traffic Patterns,” *IEEE Transactions*

on Intelligent Transportation Systems, vol. 22, no. 5, pp. 2627–2636, May 2021, doi: 10.1109/TITS.2020.2973279.

[14] H. Li, M. Li, X. Lin, F. He, and Y. Wang, “A spatiotemporal approach for traffic data imputation with complicated missing patterns,” *Transp Res Part C Emerg Technol*, vol. 119, no. November 2019, p. 102730, 2020, doi: 10.1016/j.trc.2020.102730.

[15] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, “Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model,” *Transp Res Part C Emerg Technol*, vol. 104, no. May, pp. 66–77, 2019, doi: 10.1016/j.trc.2019.03.003.

[16] N. Ranjan, S. Bhandari, H. P. Zhao, H. Kim, and P. Khan, “City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN,” *IEEE Access*, vol. 8, pp. 81606–81620, 2020, doi: 10.1109/ACCESS.2020.2991462.

[17] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, “A Spatial-Temporal Attention Approach for Traffic Prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4909–4918, 2021, doi: 10.1109/TITS.2020.2983651.

[18] F. Zhao, G. Q. Zeng, and K. Di Lu, “EnLSTM-WPEO: Short-term traffic flow prediction by ensemble LSTM, NNCT weight integration, and population extremal optimization,” *IEEE Trans Veh Technol*, vol. 69, no. 1, pp. 101–113, 2020, doi: 10.1109/TVT.2019.2952605.

[19] S. Fang, X. Pan, S. Xiang, and C. Pan, “Meta-MSNet: Meta-Learning based Multi-Source Data Fusion for Traffic Flow Prediction,” *IEEE Signal Process Lett*, vol. 28, pp. 1–1, 2020, doi: 10.1109/lsp.2020.3037527.

[20] S. Shahriari, M. Ghasri, S. A. Sisson, and T. Rashidi, “Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction,” *Transportmetrica A: Transport Science*, vol. 16, no. 3, pp. 1552–1573, 2020, doi: 10.1080/23249935.2020.1764662.

[21] Y. Li, S. Chai, Z. Ma, and G. Wang, “A Hybrid Deep Learning Framework for Long-Term Traffic Flow Prediction,” *IEEE Access*, vol. 9, pp. 11264–11271, 2021, doi: 10.1109/ACCESS.2021.3050836.

[22] Z. Wang, X. Su, and Z. Ding, “Long-Term Traffic Prediction Based on LSTM Encoder-Decoder Architecture,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6561–6571, 2021, doi: 10.1109/TITS.2020.2995546.

[23] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” *Sensors (Switzerland)*, vol. 17, no. 7, pp. 1–16, 2017, doi: 10.3390/s17071501.

[24] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, “Road Traffic Forecasting: Recent Advances and New Challenges,” *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018, doi: 10.1109/MITS.2018.2806634.

[25] Z. Bartlett, L. Han, T. T. Nguyen, and P. Johnson, “A novel online dynamic temporal context neural network framework for the prediction of road traffic flow,” *IEEE Access*, vol. 7, pp. 153533–153541, 2019, doi: 10.1109/ACCESS.2019.2943028.

[26] “Grab.” https://www.grab.com/my/ (accessed May 29, 2023).

[27] J. Song, Y. Wu, Z. Xu, and X. Lin, “Research on car-following model based on SUMO,” *Proceedings of 2014 IEEE 7th International Conference on Advanced Infocomm Technology, IEEE/ICAIT 2014*, no. January, pp. 47–55, 2015, doi: 10.1109/ICAIT.2014.7019528.

[28] D. Krajzewicz and C. Rossel, “Simulation of urban mobility (SUMO),” *Centre for Applied Informatics (ZAIK) and the Institute of …*, pp. 1–35, 2007.

[29] P. A. Lopez *et al.*, “Microscopic Traffic Simulation using SUMO,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-Novem, pp. 2575–2582, 2018, doi: 10.1109/ITSC.2018.8569938.

[30] A. Horni, K. Nagel, and K. W. Axhausen, *The Multi-Agent Transport Simulation Title of Book : The Multi-Agent Transport Simulation MATSim*. 2016.

[31] “Repast Simphony.” https://repast.github.io/ (accessed Jul. 13, 2020).

[32] M. Adnan, F. C. Pereira, and C. L. Azevedo, “SimMobility : A Multi-Scale Integrated Agent-based Simulation Platform,” no. July, 2016.

[33] U. Wilensky, “NetLogo.” 1999.

