Enhancing Urban Traffic Flow Management and Analysis through Deep learning Techniques

MSc Research Project

Data Analytics

Caroline Vincent

Student ID: x22153926

School of Computing
National College of Ireland

Supervisor: Mr. Taimur Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Caroline Vincent
Student ID:	22153926
Programme:	MSc. Data Analytics **Year:** 2023
Module:	Research in Computing
Supervisor:	Mr. Taimur Hafeez
Submission Due Date:	14/12/2023
Project Title:	Enhancing Urban Traffic Flow Management and Analysis through Deep learning Techniques
Word Count:	7771 **Page Count** 22

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 author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Caroline Vincent
Date:	14/12/2023

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 Urban Traffic Flow Management and Analysis through Deep learning Techniques

Caroline Vincent
x22153926

Abstract

As urban areas grow and vehicle count surge, conventional traffic control systems struggle to keep up with the growing complexity of traffic flow dynamics. The use of deep learning techniques to enhance the analysis and prediction of urban traffic patterns is explored in this work. This research conducts a series of experiments using Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN) to forecast traffic patterns and analyse spatio-temporal data. The experiment makes use of a comprehensive dataset of UK road traffic counts from 2017 to 2021. The aim of the research to predict traffic dynamics based on vehicle influence, traffic fluctuations during peak and non-peak hours, and geospatial traffic flow analysis around Trafalgar Square. These studies have already been used in number of studies with smaller or aggregated data sets. These factors have already been used in a number of studies with smaller or aggregated data sets. The findings reveal that deep learning models perform much better in accuracy and adaptability to temporal and spatial traffic flow variations than traditional methods. The LSTM and GRU models surpass in capturing the temporal dependencies of traffic flow than the CNN model. This research not only contribute to the field of intelligent transportation systems by offering a deep learning framework for predicting traffic but also explains the potential for these models to be combined into real-world traffic management solutions, helping in the end to create smarter city designs.

1. Introduction

In the realm of Intelligent Transportation Systems (ITS), the precise prediction and traffic flow management are paramount. With enormous rise in the number of automobiles in urban areas, the task of overseeing and managing traffic becomes very complex. In order to maintain safe and smooth transit operations, innovative solutions are much needed. The vast amount of data on traffic parameters collected by the modern traffic monitoring system from sensors, video surveillance, and inductive loops present special challenges to conventional analytical techniques due to their non-linearity and spatiotemporal features. As cities grapple with increasing volume of vehicles and traffic complexities, the need for advanced predictive analytics has never been more important. This study delves into the potential of deep learning techniques to reform our understanding and prediction of traffic flow dynamics.

1.1 Background and Importance

In the increasing landscape of urban development, efficient traffic management system is essential to the smooth functioning of city infrastructures. The most difficult objective of this research is to forecast the traffic flow that has stochastic, irregular, and non-linear nature. The dynamic nature of urban traffic and increasing number of vehicles constitute a complex problem requiring advanced analytical techniques. Traditional traffic analysis techniques, while useful, are often limited in their capacity to process the huge and complicated datasets created by modern traffic monitoring technologies. These traditional methodologies tend to overlook the complex interactions between traffic flow and several non-traffic factors, like weather conditions, time of day fluctuations, and urban development patterns. Consequently, there is a big gap in the ability to precisely estimate traffic dynamics, which is critical for planning, congestion management, and improving road safety.

The advancement of deep learning offers more opportunities in the field of traffic data management. Deep learning models like Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN) are well-suited to untangle the complexities of traffic flow because of their capacity to extract high-level features and pattern recognition from massive amounts of data. These methods hold the promise of not only increasing the precision of traffic predictions based on vehicle counts but also offers a detailed comprehension of traffic variations during peak and non-peak hours. Moreover, these models can provide localized traffic insights that are crucial for targeted urban planning, particularly in famous and crowded locations like Trafalgar Square. This research aims to utilize the advanced deep learning techniques to offer a more reliable and comprehensive tool for urban traffic analysis and management. Predicting the stochastic, non-linear, and irregular traffic flow is the difficult goal of this research. Deep learning and data analytics enable the analysis to be completed quickly and with minimal computation time.

1.2 Research Question

How can deep learning techniques enhance the prediction of traffic flow dynamics from vehicle counts, determine traffic patterns during peak and non-peak hours, and incorporate geospatial metrics for targeted analysis in areas like Trafalgar Square?

The objective of this study is to leverage deep learning techniques to improve traffic flow predictions from vehicle counts, analysing fluctuations between peak and non-peak time, and integrating geospatial data for detailed traffic analysis in specific urban location. Since the research has four experiments. The initial one is predicting the flow of traffic based on vehicle counts (Experiment 1) and the next experiment is to predict the traffic flow by analysing peak and off-peak times (Experiment 2), and finally analysing the traffic flow using geospatial metrics to Trafalgar Square (Experiment 3).

1.3 Research Objectives

The objective of this study is to employ deep learning models to enhance the predictability and understanding of traffic flow dynamics. To achieve the research question, the objective for the study is defined below in Table 1.

Table 1: Research Objectives

Objectives	Methodology	Evaluation Metrics
Objective 1	Data Preparation and integration of traffic flow data into a PostgreSQL database.	-
Objective 2	Data Cleaning and transformation using pandas data frame to ensure data quality and consistency.	-
Objective 3	Feature extraction, Normalisation, and data stationarity test.	Normality, skewness, kurtosis, and ADF test.
Objective 4	Development and Optimization of LSTM, GRU, and CNN models for traffic flow analysis.	MAE, MSE, RMSE, and R^2 score for LSTM, GRU and CNN models.
Objective 5	Visualisation and analysis of model predictions and traffic patterns using data visualisation libraries to extract actionable insights.	-

2. Related Work

Accurate traffic flow forecast is critical to Intelligent Transportation System (ITS) pursuit of effective traffic management. This multifaceted problem includes not just anticipating vehicle movement and speed but also balancing road safety in different traffic situations, like peak and off-peak times. To address the innate complexity and stochastic character of traffic, this literature review explores the state-of-the-art approaches that have been created to improve traffic prediction accuracy and reliability. It examines the application of non-parametric models, which get around the drawbacks of conventional parametric approaches and provide a more flexible and data-driven method of understanding traffic dynamics. Also, it discovers the pivotal role of deep learning in preventing overfitting, which is a major obstacle when modelling complex systems, and guaranteeing strong predictive performance. Lastly, it considers the emerging domain of data driven ITS, which combines various data sources and analytics to build a traffic infrastructure that is more intelligent and responsive. The subsequent sections 2.1,2.2, and 2.3 will explore these three critical strands, non-parametric models, overfitting in deep learning, and data-driven methods in ITS. Table 2 shows the overview of key studies in Intelligent Transportation System and Traffic stream prediction.

2.1 Non-Parametric Models for Traffic Flow Prediction

The fact that it facilitates the handling of non-linear and multi-variate data, as well as generalization are the key benefits of using neural network models. The Back Propagation (BP) algorithm's continual training is another benefit of these models. In recent years, deep learning networks have been used by researchers in the field of traffic flow forecast. The study by (Huang, et al., 2014) combined deep belief network (DBN) with multitask learning for traffic flow prediction, demonstrating an inventive strategy that departed from traditional shallow architectures and hand-engineered features, marking a significant advancement. This seminal study opened the door for more research into deep learning models, as evidenced by (Lv, et al., 2015). They used a stacked autoencoder model to attain both temporal and geographical correlations in traffic flow data. Their approach of using deep architecture model using autoencoders set a new model in the field. Also, the research conducted by (Zhene, et al., 2018) on urban traffic flow prediction used a deep learning architecture based on CNN and RNN, emphasizing the model's expertise in handling high-dimensional spatial-temporal characters.

