Enhancing Traffic Flow Prediction using Real Time Data with deep learning Techniques

MSc Research Project
Data Analytics

Nihad Kazi
Student ID:22166009

School of Computing
National College of Ireland

Supervisor: Abdul Qayum

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Nihad Kazi
Student ID:	22166009
Programme:	MSc. Data Analytics
Year:	2023
Module:	Research in Project
Supervisor:	Abdul Qayum
Submission Due Date:	31-01-2024
Project Title:	Enhancing Traffic Flow Prediction using Real Time Data with deep learning Techniques
Word Count:	6077
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other authors written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Nihad Kazi
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Traffic Flow Prediction using Real Time Data with deep learning Techniques

Nihad Kazi
22166009

Abstract

This study focuses on the implementation and comparative analysis of traffic patterns using deep learning techniques namely LSTM and GRU using two distinct datasets: traffic sensor data and traffic regional data. The integration of real time data is pivotal in research, as it aims to enhance the accuracy of traffic flow predictions. The results delves into implementation of data transformation techniques, exploratory data analysis, training of model with standard hyper parameters, comprehensive evaluation of models using R-squared, Mean Squared Error (MSE), Mean Absolute(MAE) and Root Mean Squared Error (RMSE) evaluation metrics. The output results revealed that both the models fairly performed on real time sensor data, they struggled with historical regional due to its vastness,complexity and lack of granularity. The study highlights the significance of granularity in enhancing predictive capabilities and suggests potential applications for real-time traffic management systems. Future research aims to enhance forecasts by integrating more external factors and creating real-time analysis frameworks that enable quicker reactions to shifting traffic situations. The output results revealed that both the models performed fairly on real time sensor data with best scores for

GRU: R-squared 0.890899, RMSE 0.229130 and MAE 0.168262 and LSTM: R-squared 0.893829, RMSE 0.226033 and MAE 0.163932
Regional data

GRU: R-squared 0.404736, RMSE 0.999625 and MAE 0.653231 and LSTM: R-squared 0.402661, RMSE 1.001367 and MAE 0.676723

Keywords: LSTM, GRU, sensor data, regional data, R-squared, RMSE, MAE

1 Introduction

1.1 Background:

With the rapid growth of urbanization, it has led to unprecedented challenges in managing commutation or transportation systems such as congestion of traffic, becoming a ubiquitous issue in metropolitan cities. This congestion not only hampers but also has significant impact on economy, environment and society. Traditional methods to predict traffic have faced a lot of challenges to cope up with the dynamic nature of urban traffic. However, with advancement in technology particularly with availability of real-time and evolution of modernized techniques like deep learning offer a promising solution to tackle this challengeTedjopurnomo et al. (2020). There has been a drastic change in the management of transportation systems with introduction of real time data. Real time data can be sourced from GPS enabled devices, traffic sensors, traffic cameras and many other real time sensing devices. The detailed description of real time offers insights into traffic patterns, anomalies and congestion dynamics, forming the base for predictive modelling and proactive traffic management strategies. Deep learning a subset of artificial intelligence is proven to be a powerful tool for analyzing complex patterns within large dataset. Latest techniques such as CNN, RNN and its variants such as LSTM or Gated Recurrent Unit have demonstrated remarkable capabilities. These techniques can efficiently handle temporal and spatial data and make accurate predictions. Traffic prediction involves several stages such as cleaning, normalization, feature extraction etc from real time data. The pre processed data are fed into deep learning architectures for training. Model training, validation and hyper parameter tuning are essential to maintain models performance. Models performance can be evaluated using evaluation metrics such as RMSE, R2 r-squared and Mean Absolute Error to assess the accuracy of these modelsChicco et al. (2021). Real world applications of deep learning models in traffic prediction showcase practical utility of these models. This case study highlights application of deep learning model GRU on real-time data and region data and optimize travel routes for the user. The accuracy of applied deep learning model on real-time constrained to junction gives better accuracy rather than predicting for the entire region. These application can have better impact in reducing traffic congestion, improving commutation time and enhancing overall mobility. Looking at the future traffic prediction using real time data holds immense potential. Advancement in technology such as incorporation or multimodal data sources such as text or images or audio, edge computing, re-inforcement learning can further improve accuracy and real time decision making. By integrating deep learning model with real time data such as data from traffic cameras, satellite and various other sources will pave way for a sustainable transportation system, ultimately enhancing quality of urban life.

1.2 Research Question and Objective:

How integration of real time data from sources enhance accuracy of prediction of model in urban regions?
Integration of real time data from various sources can significantly enhance the accuracy of prediction in urban regions. Real time data extraction from sensors, IoT devices, traffic cameras and other sources can assist the models ability to capture immediate changes in traffic patterns, environmental conditions, public sentiment and more. However in this case study we have considered sensor data from 6 different junctions based in United Kingdom region. The aim of this research study is to improve precision of urban planning, resource allocation, improve predicting capabilities for traffic management, leveraging minute data insights and facilitate sustainable urban development.

1.3 Document Structure:

The research document is broken down into seven sections, each of which offers details on a distinct aspect of the investigation. The second section, which is divided into eight subsections, summarizes what prior research has been done and highlights the unique characteristics of this project before closing; in the third segment, the technique utilized in the study is detailed; and in the fourth section, the design specifications, it provides information on the procedures and methods used and the project's key performance indicators. Section 5 will demonstrate how the technical solution was used for the research and tools used, Section 6 provides the depth of case studies and how the assessment procedure helped the research attain its goals. Finally, in section 7 the research paper will wrap up the topic with the relevant findings and discuss potential future studies.

2 Related Work

In this section we will discuss all the related works related to traffic prediction and analysis on the same.