[34] “SimMobility Github Wiki.” https://github.com/smart-fm/simmobility-prod/wiki

[35] A. O. Diallo, G. Lozenguez, A. Doniec, and R. Mandiau, “Comparative evaluation of road traffic simulators based on modeler’s specifications: An application to intermodal mobility behaviors,” *ICAART 2021 - Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, vol. 1, no. Icaart, pp. 265–272, 2021, doi: 10.5220/0010238302650272.

[36] D. Ziemke, L. Knapen, and K. Nagel, “Expanding the analysis scope of a MATSim transport simulation by integrating the FEATHERS activity-based demand model,” *Procedia Comput Sci*, vol. 184, no. 2019, pp. 753–760, 2021, doi: 10.1016/j.procs.2021.04.022.

[37] K. L. Soon, J. M. Y. Lim, R. Parthiban, and M. C. Ho, “Proactive Eco-friendly Pheromone-based Green Vehicle Routing for multi-agent systems,” *Expert Syst Appl*, vol. 121, pp. 324–337, 2019, doi: 10.1016/j.eswa.2018.12.026.

[38] S. Akhter, N. Ahsan, S. Jafor, S. Quaderi, and A. Al Forhad, “A SUMO Based Simulation Framework for Intelligent Traffic Management System,” no. June, 2020, doi: 10.18178/jtle.8.1.1-5.

[39] G. Yan, J. Lin, D. Rawat, and J. C. Enyart, “The Role of Network and Mobility Simulators in Evaluating Vehicular Networks,” *Communications in Computer and Information Science*, vol. 135, no. PART 2, pp. 706–712, 2011, doi: 10.1007/978-3-642-18134-4_112.

[40] C. Nawej, P. Owolawi, and T. Walingo, “Design and Simulation of VANETs Testbed Using OpenStreetMap, SUMO, and NS-2,” *2021 IEEE 6th International Conference on Computer and Communication Systems, ICCCS 2021*, pp. 582–587, 2021, doi: 10.1109/ICCCS52626.2021.9449197.

[41] OpenSim Ltd., “OMNET++ Introduction.” https://omnetpp.org/intro/

[42] C. Sommer, R. German, and F. Dressler, “Bidirectionally coupled network and road simulation for improved IVC analysis,” *IEEE Trans Mob Comput*, vol. 10, no. 1, pp. 3–15, 2011, doi: 10.1109/TMC.2010.133.

[43] S. Rani, “Infrastructure based Routing Protocols in Vehicular Ad Hoc Network: A Review,” pp. 476–481, 2017.

[44] J. S. Weber, M. Neves, and T. Ferreto, “VANET simulators: an updated review,” *Journal of the Brazilian Computer Society*, vol. 27, no. 1, 2021, doi: 10.1186/s13173-021-00113-x.

[45] I. A. Aljabry and G. A. Al-Suhail, “A Survey on Network Simulators for Vehicular Ad-hoc Networks (VANETS),” *Int J Comput Appl*, vol. 174, no. 11, pp. 1–9, 2021, doi: 10.5120/ijca2021920979.

[46] OMNeT+, “OMNeT++ Discrete Event Simulator”.

[47] M. K. Patel, “Article: Comparative Study of Vehicular Ad-hoc Network Mobility Models and Simulators,” *Int J Comput Appl*, vol. 47, no. 6, pp. 38–43, Jun. 2012.

[48] “Veins.” https://veins.car2x.org/

[49] M. Piórkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, “TraNS,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 1, pp. 31–33, 2008, doi: 10.1145/1374512.1374522.

[50] F. Kage, “VSimRTI : Vehicle-2-X Simulation Runtime Infrastructure,” pp. 1–90, 2012.

[51] V. Kumar *et al.*, “ITETRIS: Adaptation of ITS technologies for large scale integrated simulation,” *IEEE Vehicular Technology Conference*, pp. 9–13, 2010, doi: 10.1109/VETECS.2010.5494182.

[52] B. d. Greenshields, J. r. Bibbins, W. s. Channing, and H. h. Miller, “A study of traffic capacity,” *Highway Research Board proceedings*, vol. 1935, p., 1935, [Online]. Available: http://dx.doi.org/

[53] M. Umer *et al.*, “BEHAVIORAL ANALYSIS OF LWR MODEL UNDER DIFFERENT EQUILIIBRIUM VELOCITY DISTRIBUTIONS,” vol. 71, pp. 194–199, 2019.

[54] Z. Xicheng, P. Yusheng, X. Qicheng, and H. Xinyu, “Study on nonlinear car-following model based on opening community’s road,” *Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019*, pp. 5390–5394, 2019, doi: 10.1109/CCDC.2019.8832662.