The use of additional research enhances the story even further, (Kang, et al., 2017) used LSTM recurrent neural networks to forecast short-term traffic flow, showcasing the model's capacity to incorporate many traffic factors like flow, speed, and occupancy. The study emphasized the importance of LSTM in capturing the non- linear and stochastic characteristics of complex transportation systems. In addition, (Fu, et al., 2016) compared the traffic flow predictions capabilities of GRU and LSTM neural network techniques. Their findings shown that deep learning models like LSTM and GRU overtaken the traditional models like ARIMA in predicting short-term traffic flow. Continuing this trend, (Yu, et al., 2019) investigates the advantage of LSTM networks to forecast short-term traffic flow, highlighting the model's excellent accuracy and versatility in a range of traffic scenarios. Additionally, (Zhao, et al., 2019) shows how deep learning algorithms can adapt and optimize in response to urban traffic circumstances by presenting an optimized GRU algorithm to forecast truck travel speed under non recurrent congestion. These studies show how non-parametric, deep learning models are becoming more and more useful for predicting traffic flow and providing more complex, accurate, and real-time solutions than their predecessors.

2.2 Addressing Overfitting in Deep Learning

In deep learning, overfitting is a major problem, especially for difficult problems like traffic flow prediction where models run the risk of memorization rather than generalization from the data. Understanding this problem, researchers have created several methods to reduce overfitting, ensuring the stability and generalizability of deep learning models. For instance, (Liang & Liu, 2015) presented a novel method that effectively improved model performance during the fine-tuning phase by combining dropout approaches with stacked denoising encoders. It was demonstrated that this approach outperformed the use of singular dropout, indicating a breakthrough in the training of deep neural networks. Similarly, (Duan, et al., 2018) used a greedy algorithm to build an enhanced deep hybrid network for forecasting urban traffic flow with the goal of reducing computation time and increasing accuracy as network depth rose. Recent studies continue to contribute to this field, like the (Deshmukh, et al., 2023) used image processing techniques and data mining, by analysing the effectiveness of several machine learning algorithms, like KNN, SVM, Random Forest, and DNN in forecasting traffic flow and patterns. Their approach successfully decreased overfitting and increased accuracy. Additionally, utilizing the Taguchi approach, (Yang, et al., 2017) presented a unique model like stacked autoencoder Levenberg-Marquardt model for learning traffic flow aspects and optimizing structure. When this model was applied to real-world data, it outperformed excellent performance in traffic flow prediction compared to other predictors. These studies highlight the importance of addressing overfitting in deep learning models and showcasing novel strategies that enhance prediction accuracy while preserving computation efficiency.

2.3 Data Driven Approaches in Intelligent Transportation Systems

Data-driven approaches, which make use of the vast amount of data produces by urban traffic networks, have had a significant impact on the enhancement of Intelligent Transportation Systems (ITS). This shift is highlighted in a pivotal survey by (Zhang, et al., 2011), which describes the evolution from traditional technology-driven systems to multifunctional, data-driven ITS, driven by the increasing availability of data from multiple sources. This advancement has enabled more effective, intelligent transportation solutions. In line with this idea, (Chen, et al., 2021) underlined how real-time data collection can improve traffic management and lessen environmental effects by showing the critical role of deep learning in ITS, particularly in traffic flow detection. (Chen, et al., 2021), their innovative approach to traffic flow detection, which leverages deep learning and is optimized for edge computing, effectively tackles the notable challenges presented by conventional cloud computing-oriented systems.

The integration of social media data into ITS shows an innovative method of traffic research and forecasting. (Shekhar, et al., 2016) investigated this new field and offered a low-cost substitute for conventional traffic monitoring system, by putting out a framework that makes use of social media updates for real-time traffic congestion monitoring. Their method, which predicts traffic patterns and pinpoints congestion sources using sentiment analysis, showcases the promise of utilizing social media in urban traffic management. Furthermore, the research conducted in (Shekhar, et al., 2016) provides a distinctive viewpoint on the utilization of social media platforms for traffic pattern prediction in the context of vehicular traffic analysis. Similarly, (Moses & Parvathi, 2020) highlighted the importance of data-driven solutions in making traffic expectable, hence saving time and energy in daily life, in their research on vehicular traffic assessment and prediction using machine learning procedures. These studies collectively show how data-driven methods in ITS are becoming more widely used and provide a more comprehensive and integrated picture of urban traffic management.

Table 2: Key Studies in ITS & Traffic Flow Prediction

Authors	Study	Year	Methodologies/Model
(Huang, et al., 2014)	Traffic flow prediction with multitask learning	2014	Deep Belief Network (DBN)
(Lv, et al., 2015)	Geographical and temporal correlations in traffic	2015	Stacked autoencoder
(Zhene, et al., 2018)	Urban traffic flow prediction	2018	CNN and RNN
(Kang, et al., 2017)	Short-term traffic flow prediction	2017	LSTM recurrent neural networks
(Fu, et al., 2016)	Comparing LSTM and GRU for traffic flow prediction	2016	GRU and LSTM neural networks
(Yu, et al., 2019)	Short-term traffic flow prediction	2019	LSTM networks
(Zhao, et al., 2019)	Forecasting truck travel speed	2019	Optimized GRU algorithm
(Liang & Liu, 2015)	Reducing overfitting in deep learning	2015	Dropout approaches with stacked denoising encoders
(Duan, et al., 2018)	Enhanced deep hybrid network for traffic prediction	2018	Greedy algorithm for deep hybrid network construction
(Deshmukh, et al., 2023)	Forecasting traffic flow and patterns	2023	Data mining and image processing techniques
(Yang, et al., 2017)	Learning traffic flow aspects	2017	Stacked autoencoder Levenberg-Marquardt model
(Zhang, et al., 2011)	Evolution of ITS from technology to data-driven	2011	Survey study
(Chen, et al., 2021)	Traffic flow detection in ITS	2021	Deep learning optimized for edge computing
(Shekhar, et al., 2016)	Traffic monitoring using social media	2016	Sentiment analysis and social media data
(Moses & Parvathi, 2020)	Prediction and Vehicular traffic analysis	2020	Machine learning algorithms

3. Methodology

The extensive dataset, enriched with traffic flow are obtained from UK Government website. The raw traffic data of minor and major roads collected via a network of sensors, provides a view of the nation's traffic dynamics. To gather meaningful insights from this large body of data, a Knowledge Discovery in Databases (KDD) process, a systematic framework that directs the conversion of raw data into valuable knowledge. The integrity and usability of data is ensured in the initial data cleaning to refined analytics stage. Finally, deep learning models are assigned to gather understandings from the data, by integrating the principles of KDD methodology. The subsequent sections explain the modified KDD approach.

3.1 Traffic Flow Prediction Methodology

Knowledge Discovery in Databases (KDD) methodology has six important stages as presented in Figure 1. The first stage of methodology starts with a comprehensive understanding of traffic data. It includes the data requirements, type of storage and suitable models required for the prediction of traffic flow. The second step procedure involves cleaning of data to remove unwanted information, omitted values, renaming column names, and eliminating unnecessary columns.