2.1 Traffic prediction survey on smart cities

The term "smart city" in Qin et al. (2010) can be referred as use of data and latest communication technologies to analyze and integrate key information from core systems in operating cities. At the same time, these services can make intelligent responses to different types of needs arising in terms of daily livelihood, environmental protection, public safety as well as industrial and commercial activities. The 2 key aspects among the notable goals of smart cities are smart transportation systems and smart urban systems which can significantly influence lives of residents in smart cities. Advanced Traffic Management Systems (ATM's) and Intelligent Transportation Systems (ITS's) are formed on the principle of integration of information, communication and other technologies and apply them in the field of transportation to build an integrated system of people, road and vehicles. These systems constitute a large fully functional, real-time, precise and efficient transportation management framework An et al. (2011). Challenges associated due to rapid growth of urbanization are namely traffic congestion Al-Kadi et al. (2014), increase in fuel or energy consumption, enormous emission of pollutants. Intelligent traffic management systems such as ATMS and ITS can help overcome negative impacts of city-dwellers. Forecasts can also support traffic centers in managing road networks and allocating resources systematically, such as opening/closing lanes, dynamic pricing parkingQian and Rajagopal (2014), adaptive traffic lights De Gier et al. (2011), high level of automation Zhang et al. (2011).

2.2 Data from sensors for traffic prediction

Sensors are based on type of sensors which are deployed at a fixed position. Due to fixed position these sensor always measure at a specific point. They might measure single or multiple lanes based on the capacity of sensors. The biggest advantage of fixed position sensors is that it has the ability to capture data of all the vehicles passing by. A moving sensor can only capture data of one vehicle it is travelling in for example GPS sensors. Aggregate statistics such as number of vehicles or density flow can be captured using fixed position sensors. Automated fare collection (AFC) contain fixed position sensors and data can be harvested from them. AFC's are used for toll collection. The smart ticketing data can be important input for the study of urban mobility patternsLi et al. (2018). There have been more than 10 papers Mohamed et al. (2016); Zhong et al. (2015, 2016) which have to deal with smart ticketing but none of them had publicly available data sets.

2.3 Traffic flow prediction using parametric models

Traffic flow prediction is distinguishes between two broad classification namely parametric and non parametric models. Parametric models are referred as models which have a structured expression with predetermined assumptions about variable Mao et al. (2017). ARIMA, Kalman filtering, maximum likelihood estimation, Linear Regression Bao et al. (2018) are examples of parametric models. Although the characteristics of parametric models were inflexible, they were applied in the past for traffic characteristics description. Traffic was prediction ineffectively due to non-linearity and variations in traffic data Tian and Pan (2015).

2.4 Introduction of deep learning to traffic prediction

Recently deep learning was introduced to traffic prediction and widely accepted approach for describing traffic systems precisely. Description of distributed and hierarchical features for complex traffic flow data can be done by using multi layer non-linear structures of deep learning models. There have been researches done in the area of clustering approaches for traffic flow prediction Huang et al. (2014) for example deep belief networks. Li et al. (2019) proposed an intelligent swarm based model to optimize parameters of DBN and enhance its multiple steps ahead prediction capability. Lv et al. (2014) proposed attention based model like Stacked Auto-Encoder(SAE) which outperformed Random Walk approach, Feed-Forward Neural Networks, and Radial Basis Function(RBF). Each neuron of previously hidden layers are connected to each neuron of next layer hence adding to density of the model. All the features and characteristics are extracted from the dataset automatically which results into no assumptions made by Fully connected network models Wu et al. (2018).

2.5 GRU and LSTM for traffic prediction

GRU architecture was proposed as a RNN variant by Cho et al. (2014). This was proposed in the year 2014 to solve vanishing gradient problem. Application of GRU to traffic prediction problem successfully, especially traffic flow estimation in few steps. Results calculated are superior to SAE, FFNN and SVM. GRU architecture is similar to LSTM and can produce equally excellent results. However internal structure is simpler and more rapid than LSTM. The 2 control gates namely reset and update gates are responsible to overcome the vanishing gradient problem in RNN. The update gate helps to determine the past information from the time series that need to pass to the future. Keeping such information helps eliminate the risk of vanishing explosion. Reset gate however determines to decide how much information to forget. External factors such as weather conditions, changing populations and social economic events play a important role in traffic prediction as proposed by many researchers.Determination of optimal time-lags and hyper parameters value for traffic flow prediction has a stronger applicability like LSTM with a simpler configuration network Jia et al. (2017). The study Hussain et al. (2021) proposed the fundamentals of GRU network with hyperparameter optimization analysis combined with window tuning steps for time series prediction and obtained RMSE, MAE as 7.13 and 0.3 respectively. The study Tian et al. (2018) demonstrated LSTM on PeMS dataset. The study compared traffic flow prediction at two intervals 15 minutes and 60 minutes. For 15 minutes prediction interval, the LSTM model achieved a MAE and RMSE of 0.385 and RMSE of 7.6 respectively. All of the tests mentioned above are displayed in tabular format in Table 1.

2.6 Restrictions and Learning from Literature

Section 1 is limited to traffic prediction survey on smart cities and researchers fail to cover all points on advantages and disadvantages of intelligent transportation systems. Section 2 describes the study of integration of real time data from sensors for traffic prediction. Sensors installed on AFC's can be used for traffic prediction on the particular junction is discussed in second section.While the study focuses on integration of real time data, it may overlook the challenges and limitations associated with sensor data such as data quality, sensor placement, maintenance issues. The third section discussed the prediction of traffic using parametric models namely ARIMA, Kalman filtering, Linear Regression and many other techniques. However the fourth section discusses the introduction of traffic prediction using deep learning models. RNN variants namely GRU and LSTM share a similar architecture and their implementation for traffic prediction is discussed in section 5. The most important limitation is the deployment and scalability of proposed traffic prediction models.

References	**Technique**	**Data**	**Score**
Hussain et al. (2021)	GRU	PeMS	7.3 for RMSE & 0.3 for MAE
Tian et al. (2018)	LSTM	PeMS	7.6 for RMSE & 0.385 for MAE

Table 1: Comprehensive Summary of Techniques and Scores for Traffic Prediction

3 Methodology

There are 2 types of methodologies available for analysis are KDD(Knowledge discovery in databases) and Crisp-DM. In this research study we have taken KDD methodology as this is the most appropriate method for research. The methodology implementation is shown in the figure 1.