[55] P. H. Gunawan and M. Ardi Rizmaldi, “Approximation of velocity-density function for traffic flow model with obstacle problem in jalan merdeka bandung,” *2019 7th International Conference on Information and Communication Technology, ICoICT 2019*, pp. 1–6, 2019, doi: 10.1109/ICoICT.2019.8835368.

[56] O. A. Rosas-Jaimes, L. A. Q. Téllez, and G. F. Anaya, “Polynomial Approach and Non-linear Analysis for a Traffic Fundamental Diagram,” *PROMET - Traffic&Transportation*, vol. 28, no. 4, pp. 321–329, 2016, doi: 10.7307/ptt.v28i4.1965.

[57] K. L. Soon, J. M. Y. Lim, and R. Parthiban, “Extended pheromone-based short-term traffic forecasting models for vehicular systems,” *Eng Appl Artif Intell*, vol. 82, no. February 2018, pp. 60–75, 2019, doi: 10.1016/j.engappai.2019.03.017.

[58] F. Sun, A. Dubey, and J. White, “DxNAT - Deep neural networks for explaining non-recurring traffic congestion,” *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, vol. 2018-Janua, pp. 2141–2150, 2018, doi: 10.1109/BigData.2017.8258162.

References

[59] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, “Forecasting traffic congestion using ARIMA modeling,” *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, pp. 1227–1232, 2019, doi: 10.1109/IWCMC.2019.8766698.

[60] N. Karmitsa, S. Taheri, A. Bagirov, and P. Makinen, “Missing Value Imputation via Clusterwise Linear Regression,” *IEEE Trans Knowl Data Eng*, vol. 4347, no. c, 2020, doi: 10.1109/TKDE.2020.3001694.

[61] T. Yokota, B. Erem, S. Guler, S. K. Warfield, and H. Hontani, “Missing Slice Recovery for Tensors Using a Low-Rank Model in Embedded Space,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, no. 26108003, pp. 8251–8259, 2018, doi: 10.1109/CVPR.2018.00861.

[62] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, “Scalable tensor factorizations for incomplete data,” *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011, doi: 10.1016/j.chemolab.2010.08.004.

[63] X. Chen, Z. He, and L. Sun, “A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation,” *Transp Res Part C Emerg Technol*, vol. 98, no. June 2018, pp. 73–84, 2019, doi: 10.1016/j.trc.2018.11.003.

[64] C. Gong and Y. Zhang, “Urban Traffic Data Imputation with Detrending and Tensor Decomposition,” *IEEE Access*, vol. 8, pp. 11124–11137, 2020, doi: 10.1109/ACCESS.2020.2964299.

[65] A. Ben Said and A. Erradi, “Spatiotemporal Tensor Completion for Improved Urban Traffic Imputation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6836–6849, 2021, doi: 10.1109/TITS.2021.3062999.

[66] Z. Cui, R. Ke, Z. Pu, and Y. Wang, “Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values,” *Transp Res Part C Emerg Technol*, vol. 118, no. May, p. 102674, 2020, doi: 10.1016/j.trc.2020.102674.

[67] L. Li, B. Du, Y. Wang, L. Qin, and H. Tan, “Estimation of missing values in heterogeneous traffic data: Application of multimodal deep learning model,” *Knowl Based Syst*, vol. 194, p. 105592, 2020, doi: 10.1016/j.knosys.2020.105592.

[68] A. Liu, C. Li, W. Yue, and X. Zhou, “Real-Time Traffic Prediction: A Novel Imputation Optimization Algorithm with Missing Data,” *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*, vol. d, 2018, doi: 10.1109/GLOCOM.2018.8647193.

[69] E. Joelianto, M. F. Fathurrahman, H. Y. Sutarto, I. Semanjski, A. Putri, and S. Gautama, “Analysis of Spatiotemporal Data Imputation Methods for Traffic Flow Data in Urban Networks,” *ISPRS Int J Geoinf*, vol. 11, no. 5, 2022, doi: 10.3390/ijgi11050310.

[70] J. Xing, W. Wu, Q. Cheng, and R. Liu, “Traffic state estimation of urban road networks by multi-source data fusion: Review and new insights,” *Physica A: Statistical Mechanics and its Applications*, vol. 595, p. 127079, 2022, doi: 10.1016/j.physa.2022.127079.

[71] T. Sun, S. Zhu, R. Hao, and B. Sun, “Traffic Missing Data Imputation : A Selective Overview of Temporal Theories and Algorithms,” 2022.