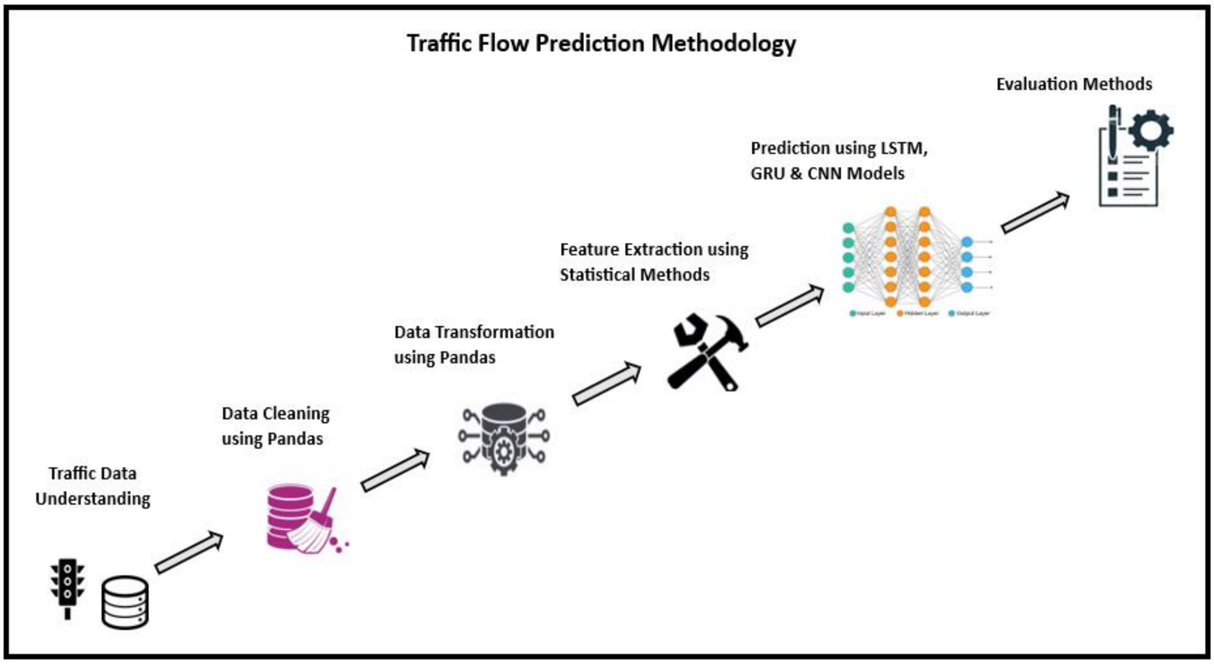


Figure 1: Methodology for Traffic Flow Prediction

The cleaned data is then transformed according to the requirements, such as encoding of categorical values to numerical, creating new columns that would help in the prediction, and filtering records. After the transformation stage, the data is then stored in the PostgreSQL database for more analysis. In the fourth stage, stationarity, normality, and seasonality of the data is analysed. The heart of the methodology lies in the application of deep learning models to capture the complex temporal and spatial dependencies within the traffic data. The final phase assesses the predictive models' performance through evaluation methods, ensuring that the predictions are both accurate and actionable. This iterative KDD process highlights a systematic style to showing patterns in traffic flow, aiding to the advancement of smarter, more effective transportation systems. The forthcoming sections explains the procedure of data understanding, data preparation, data cleaning, transformation, and feature extraction in detail.

3.2 Traffic Data Understanding

In this study, the department for transport gathers traffic data in Great Britain, including both major and minor roads. The data provides vital information such as vehicle characteristics, traffic volume during peak and off-peak times, as well as precise geographical data using longitude and latitude coordinates. Geographical aspect of the data is exceptionally noteworthy as it allows for the calculation of distances from Trafalgar Square, which serve as a valuable reference point. This comprehensive and multi-faceted traffic dataset makes a substantial contribution to transportation research and policymaking efforts by providing crucial resources for understanding and evaluating traffic patterns and trends in Great Britain.

3.3 Data Acquirement

Dataset utilized for the research is ethically obtained from UK government website[1]. The data file is downloaded from the UK government's open data portal in Comma Separated Values (CSV) format. Specifically, the Transport Department collects broad traffic data to deliver detailed statistics on the level of traffic across roads in Great Britain. Research specialists and students can conduct experiments and contribute to developments using the open data. The raw traffic count of major and minor roads in the United Kingdom have data of from 2000. Data from 2017 to 2021 which has 10,99,968 are considered. Acquired dataset serve as a foundational element for the KDD process, providing the raw data from which valuable knowledge is to be extracted.

3.4 Data Preparation, Cleaning and Transformation

The initial phase of the research is data pre-processing, which includes data preparation, data cleaning, and transformation. In data cleaning, missing values and null values are dropped. Transformation of data and storing the transformed data in a database are taken out in data transformation stage. Feature extraction is performed using numerous statistical analysis. In experiment 1,2 and 3 traffic flow data is utilized.

3.4.1 Data Cleaning and Transformation for Experiment 1

In Experiment 1, the data cleaning and transformation plays a crucial role in shaping the dataset for the analysis of vehicle influence on traffic flow. Utilizing Pandas in Python environment, a series of steps are executed, initiating with the purging of null values to preserve the integrity of vehicle and geographical attributes, and removal of irrelevant columns that did not contribute to the analysis, which leads to a refined dataset ready for transformation. The 'Count_date' column is converted to datetime format enabling to perform time-series analysis. A new column 'total_vehicles' is generated by aggregating various vehicle type counts. In the next step, one-hot encoding is applied on categorical variables like 'Direction_of_travel', 'Road_category', 'Road_type', and 'Region_name'. After dropping null values and filtering records, the traffic dataset is decreased to 3,73,860 with zero duplicates. Finally, the cleansed transformed data is stored in a PostgreSQL database in a table 'Exp1', administered through pgAdmin 4 in local system.

[1] https://www.data.gov.uk/dataset/208c0e7b-353f-4e2d-8b7a-1a7118467acc/gb-road-traffic-counts

3.4.2 Data Cleaning and Transformation for Experiment 2

In Experiment 2, the temporal impact of hourly fluctuations on traffic patterns are considered. The null values are removed from columns like "Link_length_km' and 'Link_length_miles', and the columns not relevant to the study, such as 'Start_junction_road_name' and 'End_junction_road_name' are removed. In the remaining dataset, median values are imputed to the missing values to preserve the data's integrity, mainly when skewed distributions are present. Post-cleaning, the 'Count_date' column is converted into a usable datetime format. A new column 'peak_non_peak_hour' is created to categorize hours into peak and non-peak periods, enhancing analytical granularity. Categorical variables like 'Direction_of_travel', 'Region_name', 'Road_category', 'Road_type', 'peak_non_peak_hour' are encoded using one-hot encoding. After cleaning and filtering records, the dataset has 10,99,968 data. The transformed data is then stored in PostgreSQL in a table named 'Exp2'.

3.4.3 Data Cleaning and Transformation for Experiment 3

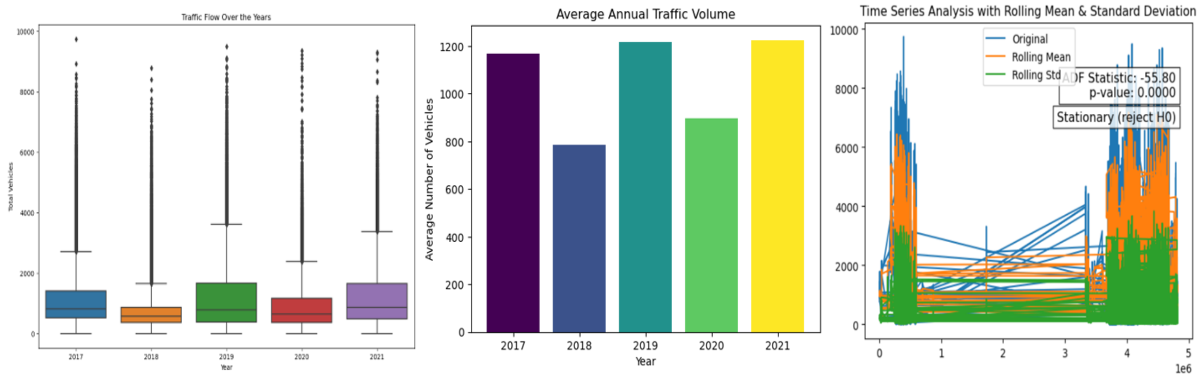
For Experiment 3, a rigorous data cleaning process on dataset is conducted to assess the influence of geographical location on traffic volume. Initially, missing values in junction names and link lengths were addressed by removal or median imputation, respectively. The 'Count_date' column is converted to a datetime format and extracted 'month' for temporal analysis. Moreover, a new column 'distance_to_central_london' is created by determining the distance of each data point from Trafalgar Square, which helps to investigate traffic volume in relation to central landmark. One-hot encoding is applied on categorical variables, after ensuring there were no duplicate records. After filtering and transforming, the dataset has 10,99,968 records. The cleaned and transformed data is stored in PostgreSQL database under a table "Exp3', enhancing the dataset's compatibility with the predictive modelling techniques to decipher traffic patterns across the urban landscape.

3.5 Feature Extraction

After PostgreSQL data storage, features required for data mining are developed through statistical analysis of Augmented Dickey-Fuller (ADF) test , normality test, skewness, kurtosis, and traffic flow trends over time.

3.5.1 Statistical Analysis for Experiment 1

Sqlalchemy build engine reads data from PostgreSQL and stores it in a pandas data frame. The Shapiro-Wilk test is conducted to verify data normality. Normality test is examined for each numeric column within the dataset using stats module from Scipy. The data appears to be regularly distributed or gaussian, the null hypothesis (p-value > 0.05) is rejected (fail to reject H0). If p-value < 0.05, then data does not look gaussian (reject H0). The resultant p-value from the dataset is 0.000, and data is not normally distributed (reject H0). Shape of the distribution is determined by Skewness and kurtosis. The data has skewness of 2.15 and kurtosis of 5.66 which defines the data is skewed on the positive side of the distribution. As shown in Figure 2a**,** the impact of vehicles over different years are represented in boxplot. The visualisation depicts that the year 2019 and 2021 has the highest traffic volume influenced by the number of vehicles. The plot is helpful in revealing trends and variations in traffic volume over the years, which could be important for transportation analysis and forecasting.



(a) Traffic flow based on vehicles (b) Annual traffic volume based on vehicles (c) Data Stationarity Test

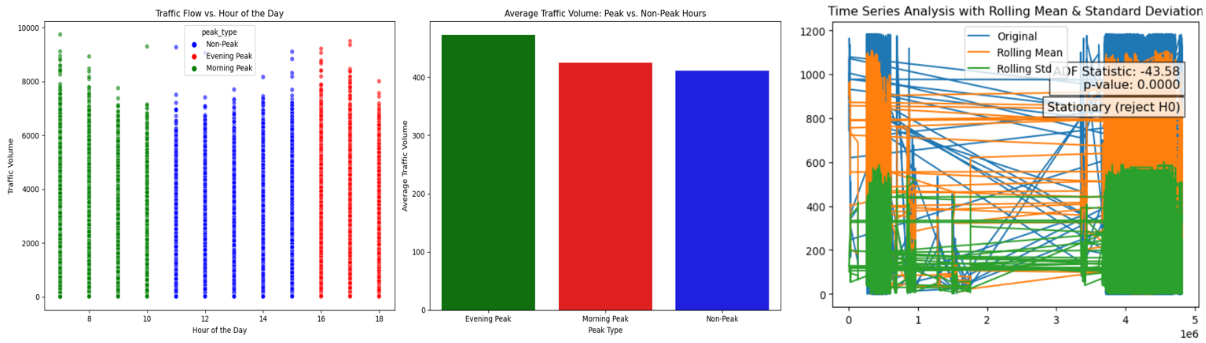
Figure 2: Statistical Analysis - Experiment 1

To check the stationarity of the data, Augmented Dickey-Fuller (ADF) test is conducted. From the Figure 2c, the results of the test are with a very negative statistic and a p-value of 0.0 (p value = 0.0 < 0.05), indicate that the time series is stationary, as the null hypothesis of a unit root can be rejected at all common significance levels.

3.5.2 Statistical Analysis for Experiment 2

In Experiment 2, normality, skewness, and kurtosis test was conducted like experiment 1. To assess the distribution of data, Shapiro-Wilk test is applied. The data is not normal, and the null hypothesis (H0) of the normality test is rejected with a p-value of 0.00. At first, the skewness and kurtosis were 3.48 and 15.56, both indicating a heavy tailed and asymmetric distribution, indicating the presence of outliers. After the removal of outliers, skewness and kurtosis value was reduced to 1.48 and 1.40, suggesting a more symmetric distribution and it shows the data is skewed on the positive side of distribution. The visualisations include a scatter plot contrasting traffic flow against hours of the day, categorized by morning peak, evening peak and non-peak, and a bar plot showing the average traffic volume for each peak type. As indicated in Figure 3b, the average traffic volume during the evening peak hours is depicted as higher than the other two peak type, where evening peak period experiences the heaviest traffic flow in the city.

To check the data stationarity, Augmented Dickey-Fuller (ADF) test is conducted. As shown in Figure 3c, the test resulted in a p-value of virtually 0, far below the alpha level of 0.05, showing that the time series is stationary and suitable for time series predicting models that assumes stationarity.



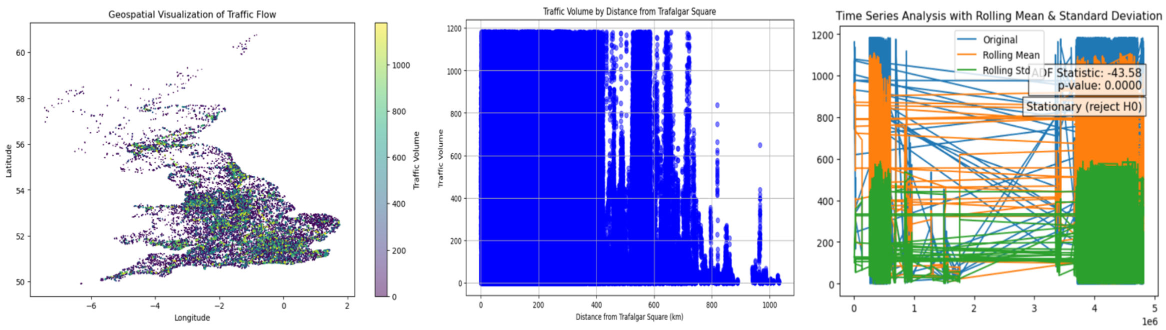
(a) Traffic flow vs. Hour of the day (b) Annual traffic volume peak vs. non-peak (c) Data Stationarity Test

Figure 3: Statistical Analysis - Experiment 2

3.5.3 Statistical Analysis for Experiment 3

In Experiment 3, Shapiro-Wilk test is performed to assess the normality of the data, indicating that the data does not follow a normal distribution. Skewness and kurtosis values reveal that the data distribution with heavy tails, indicating the presence of outliers. The outliers are then identified using the interquartile range method and after the removal of outliers the values for skewness and kurtosis are reduced to 1.48 and 1.40. A vivid visual representation of traffic volume across different geographical location is plotted using latitude and longitude as shown in Figure 4a**.** Traffic volume is plotted against the distance from Trafalgar Square which shows the distribution of traffic volume over the city in Figure 4b.

The Augmented Dickey-Fuller test is used to check the data stationarity, showing a significant ADF statistic and a p-value of zero, which explains the stationarity of the time series data. As shown in Figure 4c rolling mean and standard deviation of the traffic data is plotted, revealing the trends and volatility over time.



(a) Geospatial visualisation of traffic flow (b) Traffic volume from Trafalgar Square (c) Data Stationarity Test

Figure 4: Statistical Analysis - Experiment 3

4. Design Specification

As depicted in Figure 5, the proposed architecture has three layers of data storage layer, data analytics layer and data visualisation layer. The design subscribes to an iterative development paradigm, permitting continuous refinement and adaptation according to the conditions of the data. In the data storage layer, raw input csv files of traffic data are stored in CSV file format.

This layer ensures a uniform format for data retrieval and ensures data persistence. Additionally, the large traffic dataset is managed and queried using a robust and scalable relational database management system PostgreSQL. It is a dependable database management system that presents a safe and effective means of handling structured data, thereby serving as a scalable solution for storing traffic dataset with high transactional requirements. The following layer, given to data cleaning and transformation, is where the raw data in csv format is cleaned and transformed into a form open to analysis. Here, Pandas, a library created specifically for data cleaning and manipulation, is used in tandem with Python, which is known for its adaptability and the depth of its libraries. This makes it possible to carry out data preprocessing tasks, like managing missing values, removal of outliers, and standardizing data formats. So, the final cleaned and transformed data is stored on PostgreSQL. Statistical tests of Dickey-Fuller (ADF), skewness, kurtosis, and normality are performed at the analytics layers using SciPy, statsmodel, and sklearn. The main purpose of this study is to forecast the traffic flow in the United Kingdom.

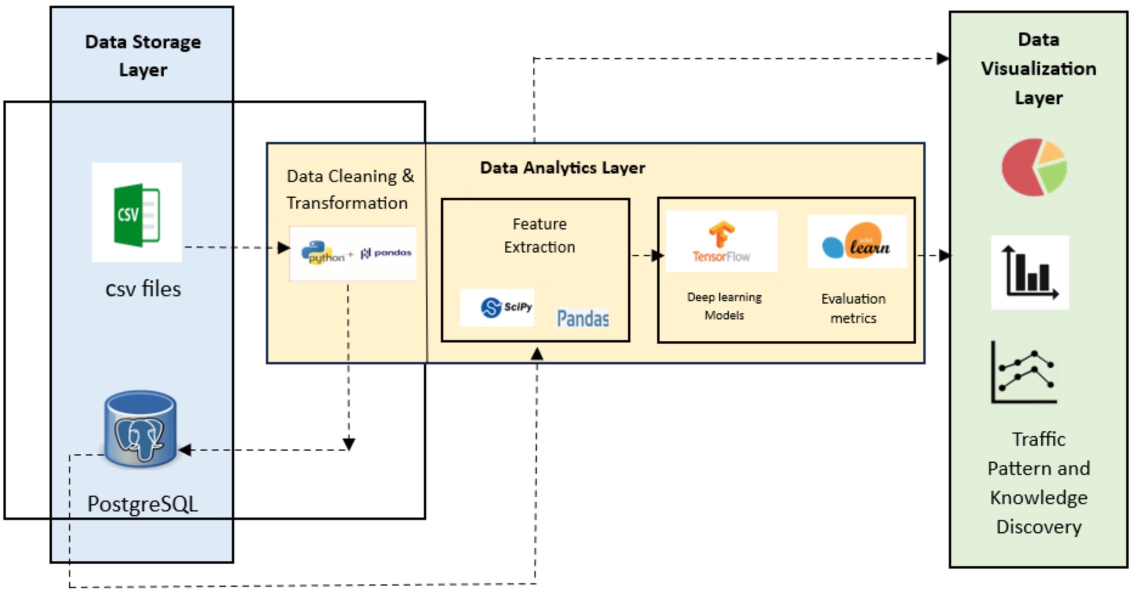


Figure 5: Design Specification

Multiple deep learning networks like LSTM, GRU, and CNN are implemented utilizing Keras and TensorFlow packages. Evaluation of each model is implemented using Scikit-learn package. The final layer in the architecture is the Data visualization layer, which is used to visualize the analysis, normality, and different traffic flow characteristics. This layer is not only about presentation, but also an exploratory tool in its own, enabling the discovery of insights that might otherwise remain hidden in the complexity of raw data. The coming section describes the implementation of the models, performance tuning, and evaluation methods performed in the study.

5. Implementation of Traffic Flow Prediction

The implementation of this study is conducted out in three experiments. The initial part of the research is predicting the traffic flow based on the vehicle counts (Experiment1). The second part of the research is to predict the traffic flow by examining the peak and off-peak hour (Experiment 2). The final part of the research is to analyse the traffic flow based on the geographical proximity to Trafalgar Square (Experiment 3). Experiment 1, Experiment 2 and Experiment 3 are conducted out using various deep learning networks like LSTM, GRU, and CNN.

5.1 Data Preparation

Although data are retrieved and redirected using pandas and statistical tests, pre-processing is still necessary before data can be entered into deep learning models. For all the three experiments, data preparation is common. The initial stage is to normalize the data using sklearn from the data frame. Normalisation of the data is performed using MinMaxScaler normalization method. The next step is to train and test the model, after spliting the data into training and testing set. During the model fit, a portion of testing data is utilized as validation. In all the three experiments, data is divided into 80% of training and 20% of testing using train test split from sklearn.

5.2 Predicting Traffic Flow Dynamic from Vehicle Counts (Experiment 1)

To predict the flow of traffic based on the influence of vehicle counts, three distinct deep learning models are implemented. The LSTM model used a sequence of layers starting with an LSTM layer with 50 neurons, a dropout layer set at 0.2 to prevent overfitting, and a denser layer with a single output unit, totalling 13051 parameters that are all trainable as shown in Figure 6 shows. The GRU model architecture is same as that of LSTM model by using a GRU layer in place of the LSTM layer. As depicted in Figure 6, GRU layer with 50 units for processing sequence data, a dropout layer to prevent overfitting, and dense layer with one unit. The model has 9,951 parameters all of which are trainable, showing that the model is relatively compact.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 50)                13000

 dropout_5 (Dropout)         (None, 50)                0

 dense_5 (Dense)             (None, 1)                 51

=================================================================
Total params: 13051 (50.98 KB)
Trainable params: 13051 (50.98 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_1 (GRU)                 (None, 50)                9900

 dropout_6 (Dropout)         (None, 50)                0

 dense_6 (Dense)             (None, 1)                 51

=================================================================
Total params: 9951 (38.87 KB)
Trainable params: 9951 (38.87 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 12, 64)            256

 max_pooling1d (MaxPooling1  (None, 6, 64)             0
 D)

 flatten (Flatten)           (None, 384)               0

 dense_7 (Dense)             (None, 50)                19250

 dropout_7 (Dropout)         (None, 50)                0

 dense_8 (Dense)             (None, 1)                 51

=================================================================
Total params: 19557 (76.39 KB)
Trainable params: 19557 (76.39 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 6: LSTM, GRU and CNN Model Summary - Experiment 1

Lastly, the CNN model implemented a different approach, employing a 1D convolutional layer using 64 filters and kernel size of 3 to detect patterns over the temporal sequence, a max pooling layer to reduce dimensionality, and a dense layer with dropout for prediction. Each model was compiled with the Adam optimizer and the loss function is Mean Squared Error (MSE). Finally, three models are fit for 50 training epochs with a batch size of 50. To avoid the overfitting of the training dataset, Early Stopping function of the TensorFlow package.

5.3 Predicting Traffic Flow by analysing Peak and Off-Peak Hour (Experiment 2)

In Experiment 2, prediction of the traffic flow through peak and non-peak hours, three different neural network architecture is implemented. The LSTM network with 50 units, which is well-known for its capacity to hold onto data over lengthy sequences, which is essential for understanding temporal dynamics in traffic flow. The model includes a dropout layer to avoid overfitting and has a total of 12,501 trainable parameters. The second model utilizes a Gated Recurrent Unit (GRU) layer, also with 50 units, offering a more parameter-efficient alternative to the LSTM, with a total of 9,201 trainable parameters. Lastly, the third model is a CNN network that consists of a dense layer, max pooling to reduce dimensionality, and a 1D convolutional layer with 64 filters and a kernel size of 3. The model is designed to detect spatial-temporal patterns in traffic flow with 9,957 trainable parameters, which might be an indicative of traffic congestion during different times of the day.

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 50)	12000
dropout (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51

Total params: 12051 (47.07 KB)
Trainable params: 12051 (47.07 KB)
Non-trainable params: 0 (0.00 Byte)

Model: "sequential_1"

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 50)	9150
dropout_1 (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 1)	51

Total params: 9201 (35.94 KB)
Trainable params: 9201 (35.94 KB)
Non-trainable params: 0 (0.00 Byte)

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 7, 64)	256
max_pooling1d (MaxPooling1D)	(None, 3, 64)	0
flatten (Flatten)	(None, 192)	0
dense_2 (Dense)	(None, 50)	9650
dropout_2 (Dropout)	(None, 50)	0
dense_3 (Dense)	(None, 1)	51

Total params: 9957 (38.89 KB)
Trainable params: 9957 (38.89 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 7: LSTM, GRU and CNN Model Summary - Experiment 2

All the three models were compiled with Adam Optimizer and loss function is Mean Squared Error (MSE). Each model was fit for 50 training epochs with a batch size of 50. Early Stopping function of TensorFlow is implemented to prevent overfitting of the training dataset.

5.4 Traffic Flow Analysis Using Geospatial metrics to Trafalgar Square (Experiment 3)

Traffic Flow analysis using Geospatial metrics near Trafalgar Square is implemented in Experiment 3. Same as that of Experiment 1 and Experiment 2, LSTM, GRU and CNN models are developed. The LSTM model, with 12,650 parameters efficiently processes sequential data by utilizing memory cells, capturing temporal relationships that are essential for predicting traffic flow. The GRU model, which has 9,650 parameters and is slightly compact, uses a streamlined gating mechanism to control sequence information while requiring fewer calculations, providing a performance-complexity trade-off. However, the CNN model consisting of 16,537 parameters applies convolutional layers to extract spatial information from geographical data, offering a unique approach to understanding traffic patterns. Each model is optimized to anticipate traffic levels by learning from data patterns in both geography and time, demonstrating the adaptability of neural networks in handling complex geospatial analyses.

Adam Optimizer is used to compile all the three models and Mean Squared Error as loss function. With a batch size of 50, the three models were fit for 50 training epochs. To avoid the overfitting of the model, early stopping function of TensorFlow package is implemented.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50)                12600

 dropout (Dropout)           (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 12651 (49.42 KB)
Trainable params: 12651 (49.42 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 50)                9600

 dropout_1 (Dropout)         (None, 50)                0

 dense_1 (Dense)             (None, 1)                 51

=================================================================
Total params: 9651 (37.70 KB)
Trainable params: 9651 (37.70 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 10, 64)            256

 max_pooling1d (MaxPooling1  (None, 5, 64)             0
 D)

 flatten (Flatten)           (None, 320)               0

 dense_2 (Dense)             (None, 50)                16050

 dropout_2 (Dropout)         (None, 50)                0

 dense_3 (Dense)             (None, 1)                 51

=================================================================
Total params: 16357 (63.89 KB)
Trainable params: 16357 (63.89 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 8: LSTM, GRU and CNN Model Summary - Experiment 3

6. Evaluation

The three experiments use various deep learning networks like LSTM, GRU and CNN, which are evaluated based on training and testing R- squared value, Mean Squared error, Root Mean Square error, and Mean Absolute Error. Training and validation loss versus epochs are also discussed.

6.1 Evaluation for Experiment 1

The performance metrics of experiment 1 indicate a strong predictive capability across all the three deep learning models tested. The LSTM and GRU models explained their exceptional fit to the data by achieving nearly identical high R-squared value of 0.9972 for training and 0.9975 and 0.9988 for testing, respectively. The GRU model demonstrated better accuracy in traffic flow prediction as its ability to minimize the test RMSE to 34.40, outperforming the LSTM model's test RMSE of 51.12. With a training R-squared of 0.9923 and a testing R-squared of 0.9926, the CNN model showed a slightly lower fit, despite having a noticeably higher RMSE. Early Stopping function used in the CNN model fit has prevented the model from overfitting with appropriate hyper-parameters at 7th epoch. This could indicate the CNN's potential in capturing spatial-temporal features, albeit less precisely than the other two models in this specific scenario.

Table 3: Experiment-1 Model Evaluation

Model	Train R^2	Test R^2	Train MSE	Test MSE	Train RMSE	Test RMSE	Train MAE	Test MAE
LSTM	0.9972	0.9975	2866.854	2613.619	53.543	51.123	9.953	9.807
GRU	0.9972	0.9988	2866.854	1183.660	53.543	34.404	9.953	8.474
CNN	0.9923	0.9926	8045.519	7782.875	89.696	88.220	41.067	40.736

The comparison of the number of epochs of training and validation losses shown in Figure 9**.**

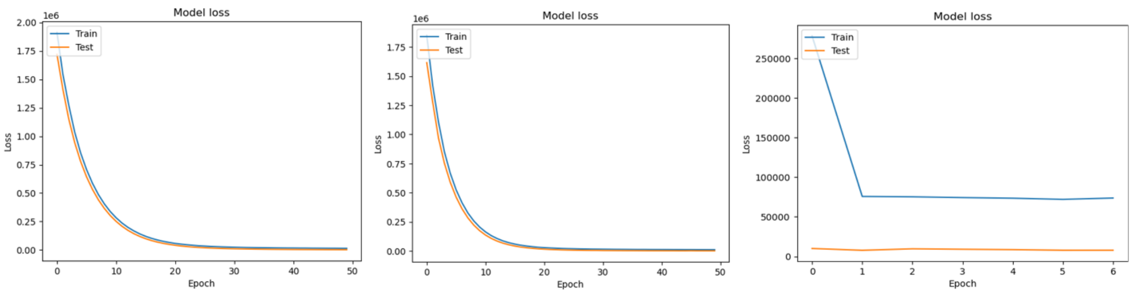


Figure 9: Loss Vs Epochs - Experiment 1

6.2 Evaluation for Experiment 2

In Experiment 2, the LSTM and GRU models demonstrated outstanding consistency and predictive accuracy, both achieving a training and test R-squared of 0.9963 as shown in Table 4**.** The exceptionally close train and test RMSE values for both models indicate accurate traffic flow predictions without overfitting, therefore demonstrated a high level of performance. The CNN model shown slightly lower R-squared of 0.9884 for training and 0.9882 for testing, still performed well. But, in contrast to the GRU and LSTM models, the CNN's higher RMSE values indicates a less accurate prediction. These metrics were reflected in the loss plots, which exhibit quick convergence early in training and indicating the models' quick adaptability to the underlying patterns in the data. The temporal dynamics of traffic flow during peak and non-peak hours were particularly well-captured by the LSTM and GRU models, while the CNN's capabilities might lie in extracting spatial patterns that the LSTM and GRU models may not fully capture. Early Stopping function used in three model fit has halted the model from overfitting at 31st epoch for LSTM, 38th epoch for GRU, and 39th epoch for CNN.

Table 4: Experiment-2 Model Evaluation

Model	Train R^2	Test R^2	Train MSE	Test MSE	Train RMSE	Test RMSE	Train MAE	Test MAE
LSTM	0.9963	0.9963	279.606	280.269	16.721	16.741	9.281	9.280
GRU	0.9963	0.9963	279.606	273.705	16.721	16.544	9.281	9.106
CNN	0.9884	0.9882	881.934	893.475	29.697	29.891	18.432	18.529

As shown in Figure 10**,** the loss graphs indicate that all models picked up on the ability to anticipate traffic flow. A steep decline in loss with the initial epochs, followed by stabilization, indicating that the models converged quickly to a solution. The steady closeness of the train and test loss lines in each models' plot demonstrated good generalization without overfitting across the epochs.

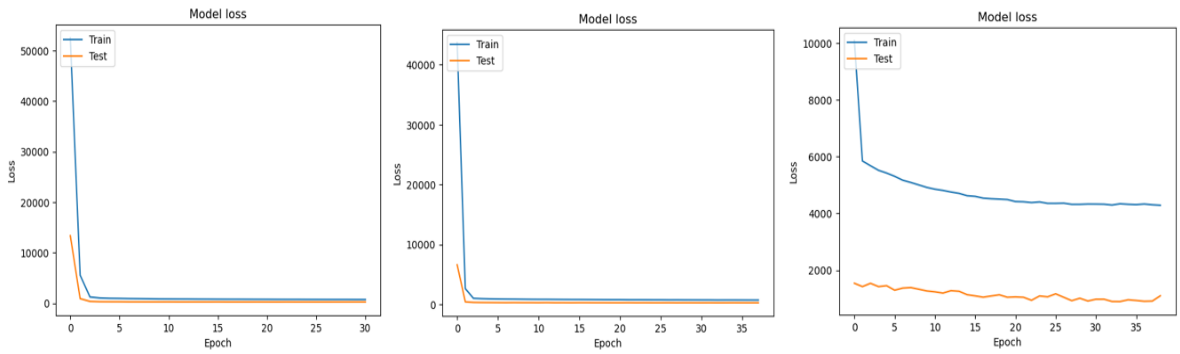


Figure 10: Loss Vs Epochs - Experiment 2

6.3 Evaluation for Experiment 3

In Experiment 3, LSTM, GRU, and CNN models are evaluated to predict the traffic volume on geographical data. The LSTM and GRU models performed almost identical performance with high R-squared values above 0.9963 for both training and testing datasets, indicating good model fits as shown in Figure 11. However, the test MSE for the GRU model was slightly higher, suggesting a little higher prediction accuracy. On the other hand, CNN model demonstrated a weaker ability to capture the geographic dependencies in the data when compared to recurrent models, as shown by its lower R-squared values of 0.9939 for training and 0.9937 for testing, with higher MSE and RMSE values. LSTM, GRU and CNN models used early stopping function to prevent the model from overfitting with the appropriate hyper-parameters at 42nd epoch for LSTM, 29th epoch for GRU, and 12th epoch for CNN.

Table 5: Experiment-3 Model Evaluation

Model	Train R^2	Test R^2	Train MSE	Test MSE	Train RMSE	Test RMSE	Train MAE	Test MAE
LSTM	0.9963	0.9963	279.494	280.500	16.718	16.748	9.384	9.385
GRU	0.9963	0.9964	279.494	267.387	16.718	16.351	9.384	9.261
CNN	0.9939	0.9937	463.780	471.836	21.535	21.721	13.726	13.773

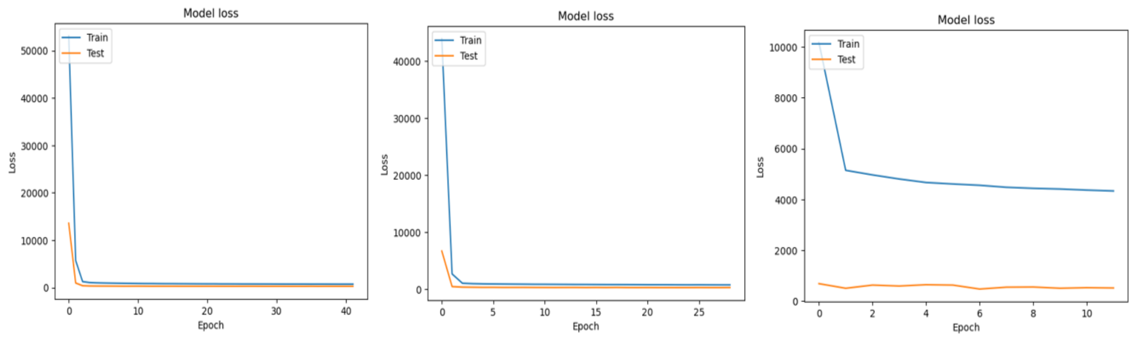


Figure 11: Loss Vs Epochs - Experiment 3

From the Figure 11, the first plot indicates a rapid convergence of the model, with a dramatic initial decrease in both training and validation loss, followed by a plateau. This implies a good fit with minimal overfitting. The second and third charts exhibit a similar pattern of quick convergence and stability of model loss.

6.4 Discussion

Using various deep learning models, the three experiments conducted offered diverse insights on the dynamics of traffic flow. These experiments helped to clarify the effects of vehicle counts, time-of-day fluctuations, and geographic metrics on traffic flow forecast.

Predicting Traffic flow Dynamic from Vehicle Counts in Experiment 1, deep learning networks like LSTM and GRU proved exceptional performance with nearly identical results, which attributes to their ability to identify sequential dependencies in the vehicle count data. The CNN model did not perform as well as the recurrent models, which suggests that CNN's strength in capturing spatial characteristics may not benefit in time series forecasting of traffic flow based on vehicle counts. For further improvement in experiment 1 would be incorporating additional features such as weather conditions and historical traffic patterns may improve the prediction accuracy.

Using Peak and non-peak hour analysis to predict traffic flow (Experiment 2) highlighted a similar pattern, with the LSTM and GRU models performing similarly to each other but with a slightly reduced accuracy compared to Experiment 1. This indicates that even while the models are reliable, the unpredictability introduced by including data from peak and non-peak hours may not be fully captured by the models used. An improvement to this experiment is that using models that can distinguish between peak and non-peak patterns more clearly, with the help of stateful LSTM layers or attention mechanisms that give the peak hours more weight.

The inclusion of geospatial metrics seems to have aided the CNN model significantly, as its performance came close to that of the other two models. This is likely because CNN has a built-in capacity for effectively handling geographical data. But the recurrent models performed slightly well with GRU showing a marginally better result when compared to LSTM. To improve these results, a hybrid model combining recurrent and convolutional layers could be applied to leverage both temporal and spatial dependencies in traffic flow prediction.

From Tables 3,4, and 5, GRU model consistently showed better performance or outperformed other models, specifically in the context of MSE and RMSE on the test sets, which is important for evaluating the prediction accuracy of continuous variables like traffic flow. This clearly explains that the GRU model has a balanced capability in handling both the temporal and spatial data and the additional complexities introduced in Experiment 2 and Experiment 3.

Table 6: Best Model Selection Evaluation

Experiment	Best Model	Test R^2	Test RMSE	Test MAE
Predicting traffic flow dynamic from vehicle counts	GRU	0.9988	34.404	8.474
Predicting traffic flow by analysing peak and non-peak hour	GRU	0.9963	16.544	9.106
Traffic flow analysis using geospatial metrics to Trafalgar Square	GRU	0.9964	16.351	9.261

Table 5 summarizes the results of three experiments designed at predicting traffic flow, where the GRU model consistently outperformed all the other models across all metrics, showing its robustness in capturing complex patterns in data.

In light of these findings, future research should discover the combination of recurrent and convolutional architectures with the integration of more varied data sources. Furthermore, considering the trade-off between interpretability and model complexity is crucial. For instance, simpler models like GRU give acceptable performance with potentially lower computational costs and easier model tuning. These experiments emphasize the significance of selecting a model based on the specific features of the dataset and the prediction task at hand, which adds to the growing body of knowledge in traffic flow prediction. It is vital to keep investigating the subtleties of model performance under even more diverse traffic conditions.

7. Conclusion and Future Work

In this research, the LSTM, GRU, and CNN models were used to improve traffic flow prediction. The objective was to determine the best model in predicting traffic dynamics using vehicle counts, identify flow patterns during peak and non-peak hours, and analysing traffic around Trafalgar Square using geospatial data. In all the three experiments, the GRU model emerged as the superior performer, consistently achieving high accuracy across all experiments, suggesting it as a potential solution for real-time traffic prediction and management systems. Key findings indicate that, although all models were able to approximate traffic patterns to some extent, the GRU model stood out for its ability to combine efficiency and predictive capability, particularly in handling temporal data variations and complex spatial-temporal relationships. These predictions have significant implications for intelligent transportation systems and urban planning, where accurate traffic predictions can result in less traffic and more effective use of the roads. One limitation of the current research is that deep learning models for urban traffic management often battle with adaptability to dynamic and unpredictable environmental conditions, like weather changes, road accidents, and variable traffic flows. This constraint questions their reliability and practical effectiveness, as these models may not respond accurately in real-time to scenarios that differ from their training data.

Future work could concentrate on improving these models for more extensive urban contexts and investigating the incorporation of additional data sources such as social media trends, large-scale events, and environmental changes. The application of transfer learning could also be used to adapt models trained in one city to another, improving scalability and reducing the requirement for large amounts of localized data. These models have the potential to be commercially incorporated into smart city infrastructures, offering a predictive traffic management system, optimization for emergency services, and dynamic toll pricing. These improvements could significantly contribute to the development of more resilient and adaptive urban transportation systems.

Acknowledgement

To start with, I would want to sincerely thank the School of Computing, National College of Ireland, Dublin, for helping me attain my dream as a Data Analytics student. Additionally, I want to express my gratitude to Mr. Taimur Hafeez, my research supervisor, for his regular support in providing guidance on technical knowledge, and general assistance during the study. In conclusion, I express my gratitude to my parents, teachers, and friends for their unwavering support and motivation.

References

Chen, C. et al., 2021. An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System. *IEEE Transactions on Intelligent Transportation Systems,* pp. 1840-1848.

Deshmukh, S. et al., 2023. Machine Learning Algorithm Comparison for Traffic Signal: A Design Approach. *8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India,* pp. 1107-1113.

Duan, Z. et al., 2018. Improved Deep Hybrid Networks for Urban Traffic Flow Prediction Using Trajectory Data. *IEEE Access.*

Fu, R., Zhang, Z. & Li, L., 2016. Using LSTM and GRU neural network methods for traffic flow prediction. *31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China,* pp. 324-328.

Huang, W., Song, G., Hong, H. & Xie, K., 2014. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning. *IEEE Transactions on Intelligent Transportation Systems,* p. 2191–2201.

Impedovo, D., Balducci, F., Dentamaro, V. & Pirlo, G., 2019. Vehicular Traffic Congestion Classification by Visual Features and Deep Learning Approaches: A Comparison.

Kang, D., Lv, Y. & Chen, Y.-y., 2017. Short-term traffic flow prediction with LSTM recurrent neural network. *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan.*

Kothai, G. et al., 2021. A New Hybrid Deep Learning Algorithm for Prediction of Wide Traffic Congestion in Smart Cities.

Liang, J. & Liu, R., 2015. Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. *8th International Congress on Image and Signal Processing (CISP).*

Li, Z., Li, C., Cui, X. & Zhang, Z., 2021. Short-term Traffic Flow Prediction Based on Recurrent Neural Network. *International Conference on Computer Communication and Artificial Intelligence (CCAI), Guangzhou, China,* pp. 81-84.

Lv, Y. et al., 2015. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems,* pp. 865-873.

Moses, A. & Parvathi, R., 2020. Vehicular Traffic analysis and prediction using Machine learning algorithms. *International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India.*

Razali, N. A. M. et al., 2021. Gap, techniques and evaluation: traffic flow prediction using machine learning and deep learning. *J Big Data 8, 152.*

Shekhar, H., Setty, S. & Mudenagudi, U., 2016. Vehicular traffic analysis from social media data. *International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India,* pp. 1628-1632.

Shu, W., Cai, K. & Xiong, N. N., 2022. A Short-Term Traffic Flow Prediction Model Based on an Improved Gate Recurrent Unit Neural Network. *IEEE Transactions on Intelligent Transportation Systems,* pp. 16654-16663.

Wang, Z. & Thulasiraman, P., 2019. Foreseeing Congestion using LSTM on Urban Traffic Flow Clusters. *6th International Conference on Systems and Informatics (ICSAI), Shanghai, China.*

Xue , C., Eastin, K., Zhang, J. & Romo, L., 2020. *Short-Term Traffic Speed Prediction via Machine Learning.* s.l., s.n.

Yang, H.-F., Dillon, T. S. & Chen, Y.-P. P., 2017. Optimized Structure of the Traffic Flow Forecasting Model With a Deep Learning Approach. *IEEE Transactions on Neural Networks and Learning Systems.*

Yu, L., Zhao, J., Gao, Y. & Lin, W., 2019. Short-Term Traffic Flow Prediction Based On Deep Learning Network. *International Conference on Robots & Intelligent System (ICRIS), Haikou, China,* pp. 466-468.

Zhang, J. et al., 2011. Data-Driven Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems,* pp. 1624-1635.

Zhao, J. et al., 2019. Truck Traffic Speed Prediction Under Non-Recurrent Congestion: Based on Optimized Deep Learning Algorithms and GPS Data. *IEEE Access.*

Zhao, W. et al., 2019. Deep Temporal Convolutional Networks for Short-Term Traffic Flow Forecasting. *in IEEE Access, vol. 7.*

Zhene, Z. et al., 2018. Deep Convolutional Mesh RNN for Urban Traffic Passenger Flows Prediction. *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation,* pp. 1305-1307.