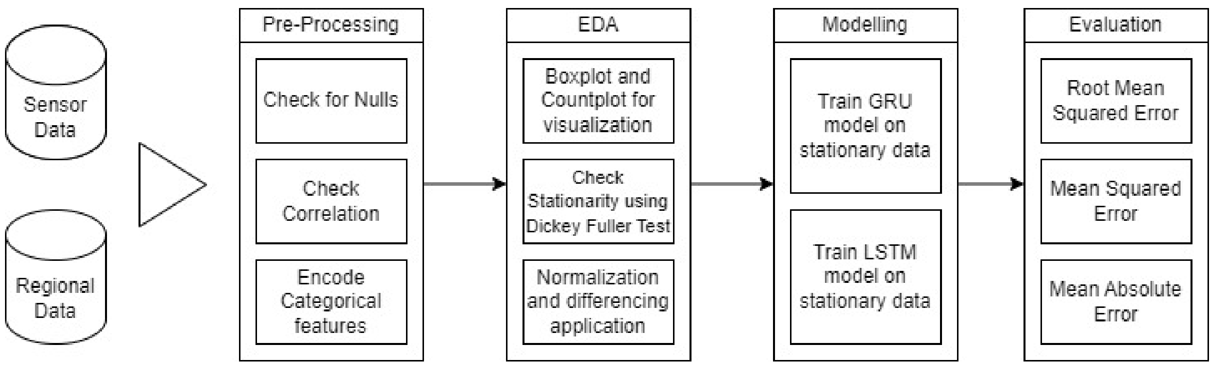

Figure 1: KDD Methodology

3.1 Data Selection and Understanding:

We have taken two open source datasets namely Traffic Sensor Data, Traffic Regional Data of United Kingdom region for this research in order to compare traffic sensor data with traffic data of entire region and perform traffic prediction on the same [1] [2]. Traffic sensor data contains data of different traffic sensors installed at different junctions whereas traffic region data contains traffic data of different regions namely London, Scotland and Wales. The traffic sensor data contains information ranging from year 2019 - 2020. The traffic data contains information ranging from 2000 - 2022. The common features among both the datasets are namely Data and Time, Region/Junction, Motor Vehicle Count etc. Traffic sensor dataset has 69985 observations and 14 features whereas traffic regional data has 267636 observations and 35 features.

3.2 Exploratory Data Analysis:

In order to have a better understanding of data and get familiar with most important characteristics of data, exploratory data analysis is performed where summarization and visualization of features are taken place Páez and Boisjoly (2023).

3.2.1 Data Cleaning and Preprocessing:

Noisy data can impair the models performance. Preprocessing and cleaning of data are essential at this point. Therefore following cleaning and preprocessing procedures are part of research. As a part of cleaning, nulls have been checked for both the dataset. No null or NA data was present in traffic sensor data although traffic data had many null values and hence as a part of treatment null values were dropped. In order to remove irrelevant features, heatmap was plotted to check collinearity between feature and drop strongly correlated variable as a part of dimensionality reduction. After implementation it was found that traffic data had highly correlated variables and were dropped as apart of treatment. Heatmap for regional data and sensor data can be seen in 3 and 2 respectively.

3.2.2 Plot Vehicles Per Region/Junction:

In order to get a proper understanding of data, we have visualized total count vehicles per region or junction with the help of Boxplot from Seaborn library. Figure 4 contains data of count of all the vehicles per junction whereas figure 5 contains count of all motor vehicles per region.

The below figure 6 shows plotting of all the vehicles count in regional data for each year ranging from 2000 - 2022

3.2.3 Check Stationarity For Traffic Forecasting:

Stationarity refers to statistical property that remains constant over a period of time van Greunen and Heymans (2023). Checking for stationarity involves assessment of time series data satisfying certain conditions namely constant mean, variance and covariance. Stationarity check plays a crucial role for several reasons. Many forecasting models are

[1] `https://www.data.gov.uk/dataset/d3a76dbd-9936-4375-9ba6-e2974fafc943/mill-road-project-traffic-sensor-data/datafile/98358736-af12-46d2-932e-ea6028d82040/preview`

[2] `https://roadtraffic.dft.gov.uk/downloads`

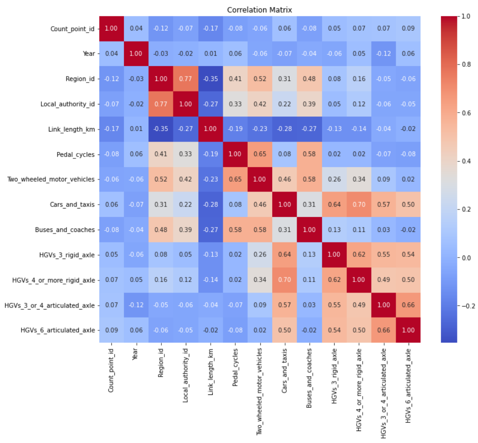

Figure 2: HeatMap for Traffic Regional Data

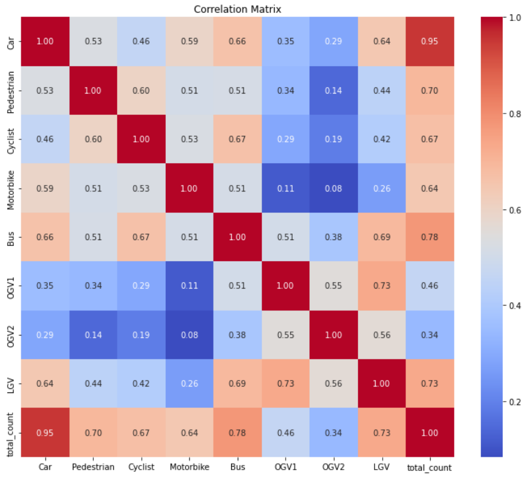

Figure 3: HeatMap for Traffic Sensor Data

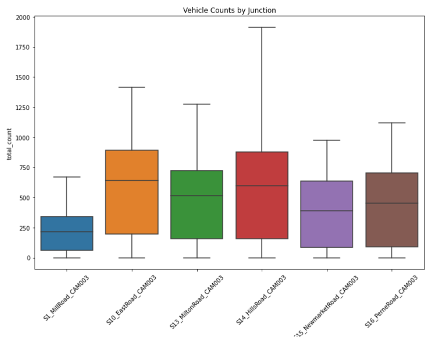

Figure 4: BoxPlot for Traffic Sensor Data

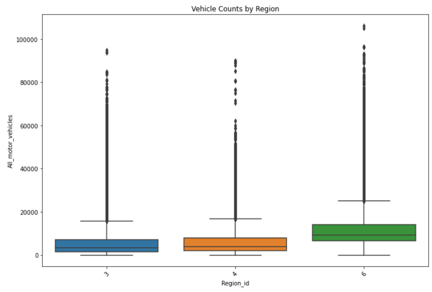

Figure 5: Boxplot for Traffic Regional Data

likely to assume the underlying data as stationary. Hence, if the data is non stationary, the assumptions may be violated, leading to inaccurate predictions. Traffic data usually exhibits underlying trends and seasonality. Understanding stationarity helps identify these components. Traffic sensor data and traffic regional data when checked was found that the data was not stationary. Methods like normalizing and differencing were applied to non stationary data to make it stationary. In order to evaluate the results post normalizing and differencing, Dickey Fuller test was used to check the stationarity. Figure 7 and figure 8 contains graph of non stationary data of sensor data and regional data respectively.