[72] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 1, pp. 4–24, 2021, doi: 10.1109/TNNLS.2020.2978386.

[73] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, “LSTM-based traffic flow prediction with missing data,” *Neurocomputing*, vol. 318, pp. 297–305, 2018, doi: 10.1016/j.neucom.2018.08.067.

[74] C. Velasco-Gallego and I. Lazakis, “Real-time data-driven missing data imputation for short-term sensor data of marine systems. A comparative study,” *Ocean Engineering*, vol. 218, no. July, p. 108261, 2020, doi: 10.1016/j.oceaneng.2020.108261.

[75] K. Henrickson, Y. Zou, and Y. Wang, “Flexible and robust method for missing loop detector data imputation,” *Transp Res Rec*, vol. 2527, no. 2527, pp. 29–36, 2015, doi: 10.3141/2527-04.

[76] M. A. Shafique, “Imputing Missing Data in Hourly Traffic Counts,” *Sensors*, vol. 22, no. 24, 2022, doi: 10.3390/s22249876.

[77] Christopher. Bishop and Michael. Tipping, “Probabilistic Principal Component Analysis Author (s): Michael E . Tipping and Christopher M . Bishop Source : Journal of the Royal Statistical Society . Series B (Statistical Methodology), Vol . 61 , Published by : Wiley for the Royal Statistical Soc,” vol. 61, no. 3, pp. 611–622, 1999.

[78] M. F. Fathurrahman, H. Y. Sutarto, and I. Semanjski, “Urban Network Traffic Analysis, Data Imputation, and Flow Prediction based on Probabilistic PCA Model of Traffic Volume Data,” *Proceedings - 2021 8th International Conference on Advanced Informatics: Concepts, Theory, and Application, ICAICTA 2021*, vol. D, 2021, doi: 10.1109/ICAICTA53211.2021.9640284.

[79] H. Hegde, N. Shimpi, A. Panny, I. Glurich, P. Christie, and A. Acharya, “MICE vs PPCA: Missing data imputation in healthcare,” *Inform Med Unlocked*, vol. 17, no. September, p. 100275, 2019, doi: 10.1016/j.imu.2019.100275.

[80] W. C. Lin and C. F. Tsai, “Missing value imputation: a review and analysis of the literature (2006–2017),” *Artificial Intelligence Review*, vol. 53, no. 2. pp. 1487–1509, 2020. doi: 10.1007/s10462-019-09709-4.

[81] M. K. Hasan, M. A. Alam, S. Roy, A. Dutta, M. T. Jawad, and S. Das, “Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021),” *Inform Med Unlocked*, vol. 27, 2021, doi: 10.1016/j.imu.2021.100799.

[82] J. Liu, G. P. Ong, and X. Chen, “GraphSAGE-Based Traffic Speed Forecasting for Segment Network with Sparse Data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1755–1766, 2022, doi: 10.1109/TITS.2020.3026025.

[83] M. Nouri, M. Reisi-Gahrooei, and M. Ilbeigi, “A Method for Granular Traffic Data Imputation Based on PARATUCK2,” *Transportation Research Record: Journal of the Transportation Research Board*, p. 036119812210892, 2022, doi: 10.1177/03611981221089298.

[84] M. Lei, A. Labbe, Y. Wu, and L. Sun, “Bayesian Kernelized Matrix Factorization for Spatiotemporal Traffic Data Imputation and Kriging,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2022, doi: 10.1109/TITS.2022.3161792.

[85] A. Baggag *et al.*, “Learning Spatiotemporal Latent Factors of Traffic via Regularized Tensor Factorization: Imputing Missing Values and Forecasting,” *IEEE Trans Knowl Data Eng*, vol. 33, no. 6, pp. 2573–2587, 2021, doi: 10.1109/TKDE.2019.2954868.

[86] X. Jia, X. Dong, M. Chen, and X. Yu, “Missing data imputation for traffic congestion data based on joint matrix factorization,” *Knowl Based Syst*, vol. 225, p. 107114, 2021, doi: 10.1016/j.knosys.2021.107114.

[87] M. Bhanu, J. Mendes-moreira, and J. Chandra, “Embedding Traffic Network Characteristics Using Tensor for Improved Traffic Prediction,” vol. 22, no. 6, pp. 1–13, 2020.

[88] T. Nie, G. Qin, and J. Sun, “ Truncated tensor Schatten p -norm based approach for spatiotemporal traffic data imputation with complicated missing patterns ,” *Transp Res Part C Emerg Technol*, vol. 141, no. January, p. 103737, 2022, doi: 10.1016/j.trc.2022.103737.

[89] H. Yu, S. Liu, H. Jiang, and Y. Ren, “Vehicle trajectory reconstruction using a tensor-based individual travel time matching method,” *Proceedings 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence, DTPI 2021*, pp. 184–187, 2021, doi: 10.1109/DTPI52967.2021.9540140.

[90] J. Li, L. Xu, R. Li, P. Wu, and Z. Huang, “Deep spatial-temporal bi-directional residual optimisation based on tensor decomposition for traffic data imputation on urban road network,” *Applied Intelligence*, no. 2021, pp. 11363–11381, 2022, doi: 10.1007/s10489-021-03060-4.

[91] X. Chen, Y. Chen, N. Saunier, and L. Sun, “Scalable low-rank tensor learning for spatiotemporal traffic data imputation,” *Transp Res Part C Emerg Technol*, vol. 129, no. August 2020, p. 103226, 2021, doi: 10.1016/j.trc.2021.103226.

[92] Y. Liang, Z. Zhao, and L. Sun, “Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns,” *Transportation Research Part C*, vol. 143, no. September 2021, p. 103826, 2022, doi: 10.1016/j.trc.2022.103826.

[93] Y. Yuan, Y. Zhang, B. Wang, Y. Peng, Y. Hu, and B. Yin, “STGAN: Spatio-Temporal Generative Adversarial Network for Traffic Data Imputation,” *IEEE Trans Big Data*, vol. 7790, no. c, pp. 1–13, 2022, doi: 10.1109/TBDATA.2022.3154097.

[94] W. Mcculloch and W. Pitts, “A Logical Calculus of Ideas Immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics*, vol. 5, pp. 127–147, 1943.

[95] I. Goodfellow *et al.*, “Generative adversarial networks,” *Commun ACM*, vol. 63, no. 11, pp. 139–144, 2020, doi: 10.1145/3422622.

[96] W. Zhang, P. Zhang, Y. Yu, X. Li, S. A. Biancardo, and J. Zhang, “Missing Data Repairs for Traffic Flow With Self-Attention Generative Adversarial Imputation Net,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7919–7930, 2021, doi: 10.1109/TITS.2021.3074564.

[97] B. Yang, Y. Kang, Y. Y. Yuan, X. Huang, and H. Li, “ST-LBAGAN: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation,” *Knowl Based Syst*, vol. 215, p. 106705, 2021, doi: 10.1016/j.knosys.2020.106705.

[98] B. Yang, Y. Kang, Y. Yuan, H. Li, and F. Wang, “ST-FVGAN: filling series traffic missing values with generative adversarial network,” *Transportation Letters*, vol. 14, no. 4, pp. 407–415, 2022, doi: 10.1080/19427867.2021.1879624.

[99] A. Kazemi and H. Meidani, “IGANI: Iterative generative adversarial networks for imputation with application to traffic data,” *IEEE Access*, vol. 9, pp. 112966–112977, 2021, doi: 10.1109/ACCESS.2021.3103456.

References

[100] D. Xu, H. Peng, C. Wei, X. Shang, and H. Li, "Traffic State Data Imputation: An Efficient Generating Method Based on the Graph Aggregator," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13084–13093, 2021, doi: 10.1109/TITS.2021.3119638.

[101] A. Borji, "Pros and cons of GAN evaluation measures: New developments," *Computer Vision and Image Understanding*, vol. 215, Jan. 2022, doi: 10.1016/j.cviu.2021.103329.

[102] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 1, pp. 4–24, 2021, doi: 10.1109/TNNLS.2020.2978386.

[103] Y. Chen and X. Michael, "A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction," *Transportation Research Part C*, vol. 143, no. August, p. 103820, 2022, doi: 10.1016/j.trc.2022.103820.

[104] X. Yao, Y. Gao, D. Zhu, E. Manley, J. Wang, and Y. Liu, "Spatial Origin-Destination Flow Imputation Using Graph Convolutional Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7474–7484, 2021, doi: 10.1109/TITS.2020.3003310.

[105] W. Liang *et al.*, "Spatial-Temporal Aware Inductive Graph Neural Network for C-ITS Data Recovery," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2022, doi: 10.1109/TITS.2022.3156266.

[106] P. Wang, T. Zhang, Y. Zheng, and T. Hu, "A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation," *International Journal of Geographical Information Science*, vol. 36, no. 6, pp. 1231–1257, 2022, doi: 10.1080/13658816.2022.2032081.