The below figure 9 and 10 shows output results after applying techniques like differencing and normalizing in the form of a lineplot from searborn library.

Table 2 and Table 3 are output results of Dickey fuller test. This test plays significant role in time series modelling, which in our case is traffic forecasting. Both the tables shows output results after application of techniques like Normalizing and Differencing to make data stationary. In all the cases P-Value is less than 0.05 which means it rejects null hypothesis and concludes time series data is stationary.

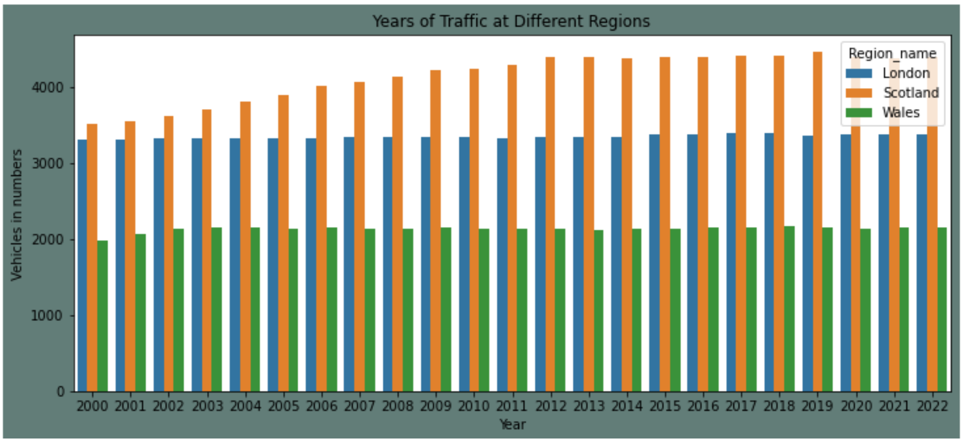

Figure 6: Countplot for Traffic Sensor Data

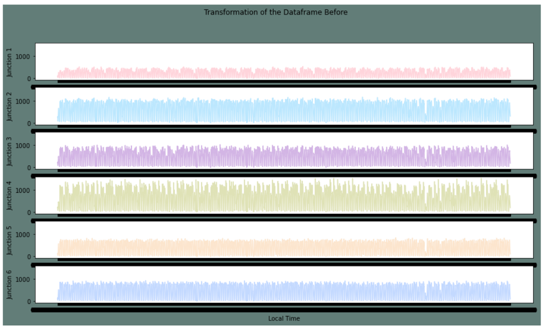

Figure 7: Stationary Plot for Traffic Sensor Data

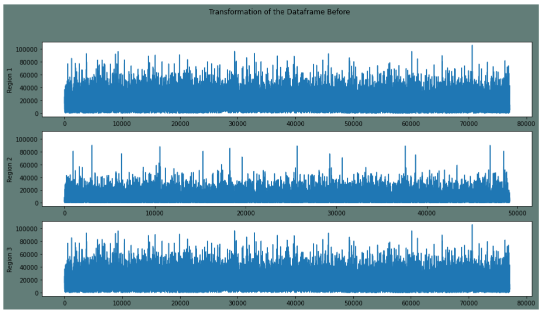

Figure 8: Stationary Plot for Traffic Regional Data

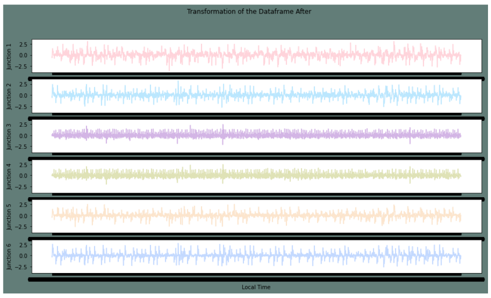

Figure 9: Stationary Plot for Traffic Sensor Data

Figure 10: Stationary Plot for Traffic Regional Data

3.3 Modelling and Evaluation:

The Later stage of implementation is selection of model, perform modelling on the selected model and evaluating the model. Assessment of the model can be done by calculating accuracy of the data Amini et al. (2023). We can attain this by splitting dataset into sets i.e. Training Set and Testing Set. The data from training set can be utilized into training model whereas data from testing set can be treated as un-labeled data for the model, used for prediction. The sample data is then compared to real samples to determine the average model accuracy. The results obtained for all the approaches includes metrics such as MSE, RMSE, MAE and R-Squared.

Table 2: **ADF test results for Sensor Data**

Junctions	p-value	ADF Statistic	Stationarity
Junction 1	3.604×10^{-23}	-12.45	Yes
Junction 2	3.540×10^{-24}	-12.94	Yes
Junction 3	9.458×10^{-25}	-13.43	Yes
Junction 4	2.193×10^{-25}	-13.57	Yes
Junction 5	4.413×10^{-24}	-12.89	Yes
Junction 6	2.759×10^{-24}	-12.99	Yes

Table 3: **ADF test results for Regional Data**

Regions	p-value	ADF Statistic	Stationarity
Region 1	0.0	-58.07	Yes
Region 2	0.0	-49.69	Yes
Region 3	0.0	-58.07	Yes

4 Design Specification

The design specification module comprises detailed requirements, limitations and objectives of deep learning system,considered to be one of the initial stages of product management. The design specification outlines the foundational elements for implementing deep learning model, architecture, framework, and associated requirements. This module also includes details on methods and algorithms which will be applied including systems expected performance. Procedures execution phase is explained in great detail in this stage. The optimal model is chosen and applied to practice data in two stages of modelling study. Evaluation matrix are selected based on responses to inquiry for research.

4.1 Modelling Techniques:

4.1.1 Long Short Term Memory RNN:

Artificial Neural Networks are influenced from biological learning systems and comprised of loosely modelled basic functions. Biological learning systems are complex webs of interconnected neurons. The most standard type of neural networks are feed forward neural networks. The architecture of these networks are organised in the form of layers which are input layer, output layer and one intermediate hidden layer. Feed forward neural networks are limited to static classification task. An extension towards dynamic classification can be done by feeding signals from previous timestamps back into the network. These networks are known are Recurrent Neural Network (RNN). The only constraint of RNN networks are vanishing gradient which limits them to look back into the time approximately ten time stamps. This issue was resolved by LSTM RNN neural networks and have a capability to learn more than 1000 timestamps Staudemeyer and Morris (2019).The LSTM are represented by the following formula given below.

4.1.2 Gated Recurrent Unit RNN:

The GRU's reduces the gating signals from LSTM RNN. The 2 gates are called update gate and reset gate. Both GRU and Simple RNN share similar parameterization. Increase in weights are also updated using BTT (backpropogation through time) stochastic gradient descent. Hence all hidden states are reflected in latest state variables. Moreover, the adaptive parameter updates all involved components of the internal state of the system.

4.2 Model Evaluation:

Both the models have been evaluated using these three evaluation metrics.

4.2.1 R-Squared:

R-squared is a statistical measure that represents proportion of variance of dependent variables are explained by independent variable or variables. Values of R2 usually range from 0 to 1 where higher values indicate better fit of model.

4.2.2 Mean squared Error:

MSE is also a statistical measure of average squared difference between actual and predicted values. The equation can be explained as average of squared differences between predicted and actual values. MSE are not negative and values equal or closer to zero indicate perfect predictions.

4.2.3 Mean Absolute Error:

MAE is a statistical measure where the average magnitude of error in set of prediction is calculated, irrespective of direction. All the aspects of MAE are similar to MSE except they are less sensitive to outliers because unlike MSE it does square the errors in the calculation.

5 Implementation

This section consists of all the implementation part which includes transformation of data, apply modelling, output results calculated etc.

Note: This research project consists of two datasets and has separate implementations of both the datasets. Two datasets are Traffic Sensor Data and Traffic Regional Data namely. Particular implementation name will be mentioned in the section name.

5.1 Tools Used:

This section contains a list of tools used for this research study.

5.1.1 Hardware Specifications:

Specifications of the system used for this study are Intel Core i5 processor, 8GB RAM and 512GB ROM.

5.1.2 Software Tools:

In this research case study we leveraged comprehensive suite of software tools for data analysis and predictive modelling. The primary IDE used is Jupyter Notebooks, widely used platform for data science tasks. Core Programming language is Python, known for its extensibilty and support for data manipulation and analysis. For data visualization, we used Python's inbuilt packages namely Matplotlib and Seaborn. These packages enable us to effectively give insights on data through visual representation.

5.2 Data Transformation 1 - Traffic Sensor Data:

This section consists all the implementation techniques related to only Traffic Sensor Data.

5.2.1 Label-Encoding Categorical Data:

Categorical features namely CountlineName (name of the junction) and direction (direction of vehicles) are relevant features and must be label encoded. We have imported Label encoder pre defined function from Sciket-Learn package for label encoding categorical data.

5.2.2 Junction-Wise Data Aggregation, Normalization and Differencing:

The dataset has 6 junctions and separate dataframe was prepared for each junction to predict traffic separately. Irrelevant features namely Car, Pedestrian, Cyclist, Motorbike etc. were added into a single feature named as total count and dropped. Descriptive statistics were calculated for each dataframe junction to get an overview of the data.Separate junction dataframes were subjected to techniques such as Normalization and Differencing to make the data stationary and ready for prediction. Parameterized normalization function was created with Dataframe (Junction Dataframe) and Column(total count column) as input arguments. A parameterized differencing function was also defined with interval as an additional argument. We have considered a week's difference as an interval.

5.3 Data Transformation 2 - Traffic Regional Data:

This section consists all the implementation techniques related to only Traffic Sensor Data.

5.3.1 Region-Wise Data Aggregation, Normalization and Differencing:

The dataset has 3 regions and separate dataframe was prepared for each region to predict traffic separately. Irrelevant features were added into a single feature named as all motor vehicles and dropped. Descriptive statistics were calculated for each dataframe region to get an overview of the data.Separate region dataframes were subjected to techniques such as Normalization and Differencing to make the data stationary and ready for prediction.

Parameterized normalization function was created with Dataframe (Region Dataframe) and Column(all motor vehicles column) as input arguments. A parameterized differencing function was also defined with interval as an additional argument.

5.4 Training Model and Hyperparameters:

To determine the difference, prediction for sensor and regional data was performed using both GRU and LSTM models with identical settings.Transformed data is then splitted and fed into the respective models for training. The model is trained for 50 epochs with a batch size of 32. Training and validation loss are plotted to visualize the epoch pattern. Model makes prediction X-test and returns predicted values. We have used RMSE, MAE and R-squared to evaluate models performance. PredictionsPlot provides a visualization of model's performance by plotting actual vs predicted values.

5.4.1 Hyperparameters for GRU model

GRU model function constructs a GRU based neural network model for traffic prediction.

- The model contains 3 layers of GRU with tanh activation. The first layer has 100 units and return sequences. The second layer has 50 units and also return sequences whereas the third GRU layer has 50 units but does not return sequences.
- To prevent overfitting, dropout layers have been added after each GRU layer.
- RMSprop optimizer with a learning rate of 0.001 is used. RMSprop have the ability to handle vanishing/exploding gradients, adapt to learning rates, prevent oscillation etc and hence are considered effective for training RNN.
- An early stopping callback is implemented with learning rate of 0.001 and patience of 10 epochs. This setting helps in stopping training when validation loss has reached a threshold value and hence prevent overfitting.

5.4.2 Hyperparameters for LSTM model

LSTM model function constructs a LSTM based neural network model for traffic prediction.