[107] R. Liu, Y. Kan, S. Zhao, B. Cheng, Z. Ma, and W. Wu, "Turning traffic volume imputation for persistent missing patterns with GNNs," *Applied Intelligence*, vol. im, 2022, doi: 10.1007/s10489-022-03568-3.

[108] M. Gori, G. Monfardini, and F. Scarselli, "A new model for Learning in Graph domains," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, pp. 729–734, 2005, doi: 10.1109/IJCNN.2005.1555942.

[109] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1–14, 2017.

[110] T. Zhang, D. gan Zhang, H. ran Yan, J. ning Qiu, and J. xin Gao, "A new method of data missing estimation with FNN-based tensor heterogeneous ensemble learning for internet of vehicle," *Neurocomputing*, vol. 420, pp. 98–110, 2021, doi: 10.1016/j.neucom.2020.09.042.

[111] H. Dong, F. Ding, H. Tan, and H. Zhang, "Laplacian integration of graph convolutional network with tensor completion for traffic prediction with missing data in inter-city highway network," *Physica A: Statistical Mechanics and its Applications*, vol. 586, p. 126474, 2022, doi: 10.1016/j.physa.2021.126474.

[112] P. Wang, T. Hu, F. Gao, R. Wu, W. Guo, and X. Zhu, "A Hybrid Data-Driven Framework for Spatiotemporal Traffic Flow Data Imputation," *IEEE Internet Things J*, vol. 9, no. 17, pp. 16343–16352, 2022, doi: 10.1109/JIOT.2022.3151238.

[113] G. Liu and J. Wang, “Dendrite Net: A White-Box Module for Classification, Regression, and System Identification,” *IEEE Trans Cybern*, no. December, 2021, doi: 10.1109/TCYB.2021.3124328.

[114] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, “Deep Learning on Traffic Prediction: Methods, Analysis and Future Directions,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2021, doi: 10.1109/TITS.2021.3054840.

[115] A. Cecaj, M. Lippi, M. Mamei, and F. Zambonelli, “Comparing deep learning and statistical methods in forecasting crowd distribution from aggregated mobile phone data,” *Applied Sciences (Switzerland)*, vol. 10, no. 18, 2020, doi: 10.3390/APP10186580.

[116] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, “Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2019, doi: 10.1109/TITS.2018.2854913.

[117] K. Lee, M. Eo, E. Jung, Y. Yoon, and W. Rhee, “Short-Term Traffic Prediction with Deep Neural Networks: A Survey,” *IEEE Access*, vol. 9, pp. 54739–54756, 2021, doi: 10.1109/ACCESS.2021.3071174.

[118] H. Yuan and G. Li, “A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation,” *Data Sci Eng*, vol. 6, no. 1, pp. 63–85, 2021, doi: 10.1007/s41019-020-00151-z.

[119] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester, Eds., in Proceedings of Machine Learning Research, vol. 28. Atlanta, Georgia, USA: PMLR, 2013, pp. 1310–1318.

[120] N. K. Manaswi, “RNN and LSTM,” in *Deep Learning with Applications Using Python*, 2018, pp. 115–126. doi: 10.1007/978-1-4842-3516-4.

[121] S. Du, T. Li, Y. Yang, and S. J. Horng, “Multivariate time series forecasting via attention-based encoder–decoder framework,” *Neurocomputing*, vol. 388, pp. 269–279, 2020, doi: 10.1016/j.neucom.2019.12.118.

[122] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, “TrafficGAN: Network-Scale Deep Traffic Prediction with Generative Adversarial Nets,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 219–230, 2021, doi: 10.1109/TITS.2019.2955794.

[123] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, “Traffic speed prediction under non-recurrent congestion: based on lstm method and beidou navigation satellite system data,” *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 2, pp. 70–81, 2019, doi: 10.1109/MITS.2019.2903431.

[124] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” *Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC 2016*, no. November 2016, pp. 324–328, 2017, doi: 10.1109/YAC.2016.7804912.

[125] R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Applying deep learning approaches for network traffic prediction,” *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, vol. 2017-January, pp. 2353–2358, 2017, doi: 10.1109/ICACCI.2017.8126198.

[126] K. Zor and K. Bulus, "A benchmark of GRU and LSTM networks for short-term electric load forecasting," *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2021*, pp. 598–602, 2021, doi: 10.1109/3ICT53449.2021.9581373.

[127] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020, doi: 10.1109/TITS.2019.2950416.

[128] L. Zhao *et al.*, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020, doi: 10.1109/TITS.2019.2935152.

[129] M. Cao, V. O. K. Li, and V. W. S. Chan, "A CNN-LSTM Model for Traffic Speed Prediction," *IEEE Vehicular Technology Conference*, vol. 2020-May, pp. 1–5, 2020, doi: 10.1109/VTC2020-Spring48590.2020.9129440.