- The model contains 5 layers of GRU with tanh activation. The first layer has 128 units and return sequences. The second and third layers has 64 units,fourth layer has 32 units and also return sequences whereas the fifth GRU layer has 32 units but does not return sequences.
- To prevent overfitting, dropout layers have been added after each GRU layer.The value is set to 0.2 which means 20 percent are randomly ignored which in turn provides a form of regularization.
- Adam optimizer is used with LSTM with a learning rate of 0.001. Adam due to its efficiency and adaptive learning rate properties is a best fit for training neural networks.
- An early stopping callback is implemented with learning rate of 0.001 and patience of 10 epochs. This setting helps in stopping training when validation loss has reached a threshold value and hence prevent overfitting.

6 Evaluation

Accoring to project pipeline, evaluation plays a critical role that helps gauge models performance and confirm the model is operating as intended. In the discussion section we have considered and compared output with best evaluation metrics score from all the cases. Here are the list of case studies listed and discussed below.

- GRU model's performance on Traffic Sensor Data
- LSTM model's performance on Traffic Sensor Data
- GRU model's performance on Traffic Regional Data
- LSTM model's performance on Traffic Regional Data

6.1 Case 1: GRU model's performance on Traffic Sensor Data

Overall results of this case study with evaluation metrics is shown in a tabular format in figure 11.

	Junction	RMSE	R-Squared	MAE
0	Junction1	0.306659	0.739601	0.225379
1	Junction2	0.267081	0.705212	0.184387
2	Junction3	0.261878	0.794659	0.191852
3	Junction4	0.229130	0.890899	0.168262
4	Junction5	0.274593	0.674920	0.187274
5	Junction6	0.271298	0.717538	0.173206

Figure 11: GRU Sensor Data Output

After the transformation of data, the GRU model is then executed on the test data of Sensor dataset with the hyper parameters mentioned above. All the 6 junctions transformed dataframe is subjected to the GRU model. The dataset was executed for 50 epochs with a batch size of 32. In order to avoid overfitting an early stopping was added with a min delta value of 0.001 and patience 10. The evaluation metrics results across 6 different junctions indicate varied performance of the prediction model. However junction 4 outperforms all the other junctions and lowest errors, signifying excellent predictive accuracy. Followed by junction 3 with strong results. R-squared for junction 4 accounts for approximately 89 percent of variance in data whereas 79 percent for junction 3. Junction 2,5 and 6 display moderate performance with Junction 2 have a slight edge over others with balance of predictive power and lower errors. Overall GRU performs fairly on the sensor data with some junctions showing notably higher precision in prediction. As junction 4 has best model performance among all the junctions, figures which include evaluation plot13 and training and validation loss graph 12 have been displayed below for junction 4. According to the evaluation plot, we can see that predicted and original value follow a similar trend.

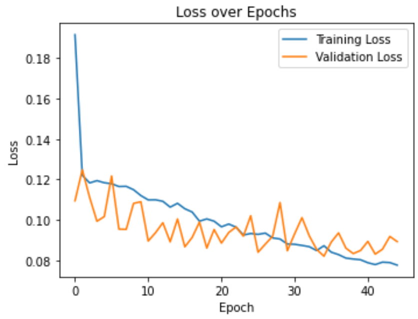

Figure 12: TV Loss Plot for GRU Sensor Data

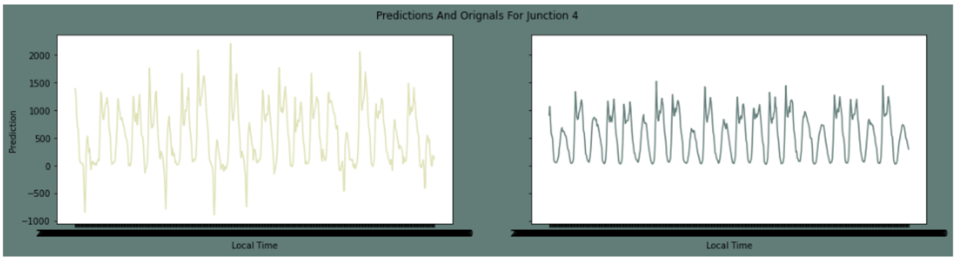

Figure 13: Evaluation Plot for GRU Sensor Data

6.2 Case 2: LSTM model's performance on Traffic Sensor Data

Overall results of this case study with evaluation metrics is shown in a tabular format in figure 20.

	Junction	RMSE	R-Squared	MAE
0	Junction1	0.285303	0.774607	0.210253
1	Junction2	0.218244	0.803162	0.158405
2	Junction3	0.221888	0.852584	0.164549
3	Junction4	0.226033	0.893829	0.163932
4	Junction5	0.252805	0.724461	0.172743
5	Junction6	0.190241	0.861108	0.126074

Figure 14: LSTM Sensor Data Output

LSTM model is applied to test data of Sensor Data with hyperparameters mentioned. All the 6 junctions transformed dataframe is subjected to the LSTM model. As LSTM and GRU both are variant of Recurrent Neural Networks, they follow a similar trend and give similar output results. As per the above output results, junction 4 data outperforms and gives a similar 89 percent high accuracy. As junction 4 is the best performing model, training and validation loss plot and evaluation plot for junction 4 is displayed in figure 15 and 16 respectively. However as per tabular output results of LSTM data, LSTM performed better than GRU as the minimum accuracy achieved was 72 percent in comparison of GRU model which was 67 percent. Hence we can conclude that LSTM can identify underlying patterns better and give accurate predictions

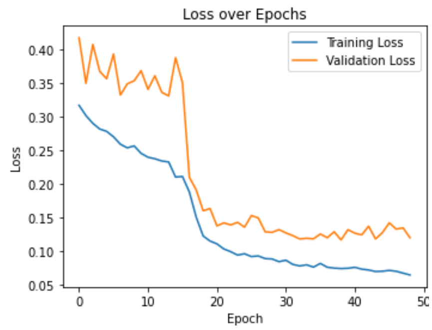

Figure 15: TV Loss Plot for LSTM Sensor Data

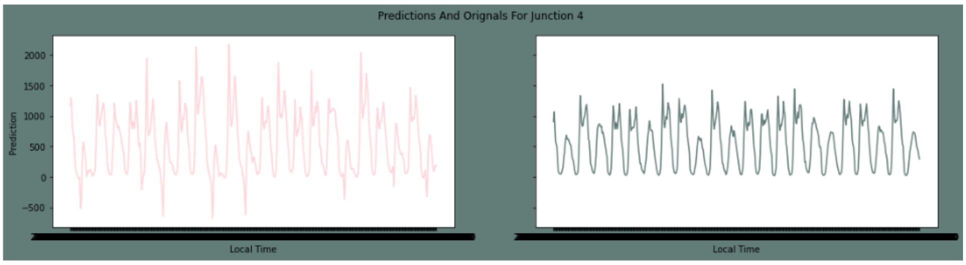

Figure 16: Evaluation Plot for LSTM Sensor Data

6.3 Case 3: GRU model's performance on Traffic Regional Data

Overall results of this case study with evaluation metrics is shown in a tabular format in figure 17