[130] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "TrafficGAN: Network-Scale Deep Traffic Prediction with Generative Adversarial Nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 219–230, 2021, doi: 10.1109/TITS.2019.2955794.

[131] S. B. Sarathi Barma, J. Dutta, and A. Mukherjee, "A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem," *Decision Making: Applications in Management and Engineering*, vol. 2, no. 2, pp. 112–125, 2019, doi: 10.31181/dmame1902089b.

[132] T. H. Nguyen and J. J. Jung, "Ant colony optimization-based traffic routing with intersection negotiation for connected vehicles," *Appl Soft Comput*, vol. 112, Nov. 2021, doi: 10.1016/j.asoc.2021.107828.

[133] Z. Shou, X. Chen, Y. Fu, and X. Di, "Multi-agent reinforcement learning for Markov routing games: A new modeling paradigm for dynamic traffic assignment," *Transp Res Part C Emerg Technol*, vol. 137, Apr. 2022, doi: 10.1016/j.trc.2022.103560.

[134] A. Mostafizi, C. Koll, and H. Wang, "A Decentralized and Coordinated Routing Algorithm for Connected and Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, no. 4, pp. 1–13, 2021, doi: 10.1109/TITS.2021.3105057.

[135] S. H. Lin and T. Y. Ho, "Autonomous vehicle routing in multiple intersections," in *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, Institute of Electrical and Electronics Engineers Inc., Jan. 2019, pp. 341–346. doi: 10.1145/3287624.3287723.

[136] A. Bilgram *et al.*, "Online and Proactive Vehicle Rerouting with Uppaal Stratego," *Transportation Research Record: Journal of the Transportation Research Board*, p. 036119812110003, 2021, doi: 10.1177/03611981211000348.

[137] T. D. Tambunan, A. B. Suksmono, I. J. M. Edward, and R. Mulyawan, "Quantum Annealing for Vehicle Routing Problem with weighted Segment," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.13469

[138] D-Wave Systems Inc., "QPU Solver Datasheet." https://docs.dwavesys.com/docs/latest/doc_qpu.html (accessed May 13, 2023).

[139] H. Guo, W. Sheng, C. Gao, and Y. Jin, "DRL Router: Distributional Reinforcement Learning-Based Router for Reliable Shortest Path Problems," *IEEE Intelligent Transportation Systems Magazine*, 2023, doi: 10.1109/MITS.2023.3265309.

[140] Z. Zhao and Y. Liang, "A deep inverse reinforcement learning approach to route choice modeling with context-dependent rewards," *Transp Res Part C Emerg Technol*, vol. 149, Apr. 2023, doi: 10.1016/j.trc.2023.104079.

[141] W. Ma, R. Hao, C. Yu, T. Sun, and B. van Arem, "Managing connected and automated vehicles with flexible routing at 'lane-allocation-free'' intersections,'" Aug. 2020, doi: 10.1016/j.trc.2023.104152.

[142] T. Giovannelli and L. N. Vicente, "An integrated assignment, routing, and speed model for roadway mobility and transportation with environmental, efficiency, and service goals," *Transp Res Part C Emerg Technol*, vol. 152, Jul. 2023, doi: 10.1016/j.trc.2023.104144.

[143] HERE, "What Is the Traffic API? - Traffic API - HERE Developer." https://developer.here.com/documentation/traffic/topics/what-is.html (accessed Sep. 04, 2019).

[144] Google, "Map and Tile Coordinates," 2019. https://developers.google.com/maps/documentation/javascript/coordinates (accessed Sep. 04, 2019).

[145] R. K. C. Chan, J. M. Y. Lim, and R. Parthiban, "A neural network approach for traffic prediction and routing with missing data imputation for intelligent transportation system," *Expert Syst Appl*, vol. 171, Jun. 2021, doi: 10.1016/j.eswa.2021.114573.

[146] J. D. Regehr and H. Hernandez-vega, "Traffic Pattern Groups Based on Hourly Traffic Variations in Urban Areas," no. March, 2017.

[147] Y. Zhang and Y. Zhang, "A Comparative Study of Three Multivariate Short-Term Freeway Traffic Flow Forecasting Methods With Missing Data," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 20, no. 3, pp. 205–218, 2016, doi: 10.1080/15472450.2016.1147813.

[148] M. Chon, J. M. Lim, K. Lun, and C. Yong, "An improved pheromone-based vehicle rerouting system to reduce traffic congestion," *Applied Soft Computing Journal*, vol. 84, p. 105702, 2019, doi: 10.1016/j.asoc.2019.105702.