	Region	RMSE	R-Squared	MAE
0	Region1	1.002521	0.401282	0.682273
1	Region2	0.981422	0.363964	0.629490
2	Region3	0.999625	0.404736	0.653231

Figure 17: GRU Regional Data Output

After the transformation of Regional Data, the GRU model is then subjected to Regional dataset on test data. As the dataset is very vast, we have divided the dataset into 3 regions. Similar to sensor data, dataset was executed for 50 epochs with a batch size of 32. Hyperparameters of Sensor Data and Regional Data are same in order to compare data and output. As there is no major difference in the output of all the 3 regions, we can say that there is no varied performance of prediction model. The output of Region 3 is marginally better than Region 1 and Region 2. Region 3 is 0.3 percent better than Region 1 and 3 percent better than Region 2. Similarly other evaluation metrics parameters namely RMSE and MAE also do not have a major difference in the output as seen in the tabular output result format. Region 3 Traning and validation loss plot and Evaluation plot of actual vs predicted values are displayed in figure 18 and 19 respectively. As the dataset of region is vast, there may be probable noise or underlying patterns in the data which are not correctly identified by the model.

Figure 18: TV Loss Plot for GRU Sensor Data

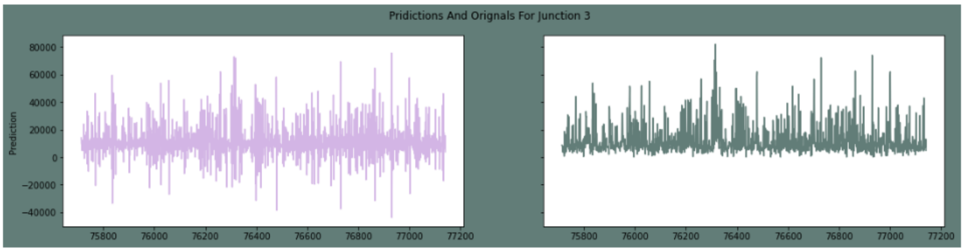

Figure 19: Evaluation Plot for GRU Sensor Data

6.4 Case 4: LSTM model's performance on Traffic Regional Data

Overall results of this case study with evaluation metrics is shown in a tabular format in figure 20

	Junction	RMSE	R-Squared	MAE
0	Region1	1.001367	0.402660	0.676723
1	Region2	0.974499	0.372905	0.634605
2	Region3	1.006856	0.396093	0.648143

Figure 20: LSTM Regional Data Output

LSTM model was subjected to all the 3 regions. A major difference was not observed in the outputs of both the models. Unlike the case of sensor data where junction 4 outperformed for both the models, Region 1 in this case shows marginally better results than other Regions. However, Region 2 has under-performed in both the cases of Regional

Data. A downward trend of epoch loss has been observed throughout the research study. Evaluation Plot for both the models has also found out to be similar for both the models. RMSE and MAE also do not have huge difference in the output similar to GRU model. Training and Validation loss plot and Evaluation plot for Region 1 is plotted in figure 21 and 22 respectively. As the dataset is vast for LSTM as well, LSTM failed to identify underlying patterns in the data, hence resulting in poor prediction.

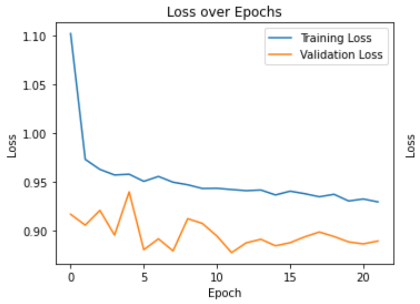

Figure 21: TV Loss Plot for LSTM Regional Data

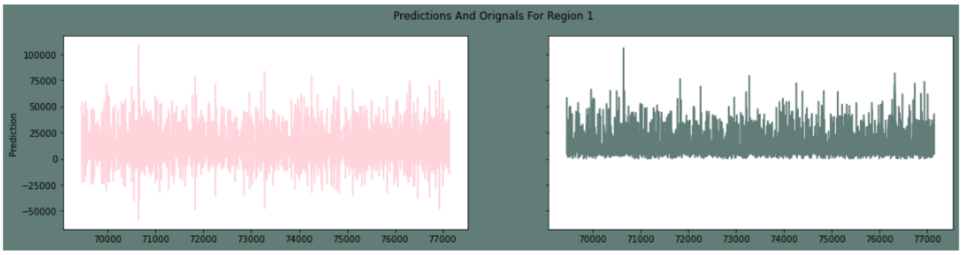

Figure 22: Evaluation Plot for LSTM Regional Data

6.5 Discussion

As per the output results, a major difference is found in the output of both the dataset i.e. Models have been correctly predicting real time data approximately 50 percent more better than regional data (historical data). The probable reason as to why there is major difference in the output result could be with nature of datasets. Sensor data is found to be more granular than regional data, hence offering high resolution of information. The granularity in the data can help the models to capture temporal patterns which seem to be more diluted in the case of aggregated regional data. Although with granularity comes with more noise in the data, if LSTM and GRU are able to filter or interpret noise effectively, they may provide better predictions at sensor level rather than regional level. A regional data may contain multiple junctions/traffic sensor data, often encompassing larger area. Due to this it may introduce more variability in data across different locations. Complex patterns of sensor data are recognized well by LSTM and GRU models due to their complex architecture. One of the reason for better performance models on sensor could be the feature exhibited by the sensor data are more relevant for prediction compared to those used for regional data. Overall the project is addressing the research question and prediction is giving better results for real time data rather than regional or historical data. In this project, it was assumed that there could have been incorrect entries in the regional data for the extremely low results. The results show the traffic prediction at multiple junctions or regions, this information could be use by local bodies to manage the congestion and ensure a smooth flow of vehicles.

7 Conclusion and Future Work

7.1 Conclusion:

The comparative analysis of LSTM and GRU models on traffic sensor data and regional data indicate choice of model is directly dependent on the granularity of data. Also the granularity of data can provide a conducive environment for the models to leverage their strengths in modeling temporal sequences, leading to better predictive results. In contrast, huge data from regional datasets introduced more complexities that were not correctly identified by the model, hence resulting less accurate predictions and low evaluation metrics result.

7.2 Future Works:

- **Incorporation of Additional Data:** External factors such as bad weather, special events and traffic regulations might improve predictive capabilities of model.
- **Real Time Analysis:** Developing a real time analysis framework, enabling traffic management and more immediate responses to changing conditions.
- **Adaptive Traffic Signal Management:** Future efforts could focus on integrating predictive model with traffic signal systems. By understanding the vehicle flow patterns, we can introduce dynamic traffic signal timing, hence optimizing traffic flow and reducing congestion.

References

Al-Kadi, O., Al-Kadi, O., Al-Sayyed, R. and Alqatawna, J. (2014). Road scene analysis for determination of road traffic density, *Frontiers of Computer Science* **8**: 619–628.

Amini, M., Sharifani, K. and Rahmani, A. (2023). Machine learning model towards evaluating data gathering methods in manufacturing and mechanical engineering, *International Journal of Applied Science and Engineering Research* **15**(2023): 349–362.

An, S.-h., Lee, B.-H. and Shin, D.-R. (2011). A survey of intelligent transportation systems, *2011 third international conference on computational intelligence, communication systems and networks*, IEEE, pp. 332–337.

Bao, G., Zeng, Z. and Shen, Y. (2018). Region stability analysis and tracking control of memristive recurrent neural network, *Neural Networks* **98**: 51–58.

Chicco, D., Warrens, M. J. and Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation, *PeerJ Computer Science* **7**: e623.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation, *arXiv preprint arXiv:1406.1078* .

De Gier, J., Garoni, T. M. and Rojas, O. (2011). Traffic flow on realistic road networks with adaptive traffic lights, *Journal of Statistical Mechanics: Theory and Experiment* **2011**(04): P04008.

Huang, W., Song, G., Hong, H. and Xie, K. (2014). Deep architecture for traffic flow prediction: Deep belief networks with multitask learning, *IEEE Transactions on Intelligent Transportation Systems* **15**(5): 2191–2201.

Hussain, B., Afzal, M. K., Ahmad, S. and Mostafa, A. M. (2021). Intelligent traffic flow prediction using optimized gru model, *IEEE Access* **9**: 100736–100746.

Jia, Y., Wu, J., Ben-Akiva, M., Seshadri, R. and Du, Y. (2017). Rainfall-integrated traffic speed prediction using deep learning method, *IET Intelligent Transport Systems* **11**(9): 531–536.

Li, L., Qin, L., Qu, X., Zhang, J., Wang, Y. and Ran, B. (2019). Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm, *Knowledge-Based Systems* **172**: 1–14.

Li, T., Sun, D., Jing, P. and Yang, K. (2018). Smart card data mining of public transport destination: A literature review, *Information* **9**(1): 18.

Lv, Y., Duan, Y., Kang, W., Li, Z. and Wang, F.-Y. (2014). Traffic flow prediction with big data: A deep learning approach, *IEEE Transactions on Intelligent Transportation Systems* **16**(2): 865–873.

Mao, B., Fadlullah, Z. M., Tang, F., Kato, N., Akashi, O., Inoue, T. and Mizutani, K. (2017). Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning, *IEEE Transactions on Computers* **66**(11): 1946–1960.

Mohamed, K., Côme, E., Oukhellou, L. and Verleysen, M. (2016). Clustering smart card data for urban mobility analysis, *IEEE Transactions on intelligent transportation systems* **18**(3): 712–728.

Páez, A. and Boisjoly, G. (2023). Exploratory data analysis, *Discrete Choice Analysis with R*, Springer, pp. 25–64.

Qian, Z. S. and Rajagopal, R. (2014). Optimal dynamic parking pricing for morning commute considering expected cruising time, *Transportation Research Part C: Emerging Technologies* **48**: 468–490.

Qin, H., Li, H. and Zhao, X. (2010). Development status of domestic and foreign smart city, *Global Presence* **9**: 50–52.

Staudemeyer, R. C. and Morris, E. R. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks, *arXiv preprint arXiv:1909.09586* .

Tedjopurnomo, D. A., Bao, Z., Zheng, B., Choudhury, F. M. and Qin, A. K. (2020). A survey on modern deep neural network for traffic prediction: Trends, methods and challenges, *IEEE Transactions on Knowledge and Data Engineering* **34**(4): 1544–1561.

Tian, Y. and Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network, *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, IEEE, pp. 153–158.

Tian, Y., Zhang, K., Li, J., Lin, X. and Yang, B. (2018). Lstm-based traffic flow prediction with missing data, *Neurocomputing* **318**: 297–305.

van Greunen, J. and Heymans, A. (2023). Determining the impact of different forms of stationarity on financial time series analysis, *Business Research: An Illustrative Guide to Practical Methodological Applications in Selected Case Studies*, Springer, pp. 61–76.

Wu, Y., Tan, H., Qin, L., Ran, B. and Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding, *Transportation Research Part C: Emerging Technologies* **90**: 166–180.

Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X. and Chen, C. (2011). Data-driven intelligent transportation systems: A survey, *IEEE Transactions on Intelligent Transportation Systems* **12**(4): 1624–1639.

Zhong, C., Batty, M., Manley, E., Wang, J., Wang, Z., Chen, F. and Schmitt, G. (2016). Variability in regularity: Mining temporal mobility patterns in london, singapore and beijing using smart-card data, *PloS one* **11**(2): e0149222.

Zhong, C., Manley, E., Arisona, S. M., Batty, M. and Schmitt, G. (2015). Measuring variability of mobility patterns from multiday smart-card data, *Journal of Computational Science* **9**: 125–130.