[149] H. Rakha and B. Crowther, "Comparison of Greenshields, Pipes, and Van Aerde car-following and traffic stream models," *Transp Res Rec*, no. 1802, pp. 248–262, 2002, doi: 10.3141/1802-28.

[150] S. Chakravarty, H. Demirhan, and F. Baser, "Fuzzy regression functions with a noise cluster and the impact of outliers on mainstream machine learning methods in the regression setting," *Applied Soft Computing Journal*, vol. 96, p. 106535, 2020, doi: 10.1016/j.asoc.2020.106535.

[151] F. Rodrigues and F. C. Pereira, "Beyond Expectation: Deep Joint Mean and Quantile Regression for Spatiotemporal Problems," *IEEE Trans Neural Netw Learn Syst*, vol. 31, no. 12, pp. 5377–5389, 2020, doi: 10.1109/TNNLS.2020.2966745.

[152] R. Mushtaq, "Augmented Dickey Fuller Test," *Econometrics: Mathematical Methods \& Programming eJournal*, 2011.

References

[153] K. Sikorski, "Bisection is optimal," *Numer Math (Heidelb)*, vol. 40, no. 1, pp. 111–117, 1982, doi: 10.1007/BF01459080.

[154] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J Comput Appl Math*, vol. 20, no. C, pp. 53–65, 1987, doi: 10.1016/0377-0427(87)90125-7.

[155] N. Polson and V. Sokolov, "Deep Learning for Short-Term Traffic Flow Prediction," pp. 1–29, 2016, doi: 10.1016/j.trc.2017.02.024.

[156] A. Belhadi, Y. Djenouri, D. Djenouri, and J. C. W. Lin, "A recurrent neural network for urban long-term traffic flow forecasting," *Applied Intelligence*, vol. 50, no. 10, pp. 3252–3265, 2020, doi: 10.1007/s10489-020-01716-1.

[157] K. Bandara, H. Hewamalage, Y. H. Liu, Y. Kang, and C. Bergmeir, "Improving the accuracy of global forecasting models using time series data augmentation," *Pattern Recognit*, vol. 120, p. 108148, 2021, doi: 10.1016/j.patcog.2021.108148.

[158] B. K. Iwana and S. Uchida, *An empirical survey of data augmentation for time series classification with neural networks*, vol. 16, no. 7 July. 2021. doi: 10.1371/journal.pone.0254841.

[159] P. Sun, N. Aljeri, and A. Boukerche, "A Fast Vehicular Traffic Flow Prediction Scheme Based on Fourier and Wavelet Analysis," *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*, 2018, doi: 10.1109/GLOCOM.2018.8647731.

[160] M. Galeso, *Waze: An Easy Guide to the Best Features*, 1st ed. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2016.

[161] Google, "Google Maps Platform | Google Developers." https://developers.google.com/maps/documentation/ (accessed Sep. 04, 2019).

[162] L. Z. Ladeira, A. M. De Souza, T. H. Silva, R. W. Pazzi, and L. A. Villas, "PONCHE: Personalized and context-aware vehicle rerouting service," *IEEE International Conference on Cloud Computing, CLOUD*, vol. 2020-Octob, pp. 211–218, 2020, doi: 10.1109/CLOUD49709.2020.00040.

[163] S. Kamishetty, S. Vadlamannati, and P. Paruchuri, "Towards a better management of urban traffic pollution using a Pareto max flow approach," *Transp Res D Transp Environ*, vol. 79, no. January, p. 102194, 2020, doi: 10.1016/j.trd.2019.11.023.

[164] A. M. Altabeeb, A. M. Mohsen, L. Abualigah, and A. Ghallab, "Solving capacitated vehicle routing problem using cooperative firefly algorithm," *Appl Soft Comput*, vol. 108, Sep. 2021, doi: 10.1016/j.asoc.2021.107403.

[165] O. S. Map, "Open Street Map." https://www.openstreetmap.org/ (accessed Sep. 03, 2019).

[166] "SUMO - Summary Output." https://sumo.dlr.de/docs/Simulation/Output/Summary.html (accessed May 13, 2023).

[167] "SUMO - Emissions." https://sumo.dlr.de/docs/Tools/Emissions.html (accessed May 14, 2023).

[168] I. o. T. S. DLR, "TraCI." http://sumo.dlr.de/wiki/TraCI#TraCI_Commands

[169] A. Acosta, "'TraCI4MATLAB,.'" https://www.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab