National College of Ireland

Big Data-driven Performance Improvement of Traffic Flow Prediction and Speed Limit Classification using Deep Learning

MSc Research Project
Data Analytics

Sankara Subramanian Venkatraman
Student ID: x18179541

School of Computing
National College of Ireland

Supervisor: Mr.Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Sankara Subramanian Venkatraman
Student ID:	x18179541
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Mr.Hicham Rifai
Submission Due Date:	17/08/2020
Project Title:	Big Data-driven Performance Improvement of Traffic Flow Prediction and Speed Limit Classification using Deep Learning
Word Count:	5,618
Page Count:	23

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:	
Date:	26th September 2020

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):	

Big Data-driven Performance Improvement of Traffic Flow Prediction and Speed Limit Classification using Deep Learning

Sankara Subramanian Venkatraman
x18179541

Abstract

Traffic flow and vehicle speed limit are the common problems faced in day-to-day life by every country. Improving the performance of traffic flow and speed limit benefits the road users and transportation authorities. In urban traffic network, it is challenging to predict the traffic parameters (flow, speed and occupancy) due to their complex nature. Additionally, various non-traffic parameters such as weather, light and road surface conditions influence traffic parameters. Several studies in the past have taken these parameters with lesser or aggregated data. This research considers the non-traffic parameters and traffic big data of the United Kingdom between the years 2010 and 2018. The significance of using non-traffic parameters along with traffic parameters during peak and non-peak hours is analysed. In the first part of this research, the traffic flow is predicted using different Long Short-Term Memory (LSTM) models, and the second part involves the classification of speed limit using Convolutional Neural Network (CNN) model. Finally, the models are validated using evaluation metrics of training and testing accuracy, RMSE (Root Mean Squared Error) value and confusion matrix. Traffic flow prediction and speed limit classification with non-traffic parameters perform better than traffic-only parameters with increased accuracy (1%) and lowered RMSE value (30% (traffic flow) and 33% (speed limit)).

1 Introduction

Studies on traffic flow and speed prediction are essential for the Intelligent Transportation System (ITS). With the increasing number of vehicles and population, it became troublesome for all the countries to manage the traffic and expand their cities. The modern traffic monitoring system uses sensors, inductive loops and video surveillance to collect traffic parameters (flow, speed and occupancy) of the vehicles. The data generated by this system are huge, non-linear and has Spatio-temporal features. Hence to lead a quality life, the research on traffic flow is important, and it supports government bodies to regulate transportation services.

1.1 Background and Importance

Most of the studies in the past have utilized small datasets or aggregated data (5, 10 and 15 minutes) interval to predict short-term traffic flow. A very few researches have focused

on predicting traffic flow with massive data but without considering non-traffic parameters. Research on traffic flow prediction has considered non-linear factors of breakdown and recovery of vehicles Arif et al. (2018), temporal factors of rainfall Jia et al. (2017) and weather Essien et al. (2019). This research bridges the gap by considering various non-traffic temporal parameters of weather, light and road surface conditions along with huge traffic dataset containing spatial characteristics.

The challenging objective of this research is to predict the traffic flow that has stochastic, non-linear and irregular nature. In most of the deep learning analysis, models directly choose the features from the data. But the seasonality, trends and patterns in the time-series data remain uncovered. So, to overcome this, the irregular and non-linear data are transformed into linear and stationary data using feature extraction. The feature extraction is carried out using various Exploratory Data Analysis (EDA) such as normality check, kurtosis, skewness and seasonality of the data. Dickey-Fuller test is conducted to check the stationary of the data using a hypothesis test. To process the huge volume and variety of data generated by ITS, Apache Spark is utilized. With the help of big data analytics and deep learning, the analysis can be performed with high processing speed and less computation time.

1.2 Research Question

Can non-traffic temporal parameters of weather, light and road surface conditions have a significant impact on traffic parameters (flow, speed) using deep learning models? Will they affect the traffic flow and speed limit during the peak and non-peak hours?

The objective of this research is to identify whether the non-traffic parameters helps in improving the performance of traffic flow prediction and speed classification. The insights from the research will benefit the United Kingdom transportation department. Since the research has 2 objectives, it is carried into 4 experiments. The first one is the prediction of traffic flow (regression), without (Experiment-1) and with (Experiment-2) non-traffic parameters. The second is to classify the speed limit (classification), without (Experiment-3) and with (Experiment-4) non-traffic parameters. Finally, the effect of non-traffic parameters on peak and non-peak hours is answered using statistical methods.

1.3 Research Objectives

The objectives are defined in Table 1 to achieve the above research question.

1.4 Limitation, Assumption and Structure of the Report

This research predicts traffic flow and classifies speed limit for weekdays only. Also, the spatial correlation of the adjacent roads was not considered in this analysis. In the classification of the speed limit, the data is not divided into a multi-class category due to class imbalance.

It is assumed that weather, light and road surface conditions remain the same within the local authority. Also, the morning peak hours is assumed from (7 to 10), evening peak hours from (16 to 18) and non-peak hours from (11 to 15).

The rest of the research is organized as follows. Section 2 critical analyze related works of parametric, non-parametric models and big data analytics in ITS. In Section 3, the methodology of the research project is explained in detail, the design specification of the research is demonstrated in Section 4. In Section 5, implementation of the models

are illustrated, evaluation and discussions in Section 6 and finally concluded in Section 7.

Table 1: Research Objectives

Objectives	**Description**	**Evaluation Metrics**
Objective 1	Data preparation of traffic flow and road safety dataset.	-
Objective 2	Data cleaning and transformation using Apache SparkSQL.	-
Objective 3	Feature extraction and data stationary test using normality, skewness, kurtosis and ADF tests.	**Normality** – if ($p < 0.05$) then reject the Null hypothesis and data is not normal. **Kurtosis** (-10 to +10) **Skewness** (-3 to +3) **ADF** - if ($p < 0.05$) then reject the Null hypothesis and data is stationary.
Objective 4	Implementation of deep learning models 1) LSTM (traffic flow prediction) 2) CNN (speed limit classification)	LSTM - Train, Test accuracy and RMSE CNN - Train, Test accuracy and Confusion Matrix
Objective 5	Visualize the graphs using matplotlib, seaborn and scikit-plot Python libraries.	-

2 Related Work

One of the most common problems addressed in the ITS is prediction of traffic parameters (flow and speed) of a road based on the peak or non-peak hours. Traffic flow can be predicted for short-term, medium-term and long-term. Due to the stochastic nature of traffic, it is challenging to predict the flow accurately. In this research, raw traffic counts of major and minor roads and road safety accidents data of the UK government are used. Non-parametric approaches are used to predict short-term traffic volume and classify speed limit. Generally, traffic volume and speed limit have been predicted and compared using strategies like parametric and non-parametric approaches. Simultaneously, ITS generates a huge volume of data and it difficult to obtain insights with local machine configuration. So, Big data analytics is leveraged with deep learning to overcome this issue. Sections 2.1 and 2.2 summarises the usage of parametric and non-parametric models in the application of ITS. In section 2.3, Big data analytics in ITS is discussed.

2.1 Parametric Models

Widely used parametric models are ARMA, ARIMA Alghamdi et al. (2019), ARIMA-SVM Chi and Shi (2019), SARIMA Luo et al. (2018) and partial least squares Gu and Zhou (2019). The most common assumption of time-series models are data is stationary and auto-correlation does not change to time. The stationary of the data is

tested using Augmented Dickey-Fuller (ADF) and KPSS tests. Parametric models use Auto-Regressive (AR), Integrated (I) and Moving Average (MA) parameters to obtain stationary traffic flow time-series data. Short-term traffic prediction have found hybrid models produces better accuracy than ARIMA and RBF-ANN models individually Li et al. (2017). Traffic flow prediction based on improved SARIMA and GA has prediction accuracy same as ANN Luo et al. (2018). In urban arterial roads, Artificial Neural Network (ANN) outperforms traditional ML models like KNN, SVM and SVR in the prediction of traffic flow Bartlett et al. (2019). The disadvantages of parametric approaches are shallow architecture and low accuracy due to spatial-temporal, non-linear, dynamic and chaotic characteristics of traffic flow. So, non-parametric or deep learning is preferred over parametric models in this research.

2.2 Non-Parametric Models

The main advantage of using neural network models are generalization, ability to handle non-linear and multi-variate data. Also, it has the advantage of continuous training using Back Propagation (BP) algorithm. Researchers have used various deep learning techniques like Deep Belief Network (DBN) for traffic flow prediction with multi-task learning Huang et al. (2014) and traffic speed prediction Jia et al. (2016). Lv, Duan, Kang, Li and Wang (2015) have used Stacked Auto Encoders (SAE) for traffic flow prediction Lv et al. (2015). The 5 different types of LSTM models widely used in the research are Vanilla-LSTM, Stacked-LSTM, Bidirectional-LSTM, CNN-LSTM and Conv-LSTM.

Liu, Zheng, Feng and Chen (2017) have used Conv-LSTM for short-term traffic prediction Liu et al. (2017). For urban traffic passenger prediction, Zhene et al. (2018) have used CNN-LSTM Zhene et al. (2018). Wang and Thulasiraman (2019) have used Vanilla-LSTM for forecasting traffic flow in clusters Wang and Thulasiraman (2019). Urban traffic speed prediction using deep learning by Essien, Petrounias and Sampio (2019) have used Bi-directional LSTM Essien et al. (2019). For the prediction of short-term traffic congestion, CNN has been used Chen et al. (2018). Some researchers have considered traffic flow with the effect of non-traffic factors such as weather, rainfall, temperature, accidents, peak and non-peak hours for predicting the traffic flow. Urban traffic passenger flow prediction by Zhene et al. (2018) has considered holiday and workday factors. Similarly, Zheng et al. (2019) have predicted traffic flow considering parameters of weather, weekday, weekend and holiday Zheng et al. (2019). Forecasting of traffic flow on urban area by Wang and Thulasiraman (2019) have considered peak and non-peak hours. The complexity of Spatio-temporal and periodicity characteristics in traffic data is handled by insensitive to time gap and cell state (short-term and long-term memory) nature of LSTM.

Traffic speed prediction with rainfall-integrated proved LSTM has better prediction accuracy than DBN considering additional parameter of rainfall Jia et al. (2017). Liu et al. (2017) concluded short-term memory characteristics of LSTM made it suitable for processing time-series data. Also, LSTM has a disadvantage of considering only temporal factors it is better to use CNN-LSTM or Bi-directional LSTM for extracting Spatio-temporal features. Essien et al. (2019) have predicted data fusion of traffic with rainfall and weather parameter provide better accuracy in urban traffic than traffic-only speed prediction. Urban traffic speed prediction using the CNN model is effective for recognizing Spatio-temporal patterns, non-linear and non-periodic characteristics. It has

observed that CNN models produce improvement in accuracy up to 23.8% compared to other models Ren and Yang (2018). So, LSTM and CNN models are widely preferred over other deep neural network models for time-series prediction.

Input, output and hidden are 3 layers of LSTM model. Based on the number of hidden layers and input sequence, the type of the LSTM model varies. High level of information is extracted by LSTM model for time-series data. The 3 types of prediction designs are One-to-One, Many-to-One and Many-to-Many predictions Wang and Thulasiraman (2019). In this research, the prediction model of Many-to-Many is adapted for predicting the traffic flow and speed limit classification of different roads. The traffic flow of Road (R_1) predicted at time n, time-series traffic flow data fed to the model from ($t = 1, 2, 3, \ldots, n-1$). In Conv-LSTM, the input time-series vector fed to LSTM model as $C_t = (C_1, C_2, C_3, \ldots C_{t-1})$ after convolutional and pooling process Liu et al. (2017). Similarly, Essien et al. (2019) have considered the input sequence of traffic time-series as $x_t = x_1 + x_2 + x_3 + \ldots + x_{t-1}$ and output sequence of $y_t = y_1 + y_2 + y_3 + \ldots + y_{t-1}$, where t represents the prediction time Essien et al. (2019). The input can be represented as a time-series matrix (X) to predict the output matrix-vector (Y) to preserve the Spatial-temporal characteristics of traffic speed. The input matrix X and output matrix-vector Y of traffic speed (S) at different roads Ren and Yang (2018) can be represented as

$$X = \begin{bmatrix} S_1(t-1) & S_1(t-2) & S_1(t-3) & S_1(t-n) \\ S_2(t-1) & S_2(t-2) & S_2(t-3) & S_2(t-n) \\ S_3(t-1) & S_3(t-2) & S_3(t-3) & S_3(t-n) \end{bmatrix} Y = \begin{bmatrix} (S_1(t) \\ S_2(t) \\ S_3(t) \end{bmatrix}$$

2.3 Big data analytics in ITS

Parallel and distributed computation of data can be achieved by big data analytics. Apache Hadoop with Apache Spark is the most commonly used framework to store and process (Extract, Transform and Loading) the massive amount of data generated in ITS Zhu et al. (2019). According to Guerreiro et al. (2016), Apache Spark and SparkSQL are the widely used tools for ETL in big data processing ITS. Apache Spark is efficient and faster compared to other traditional algorithms like Hadoop MapReduce. Also, it processes data in-memory Garate-Escamilla et al. (2019) and optimizes computational time. Apache Spark can be used for both batch and streaming pipeline in multiple-cluster environments to implement Machine Learning models Leite et al. (2018). The huge volume of data generated from various sensors on different roads is widely processed using Apache Spark.

In this research, short-term traffic flow is predicted using LSTM model and speed classification using the CNN model on various roads is discussed. Data preparation, cleaning and transformation are handled using Apache SparkSQL. The input spatial and temporal characteristics of traffic flow are fed into the LSTM model as a matrix to predict the output matrix-vector. Similarly, the same technique is followed by traffic speed classification using the CNN model. A novel technique of considering morning peak, evening peak and non-peak hours of all the roads in weekdays is studied in the research. Additionally, non-traffic parameters of weather, light and road conditions are integrated to study the influence of these factors.

3 Methodology

The raw traffic count of major and minor roads and road safety data are sourced from the UK government website. Traffic flow, speed and weather conditions data from various sensors on different roads are massive. It should be collected and processed accurately to enhance the ITS system. Finally, appropriate models are applied to obtain insights from the data. A novel methodology of Knowledge Unveiling in Big Data (KUBD) Lounes et al. (2018) is incorporated into this research to implement the above steps. KUBD is a combination of Knowledge Discovery in Databases (KDD) and Big data analytics. The below section explains modified KUBD for the application of ITS, traffic flow prediction and speed classification.

3.1 Traffic Flow Prediction and Speed Classification Methodology

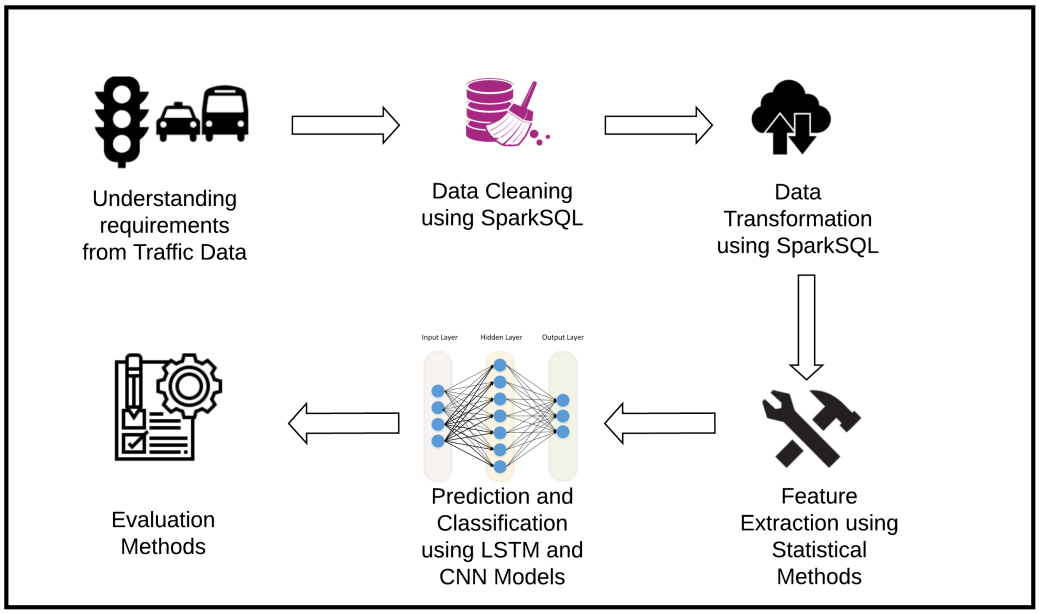


Figure 1: Traffic Flow Prediction Methodology

KUBD process has 6 important phases or stages in the proposed methodology and explained in Figure 1. Flexibility is the advantage of this methodology, as it can be modified according to the requirements. The modified KUBD starts with the first stage of understanding the requirements of data, storage types and appropriate models required for traffic flow prediction and speed classification. During the second stage of the process, data stored on the cloud is cleaned to remove noisy data, missing values, renaming column names according to standards and dropping unnecessary columns. After the cleaning phase, the data is transformed according to the requirements in the third stage. For example, the categorical string values are replaced with numerical values, joining of two tables and filtering records based on conditions are carried in this stage. After the data transformation, the data is stored in the PostgreSQL database for further analysis. Unlike regression or classification problems, features cannot be extracted based on the correlation analysis. In the fourth stage, a set of statistical analyses are carried out to check the

stationary, normality and seasonality in the time-series data, and the relevant features are extracted from the analysis. In the fifth stage, the multivariate time-series models of LSTM and CNN are applied to the dataset. Finally, appropriate metrics are used to evaluate model performance.

The upcoming sections explain the process of traffic data understanding, data preparation, cleaning, transformation and feature extraction in detail.

3.2 Traffic Data Understanding

In the previous studies on traffic flow and speed prediction in the United Kingdom, temporal factors of (rainfall and temperature) weather conditions and accidents with spatial characteristics of road type, direction and link length have considered. In this research, the traffic prediction generalization ability and the performance of traffic prediction are improved for the UK, considering the temporal factors of weather, light and road surface conditions. Also, it helps to regulate public transportation on weekdays during the morning peak, evening peak and non-peak hours.

3.3 Data Acquisition

Datasets for the analysis are ethically sourced from the websites [1,2]. The files are downloaded in Comma Separated Values (CSV) format from the UK government website. The open data are available for research scholars and students for conducting experiments and contribute to the improvements. The raw traffic count of major and minor roads in the United Kingdom have data of 3.91 million from 2000. As tabulated in Table 2, data from the year 2010 to 2018 which has 1.61 million records are considered. The road safety dataset contains weather, light and road surface conditions have data of 1.26 million records from the year 2010 to 2018.

Table 2: Dataset Description

Dataset	Record Count	Features
Raw Traffic Count	16,16,712	35
Road Safety Dataset	12,65,735	32

3.4 Data Preparation, Cleaning and Transformation

The crucial stage of the research is data pre-processing and cleaning, where missing and null value records are dropped. Merging of two datasets, transformation and storing data in a database are carried in data transformation. The feature engineering process is carried out using various statistical analysis in feature extraction. In experiment 1, only traffic flow data is utilized, and in experiments 2, 3 and 4 both traffic flow and road safety data are utilized.

3.4.1 Data Cleaning and Transformation for Experiment 1

Since the research is about traffic flow prediction on a major and minor road, geographical attributes cannot be null. After dropping null value records and filtering weekday

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

[2] https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data

records, the traffic dataset is reduced to 9,99,720. Data transformation is processed using SparkSQL by invoking SparkSession. Initially, the data is read from Google Cloud Storage (GCS) directly using SparkSession and stored as temporary tables. According to business logic, a new column "month" is created from count_date column and traffic volume of different vehicles are aggregated grouped on spatial parameters of travel direction, road type, road category, latitude, longitude, region, link length and local authority id. In the next step, categorical string variables of the month, road type, road category, travel direction are transformed to numerical values. Hour column is clustered into morning peak hours (7 to 10), evening peak hours (16 to 18) and non-peak hours (11 to 15) to a new column "peak_non_peak_hour". Finally, the cleaned transformed data is loaded into a PostgreSQL table in Google Cloud Platform (GCP) which has 12 columns and 2,45,730 records.

3.4.2 Data Cleaning and Transformation for Experiment 2

Data transformation, aggregation and clustering peak and non-peak hours are repeated for experiment 2, traffic flow prediction with weather, light and road surface conditions. As the temporal characteristics remain the same within the locality, the road safety data containing features of weather, light and road conditions are joined with traffic flow data. It is joined on the conditions of date, hour and local authority. The temporal features are categorical values, and the file available in the road safety manual is shown in Table 3. While joining two tables, data missing or out of range values (-1) are filtered from the analysis, only valid conditions from 1 to 9 are considered. The transformed data is loaded into a PostgreSQL table which has 15 columns and 1,53,858 records.

Table 3: Non-traffic temporal features

Code	Light condition	Weather Condition	Road Surface Condition
1	Daylight	Fine no high winds	Dry
2	-	Raining no high winds	Wet or damp
3	-	Snowing no high winds	Snow
4	Darkness - lights lit	Fine + high winds	Frost or ice
5	Darkness - lights unlit	Raining + high winds	Flood over 3cm. deep
6	Darkness - no lighting	Snowing + high winds	Oil or diesel
7	Darkness - lighting unknown	Fog or mist	Mud
8	-	Other	-
9	-	Unknown	-
-1	Data missing	Data missing	Data missing

3.4.3 Data Cleaning and Transformation for Experiment 3

Like experiment 1 data transformation, aggregation and clustering peak and non-peak hours are repeated for experiment 3, traffic speed limit classification without non-traffic parameters. Traffic safety dataset containing speed limit data is joined with traffic flow data on the conditions of date, hour and local authority id. Finally, data is stored into PostgreSQL which has 1,79,366 records and 12 columns.

3.4.4 Data Cleaning and Transformation for Experiment 4

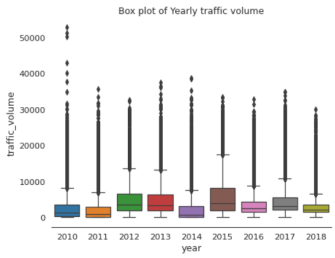
The data cleaning and transformation process of traffic speed limit classification with non-traffic temporal characteristics of weather, light and road conditions are like experiment 2. Also, records are filtered with a speed limit of less than 0. The output data containing 1,94,881 records and 15 columns are stored in the PostgreSQL table.

3.5 Feature Extraction

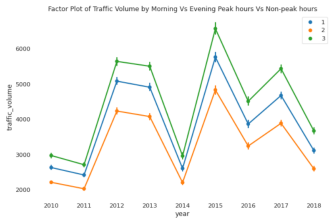
After data stored into PostgreSQL, features required to perform data mining is engineered based on a statistical analysis of normality test, skewness, kurtosis, traffic flow trends over the years, traffic patterns on the morning, evening and non-peak hours and Augmented Dickey-Fuller (ADF) test.

3.5.1 Statistical Analysis for Experiment 1

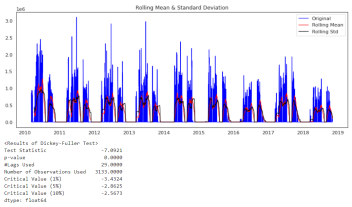
Data is read from PostgreSQL through sqlalchemy create engine and stored in pandas dataframe. The null (H_0) and alternate hypothesis (H_1) for normality is defined based on alpha value = 0.05. The normality test is conducted using stats module from Scipy. The null hypothesis is p-value > 0.05 (alpha value), then data looks like gaussian or normally distributed (fail to reject H_0). If p-value < 0.05, then data does not look gaussian (reject H_0). The obtained p-value from traffic flow dataset is 0.000, so (reject H_0) and data is not normally distributed. Skewness and kurtosis define the shape of the distribution. kurtosis of 5.91 and skewness of 2.19 defines the data is skewed on the positive side of the distribution.



(a) Traffic Volume Distribution 2010-2018



(b) Peak Vs Non-peak hours in Traffic Flow



(c) Data Stationary Test using ADF

Figure 2: Statistical analysis Experiment 1

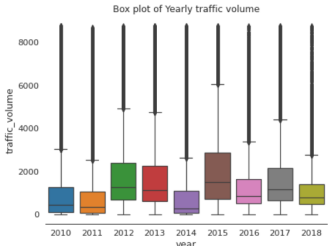
Traffic volume over the years from 2011 to 2018 almost remains the same except for 2010 shown in Figure 2a. Seasonality of the traffic flow over the years have also determined and found an interesting pattern of the year 2012 and 2013 have similar seasonality. The main objective of this research is to determine the traffic flow during morning, evening and non-peak hours.

As shown in Figure 2b, traffic volume in a non-peak hour (green) tops followed by morning peak hours (blue) and finally evening peak hours (orange). To check the stationary of the data, Dickey-Fuller test is conducted. The null hypothesis (H_0) p>0.05, then the data is non-stationary and has a time dependent component else alternate hypothesis (H_1) data is stationary. In this case, p-value = 0.000 < 0.05 (Figure 2c), so null hypothesis is rejected. Hence the data is stationary. Finally, the temporal factor of traffic volume feature is pivoted based on the "year" column to pass the data into the

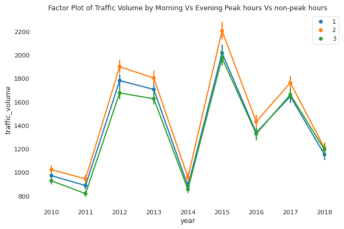
LSTM model. The actual data of 9,99,720 with 12 columns are transformed into 2,45,730 records and 13 columns.

3.5.2 Statistical Analysis for Experiment 2

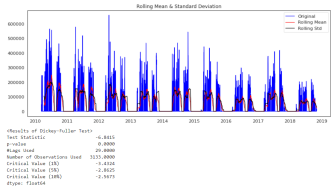
In addition to traffic volume, non-traffic temporal parameters of weather, light and road conditions are considered for experiment 2. Like experiment 1 normality, skewness and kurtosis tests were conducted. The normality test null hypothesis (H_0) is rejected as p-value=0.000, so the data is not normal. Initially, the skewness and kurtosis value of 4.92 and 39.4 were obtained. The high value of kurtosis indicates the data has outliers.



(a) Traffic Volume Distribution with non-traffic parameters 2010-2018



(b) Peak Vs Non-peak hours with non-traffic parameters



(c) Data Stationary Test with non-traffic parameters using ADF

Figure 3: Statistical analysis Experiment 2

So, after removing the outliers, skewness and kurtosis value of 2.0 and 4.7 were obtained. It defines the data is skewed on the positive side of the distribution. From Figure 3a, traffic volume over the years from 2010 to 2018 almost remains the same. Seasonality of the traffic flow over the years has determined and found that traffic flow with weather, light and road conditions have no similar seasonality pattern. As shown in Figure 3b, traffic volume in evening peak hour (orange) tops followed by morning peak hours (blue) and finally non-peak hours (green). From this, we can derive an insight that weather, road and light conditions affect morning and evening peak hours.

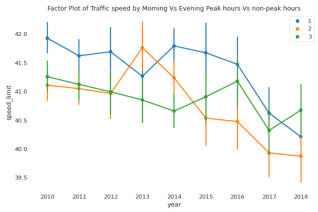
To check the stationary of the data, Dickey-Fuller test is conducted. In this case, p-value = 0.000 $<$ 0.05 (Figure 3c), so null hypothesis is rejected. Hence the data is stationary. Finally, the temporal factor of traffic volume, weather, light and road conditions features are pivoted based on the "year" column to pass the data into the LSTM model. The actual data of 1,53,858 with 15 columns are transformed into 1,16,561 records and 40 columns.

3.5.3 Statistical Analysis for Experiment 3

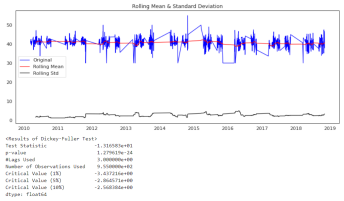
Traffic speed classification without non-traffic parameters data is read from PostgreSQL. The tests of normality, skewness, kurtosis, seasonality, speed limit distribution and data stationary tests are conducted. The data is not normal as p-value=0.000 $<$ 0.05 (reject H_0). Skewness and kurtosis of 0.77 and -0.92 are obtained for the traffic speed limit. The speed limit of the UK roads varies from 20 to 70 depending upon the time. Figure 4a represents the traffic speed limit distribution from 2010-2018.

The average speed limit has a similar pattern from 2010 to 2014. From 2015 the seasonality of traffic speed has a varied pattern. Unlike in experiment 1 and 2, there are no similar patterns on traffic speed limit based on the morning (blue), evening (orange)

(a) Speed limit 2010-2018



(b) Peak Vs Non-Peak in Traffic Speed



(c) Data Stationary Test in Traffic Speed Limit

Figure 4: Statistical analysis Experiment 3

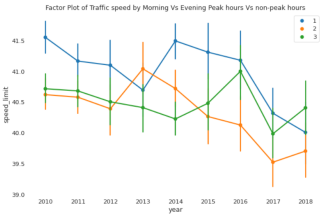
peak hours and non-peak (green) hours. Figure 4b shows how the average speed limit value changes over the years based on peak and non-peak hours.

Dickey-Fuller test is conducted to check the stationary of the data. In this case, p-value = $1.279619e^{-24} < 0.05$ (Figure 4c), so the null hypothesis is rejected. Hence the data is stationary. For speed limit classification, the data is classified into 2 categories Low Speed (1) and High Speed (2). The speed limit from 0 to 20 as Low speed and 30 to 70 is classified as High Speed. Because of the data imbalance, the data is not divided into a multi-class category. Finally, the temporal factor of the traffic speed limit feature is pivoted based on the "year" column to pass the data into the CNN model. Data obtained from the PostgreSQL with 1,79,366 records and 12 columns are transformed into 1,26,200 and 13 columns.

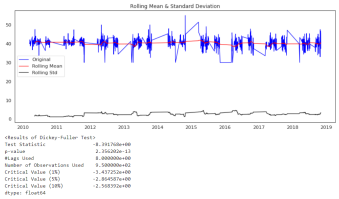
3.5.4 Statistical Analysis for Experiment 4

Traffic speed classification with non-traffic parameters data is read from PostgreSQL. The tests of normality, skewness, kurtosis, seasonality, speed limit distribution and data stationary tests are conducted. The data is not normal as p-value=0.000 < 0.05 (reject H_0). Skewness and kurtosis of 0.83 and -0.83 are obtained for the traffic speed limit. Figure 5a represents the traffic speed limit distribution from 2010-2018.

(a) Speed limit with non-traffic parameters 2010-2018



(b) Peak Vs Non-Peak with non-traffic parameters



(c) Data Stationary Test with non-traffic parameters

Figure 5: Statistical analysis Experiment 4

Research Objectives

The average speed limit has a similar pattern from 2010 to 2014. From 2015 the seasonality of traffic speed has a varied pattern. Figure 5b shows how the average speed limit value changes over the years based on peak and non-peak hours.

To check the stationary of the data, Dickey-Fuller test is conducted. In this case, p-value=$2.356202e^{-13}$ < 0.05 (Figure 5c), so the null hypothesis is rejected. Hence the data is stationary. Like experiment 3, the speed limit from 0 to 20 as Low speed and 30 to 70 is classified as High Speed. Finally, the temporal factor of the traffic speed limit, weather, light and road conditions features are pivoted based on the "year" column to pass the data into the CNN model. Data obtained from the PostgreSQL with 1,94,881 records and 15 columns are transformed into 1,25,999 and 40 columns.

4 Design Specification

As presented in Figure 6, the proposed design architecture has 3 layers of data storage layer, big data analytics layer and data visualization layer. This architecture can be modified at any point according to the requirements of the data. In the data storage layer, raw input CSV files of traffic data are moved from the local system to Google Cloud Storage (GCS) bucket. The main advantage of this process is to reduce the dependencies of storage, memory and machine failure in the local system. The big data analytics layer is integrated with the storage and visualization layer. This layer is provisioned on Google Cloud Platform (GCP) and software required for the analysis are installed. The data from GCS is cleaned and business logic are implemented using Apache SparkSQL and stored in a relational database of PostgreSQL server in GCP. So, the final transformed data required for performing the analysis is stored on PostgreSQL. In the analytics layer, statistical tests of normality, skewness, kurtosis, trends, seasonality and Dickey-Fuller (ADF) are conducted using scipy and statsmodel and sklearn. The objective of this research is to predict traffic flow and speed limit classification in the United Kingdom.

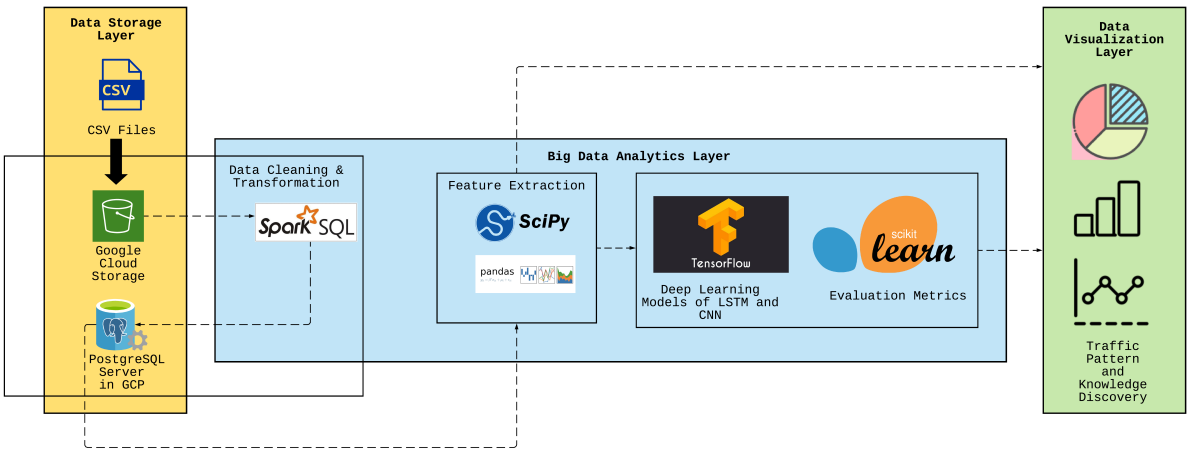


Figure 6: Design Specification for Traffic Flow Prediction

Using Keras and TensorFlow packages, deep learning models of LSTM and CNN are implemented. Scikit-learn package is used for implementing the evaluation. Data visualization layer is utilized to visualize the trends, normality and patterns of traffic flow characteristics. Also, this gives a better understanding of the results obtained from

the models. The design architecture is implemented end-to-end in GCP using various services such as GCS, PostgreSQL server and AI Notebook. Both GCS and PostgreSQL server are used for storing raw and transformed data. The AI notebook is used for ETL process, feature extraction, model deployment, hyper-parameter tuning and evaluates the model performance. The upcoming section explains the implementation of the models, performance tuning and evaluation methods performed in the research.

5 Implementation of Traffic flow Prediction and Speed Classification

As discussed earlier, the implementation of this research is carried in 2 parts and 4 experiments. The first part of this research is traffic flow prediction for the year 2018 based on traffic flow (volume) from the year 2010 to 2017, without non-traffic parameters (Experiment 1) and with non-traffic parameters of weather, light and road conditions (Experiment 2). The second part of this research is to classify the speed limit, Low Speed (1) and High Speed (2) for the year 2018 based on speed limit from the year 2010 to 2017 without non-traffic parameters (Experiment 3) and with non-traffic parameters (Experiment 4). Experiment 1 and Experiment 2 are carried out using various LSTM models such as vanilla-LSTM, stacked-LSTM and Bi-directional LSTM. Experiment 3 and Experiment 4 are conducted using 1-D CNN model.

5.1 Data Preparation for the Models

Even though data are extracted and pivoted by statistical tests and pandas, a set of the pre-processing must be carried out to data into the deep learning models. Data preparation is common for all the experiments. The first step is to normalize the data from the pivoted dataframe using sklearn. MinMaxScaler normalization is used for normalizing the data. It is applicable only for Experiment-1 and Experiment-2 (Regression). The second step is to split the data into training and test dataset to train and test the model built. A part of testing data is used as validation during the model fit. The data is split into 90% for training and 10% for testing using train_test_split from sklearn. The 2-D data (samples, features) is reshaped into 3-D (sample, time steps (1), features) to feed the time-series data into the model.

5.2 Long-Short Term Memory (LSTM) for Traffic Flow Prediction (Experiment 1 and 2)

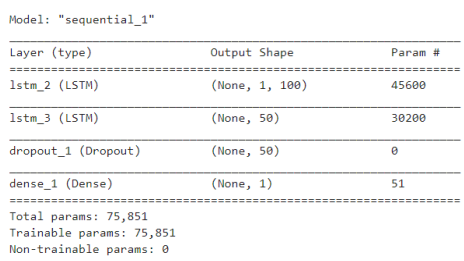
Deep Learning models are inspired from working of neurons in a brain. The architecture of Neural Network has 3 layers namely, input, hidden and output layer. The various LSTM Sequential models are implemented using Keras package in Python 3.

5.2.1 Experiment 1 and 2

3-D data (training data, 1, 13) is passed as input for experiment 1. The 13 features include spatial characteristics of travel direction, road category, road type, peak-non-peak hours, month and traffic volume time-series data from the year 2010 to 2017. The output of the model is to predict the traffic volume for the year 2018. The vanilla-LSTM model has 1

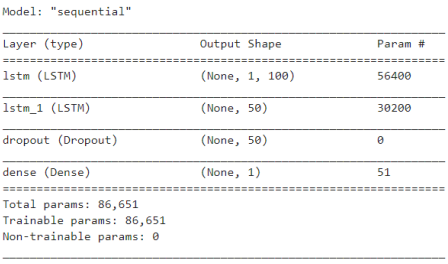
hidden layer with 100 neurons, and bidirectional and stacked LSTM models have 2 hidden layers. The first layer has 100 neurons while second layer has 50 neurons. The dropout layer, dense layer, activation layer, optimizer and loss function hyper-parameters are tuned for iterative testing. The objective is to achieve accurate predictions with minimal error. Figure 7a shows that the stacked-LSTM model has 75,581 trainable parameters with 100 and 50 neurons in the 1^{st} and 2^{nd} layers.

The optimizer used is Adam, the activation function is Rectified Linear Unit (ReLu) and the loss function is Mean Absolute Error (MAE). Finally, the model is fit for 50 training epochs with a batch size of 50. EarlyStopping function of the Tensorflow package is used to avoid overfitting of the training dataset. It stops the training epochs if there is no improvement in the loss of the validation dataset [3,4].



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_2 (LSTM)                (None, 1, 100)            45600
_________________________________________________________________
lstm_3 (LSTM)                (None, 50)                30200
_________________________________________________________________
dropout_1 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 51
=================================================================
Total params: 75,851
Trainable params: 75,851
Non-trainable params: 0
```

(a) Experiment-1 Model Summary



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 1, 100)            56400
_________________________________________________________________
lstm_1 (LSTM)                (None, 50)                30200
_________________________________________________________________
dropout (Dropout)            (None, 50)                0
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
Total params: 86,651
Trainable params: 86,651
Non-trainable params: 0
_________________________________________________________________
```

(b) Experiment-2 Model Summary

Figure 7: LSTM Model Summary Experiment 1 and 2

3-D data (training data, 1, 40) is passed as input for experiment 2. The 40 features include spatial characteristics of travel direction, road category, road type, peak non-peak hours and month. The rest of the columns include temporal features of weather, light and road surface condition data (2010 to 2018) and 8 years of traffic volume data from the year 2010 to 2017. The output of the model is to predict the traffic volume for the year 2018 on various roads. The stacked-LSTM model summary is shown in Figure 7b. It has 86,651 trainable parameters with 100 and 50 neurons in the hidden layer. Like experiment 1, 3 different LSTM models are applied to the traffic volume prediction with non-traffic parameters. The main goal of the 1^{st} and 2^{nd} experiments is to check if the non-traffic temporal features have a significant effect on traffic flow prediction.

5.3 Convolutional Neural Network (CNN) for Traffic Speed Limit Classification (Experiment 3 and 4)

The second part of the research work is to classify the speed limit of the various road in the UK. As suggested by previous research works, CNN model is best suitable for classification problems. Before the CNN model is applied to the dataset, the speed limit column is classified into Low Speed (1) and High Speed (2) based on the median speed limit. Then the process of train-test split and reshape is carried to feed the time-series data into 1-D CNN model.

[3] https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/

[4] https://towardsdatascience.com/time-series-analysis-visualization-forecasting-with-lstm-77a905180eba

5.3.1 Experiment 3 and 4

Using Keras.utils package, the speed limit classification column is converted into a categorical variable, which is like one-hot encoding. The input data is passed into the CNN model in the shape of (training data, 13, 1). Batch size of 50 and training epochs 50 are assigned in advance before feeding the data into the model. Figure 8a represents the CNN model summary for experiment 3. The 1-D CNN for speed limit classification has 1 convolution layer, 1 pooling layer and 3 hidden layers with (64, 32, 16 neurons) in the 1^{st}, 2^{nd} and 3^{rd} layers respectively. To batch the data to the pooling layer, BatchNormalization – a deep learning technique is used to reduce the training epochs is added [5]. The loss function used is categorical_crossentropy and the optimizer used is Adam.

Model: "sequential_8"

Layer (type)	Output Shape	Param #
conv1d_8 (Conv1D)	(None, 12, 128)	512
batch_normalization_8 (Batch	(None, 12, 128)	512
max_pooling1d_8 (MaxPooling1	(None, 6, 128)	0
flatten_8 (Flatten)	(None, 768)	0
dense_32 (Dense)	(None, 64)	49216
dropout_24 (Dropout)	(None, 64)	0
dense_33 (Dense)	(None, 32)	2080
dropout_25 (Dropout)	(None, 32)	0
dense_34 (Dense)	(None, 16)	528
dropout_26 (Dropout)	(None, 16)	0
dense_35 (Dense)	(None, 3)	51

Total params: 52,899
Trainable params: 52,643
Non-trainable params: 256

(a) Experiment-3 Model Summary

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 40, 128)	512
batch_normalization (BatchNo	(None, 40, 128)	512
max_pooling1d (MaxPooling1D)	(None, 20, 128)	0
flatten (Flatten)	(None, 2560)	0
dense (Dense)	(None, 64)	163904
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_2 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 3)	51

Total params: 167,587
Trainable params: 167,331
Non-trainable params: 256

(b) Experiment-4 Model Summary

Figure 8: CNN Model Summary Experiment 3 and 4

Like experiment 3, the data is passed into the model in the shape of (training data, 40, 1). Non-traffic temporal parameters of weather, light and road conditions along with speed limit are passed into the 1-D CNN model. Figure 8b shows the model summary for experiment 4. It has 1 convolutional layer, 1 pooling layer and 3 hidden layers with 64, 32 and 16 neurons in the 1^{st}, 2^{nd} and 3^{rd} layers respectively. The loss function used is categorical_crossentropy and the optimizer used is Adam.

6 Evaluation

Experiment 1 and Experiment 2 uses various LSTM are evaluated based on training and testing accuracy, and Root Mean Square Error (RMSE) value. Experiment 3 and Experiment 4 uses CNN model are evaluated using training and testing accuracy and confusion matrix. Also, training and validation loss against epochs are discussed.

6.1 Evaluation for Traffic Flow Prediction without Non-Traffic Parameters (Experiment-1)

Experiment 1 is conducted using vanilla-LSTM, stacked-LSTM and bi-directional LSTM models. In this experiment, the morning peak hour, evening peak hour, non-peak hours

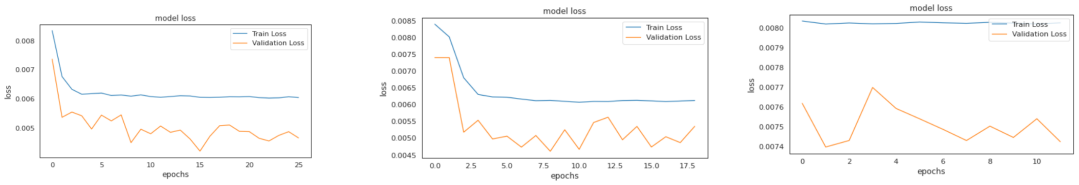
[5] https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/

and combined hours testing and training accuracy and RMSE values are tabulated in Table 4. It is found that the training and testing accuracy value remains unchanged for all types of LSTM models. In general, RMSE is the standard deviation of the residuals, it tells how well the predicted value is close to the actual value. An interesting pattern is found in RMSE, the value of vanilla-LSTM and Stacked-LSTM values are same for the morning, evening and non-peak hours. The Bi-directional LSTM has lesser RMSE values than the other 2 models. In the combined experiment, RMSE value, training and testing accuracy of all the models remain the same. EarlyStopping function used in the model fit has stopped the model from overfitting with right hyper-parameters at 19^{th} epoch for bi-directional LSTM, 12^{th} epoch for stacked-LSTM and 26^{th} epoch for Vanilla-LSTM.

Table 4: Experiment-1 LSTM Model Evaluation

Morning Peak			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9246	0.9241	0.193
Bidirectional	0.9246	0.9241	0.126
Stacked-LSTM	0.9246	0.9241	0.193
Evening Peak			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9247	0.9241	0.191
Bidirectional	0.9247	0.9241	0.136
Stacked-LSTM	0.9247	0.9241	0.191
Non-Peak			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9246	0.9241	0.194
Bidirectional	0.9246	0.9241	0.126
Stacked-LSTM	0.9246	0.9241	0.194
Combined			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9249	0.9236	0.166
Bidirectional	0.9249	0.9236	0.166
Stacked-LSTM	0.9249	0.9236	0.166

The evaluation of training and validation loss against the number of epochs of different models for combined hours is shown in Figure 9.



(a) Vanilla-LSTM (b) Bi-directional LSTM (c) Stacked-LSTM

Figure 9: Loss Vs Epochs in Experiment-1

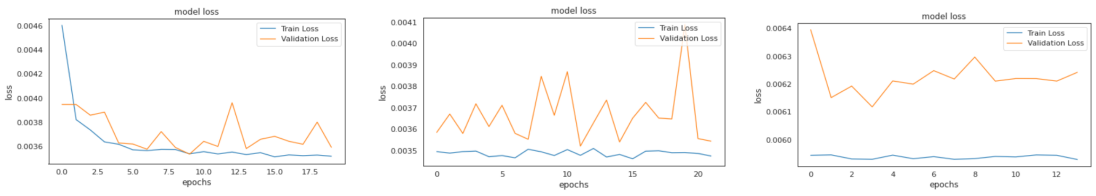
6.2 Evaluation for Traffic Flow Prediction with Non-Traffic Parameters (Experiment-2)

Experiment 2 is also the prediction of traffic flow with non-traffic parameters of road, light and road surface temporal factors. Table 5 shows the experimental results of the morning, evening, non-peak and combined hours. RMSE value of Bi-directional LSTM is less than other models for the morning, evening, non-peak and combined hour experiments.

Table 5: Experiment-2 LSTM Model Evaluation

Morning Peak			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9367	0.9266	0.126
Bidirectional	0.9367	0.9266	0.125
Stacked-LSTM	0.9367	0.9266	0.193
Evening Peak			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9331	0.9285	0.131
Bidirectional	0.9331	0.9285	0.127
Stacked-LSTM	0.9331	0.9285	0.185
Non-Peak			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9314	0.9312	0.103
Bidirectional	0.9314	0.9312	0.104
Stacked-LSTM	0.9314	0.9312	0.146
Combined			
Models/Metrics	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	0.9329	0.9329	0.083
Bidirectional	0.9329	0.9329	0.082
Stacked-LSTM	0.9329	0.9329	0.121

EarlyStopping function used in the model fit has stopped the model from overfitting with right hyper-parameters at 22^{nd} epoch for bi-directional LSTM, 14^{th} epoch for stacked-LSTM and 20^{th} epoch for Vanilla-LSTM. Figure 10 shows the evaluation of loss against the number of epochs for combined hours.



(a) Vanilla-LSTM (b) Bi-directional LSTM (c) Stacked-LSTM

Figure 10: Loss Vs Epochs in Experiment-2

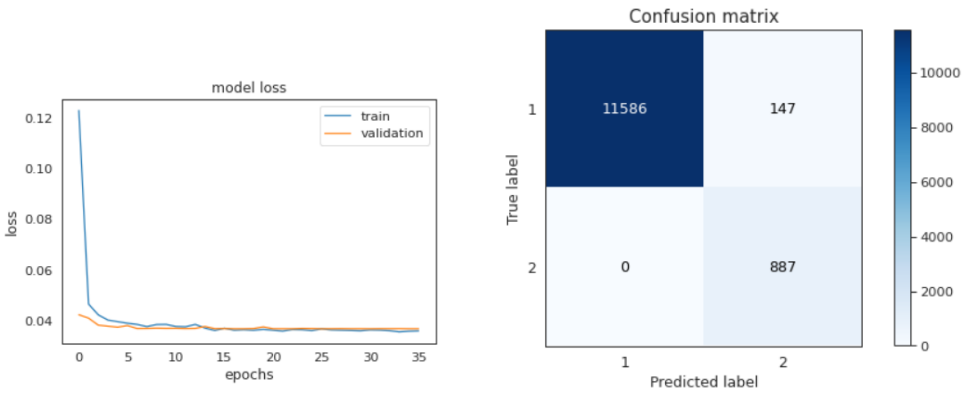
6.3 Evaluation for Traffic Flow Classification without Non-Traffic Parameters (Experiment-3)

Experiment 3 is conducted using the CNN model. The classification of Low speed (1) and High speed (2) is the objective of the model. In this experiment, the morning peak hour, evening peak hour, non-peak hours and combined hours of testing and training accuracy are tabulated in Table 6. EarlyStopping function used in the model fit has stopped the model from overfitting with right hyper-parameters at 36^{th} epoch. Testing and training accuracy values for the morning, evening and non-peak hours indicate the model is a good fit.

Table 6: Experiment-3 CNN Model Evaluation

Morning Peak		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9877	0.9880
Evening Peak		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9876	0.9869
Non-Peak		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9877	0.9869
Combined		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9867	0.9883

Loss value against the number of epochs is shown in Figure 11a. The confusion matrix is shown in Figure 11b shows how well the model predicts the Low Speed (1) and High Speed (2) against actual value. The model has predicted 147 records in the False Positive quarter.



(a) Loss Vs Epochs (b) Confusion Matrix

Figure 11: CNN Experiment-3 Evaluation

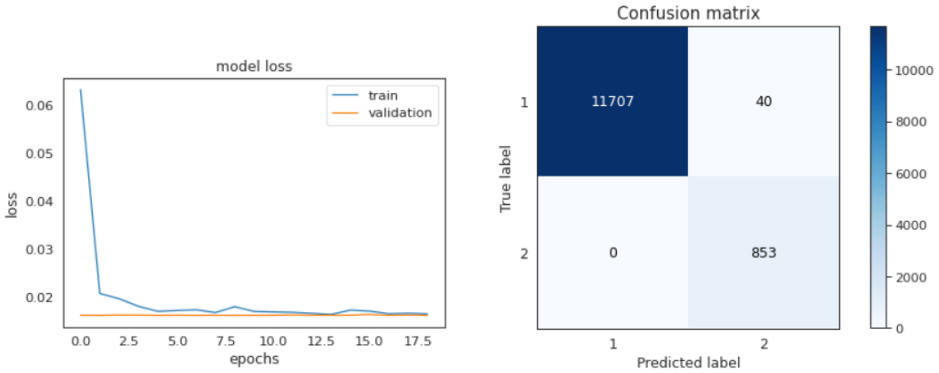
6.4 Evaluation for Traffic Flow Classification with Non-Traffic Parameters (Experiment-4)

Experiment 4 is conducted using the CNN model. The classification of Low speed (1) and High speed (2) have used non-traffic temporal features of light, weather and light conditions to the analysis. For a batch size of 50 and 50 training epochs, training and testing accuracy of the morning, evening, non-peak and combined hours values are tabulated in Table 7. EarlyStopping function used in the model fit has stopped the model from overfitting with right hyper-parameters at 19^{th} epoch based on validation loss value. The model is a good fit.

Table 7: Experiment-4 CNN Model Evaluation

Morning Peak		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9855	0.9959
Evening Peak		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9954	0.9983
Non-Peak		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9959	0.9947
Combined		
Models/Metrics	**Train Accuracy**	**Test Accuracy**
CNN	0.9956	0.9968

Figure 12a and Figure 12b shows the loss value against the number of training epochs and confusion matrix of classification with non-traffic parameters. The model has predicted only 40 records in the False Positive quarter.



(a) Loss Vs Epochs (b) Confusion Matrix

Figure 12: CNN Experiment-4 Evaluation

6.5 Discussion

Experiment 1 and Experiment 2 are analysed and compared for traffic flow prediction. Experiment 3 and Experiment 4 are compared for traffic speed limit classification with

and without non-traffic parameters. Table 8 compares the training and testing accuracy and RMSE value of traffic flow prediction with and without non-traffic parameters.

Table 8: Experiment-1 and 2 LSTM Model Comparison

Models/ Metrics	**Non-traffic parameters**	**Train Accuracy**	**Test Accuracy**	**Test RMSE**
Vanilla-LSTM	No	0.9249	0.9236	0.166
	Yes	0.9329	0.9329	0.083
Bi-directional	No	0.9249	0.9236	0.166
	Yes	0.9329	0.9329	0.082
Stacked-LSTM	No	0.9249	0.9236	0.166
	Yes	0.9329	0.9329	0.121

For traffic flow prediction with spatial and temporal characteristics for different roads, there is an increase in accuracy and decrease in RMSE value while comparing Experiment 1 and 2. It can be concluded that non-traffic parameters have a significant impact on traffic flow prediction.

Table 9: Experiment-3 and 4 CNN Model Comparison

Models/ Metrics	**Non-traffic parameters**	**Test Accuracy**	**TP**	**TN**	**FP**	**FN**	**Test RMSE**
CNN	No	0.9883	11586	887	147	0	0.082
	Yes	0.9968	11707	853	40	0	0.045

Also, while comparing Experiment 3 and 4, it is found that traffic speed limit classification with and without non-traffic parameters has a good fit and better accuracy. Table 9 gives an overview of experiment 3 and 4. Also, the traffic speed limit classification with non-traffic parameters has a significant effect with better test accuracy, more True-Positive and less False-Positive classification.

From the results, it is observed that the non-traffic parameters have a significant impact on traffic parameters (flow and speed limit). The trade-off between bias and variance is handled effectively. As suggested in the research Kang et al. (2018) the inclusion of spatial-temporal characteristics improves the prediction accuracy holds for this research. Traffic speed prediction using CNN for non-linear spatial-temporal characteristics Ren and Yang (2018) has obtained an RMSE value of 0.241, traffic speed prediction using Deep Belief Network (DBN) Jia et al. (2016) has obtained Normalized Root Mean Square Error (RMSN) of 0.07310 and improved traffic speed prediction Essien et al. (2019) using R-LSTM with rainfall and temperature parameter got an RMSE value of 0.0892, whereas in this research an RMSE value of 0.045 is achieved.

Research on traffic flow prediction Liu et al. (2017) using bi-directional LSTM improves the accuracy of the deep learning model is proved in this research. Short-term traffic prediction Ma et al. (2019) has proposed that LSTM model architecture can handle historical information for time-series compared to other models. Also, traffic flow prediction using stacked Auto Encoder (SAE) Lv et al. (2015) has obtained an accuracy of 93% without non-traffic parameters but in this research accuracy of 93.29% is obtained with non-traffic parameters using LSTM model. It is concluded that LSTM and CNN models are simple and effective for time-series data due to the nature of long-time dependencies and, handles non-linearity.

7 Conclusion and Future Work

In this research, big data traffic flow prediction and speed classification with and without temporal non-traffic parameters for the United Kingdom are studied. The main objective of performance improvement in traffic flow is primarily achieved by data pre-processing, transformation and exploratory data analysis. From the statistical data analysis, it has observed that traffic flow of morning and evening peak hours are increased by the inclusion of non-traffic parameters, but, there is no change in speed limit. It helped to visualize the patterns, trends and seasonality on the data. From the experiments 1 and 2 conducted using various LSTM models, it is observed that bi-directional LSTM has lesser RMSE (0.082) value. Also, in terms of accuracy, traffic parameters with non-traffic parameters perform better than without non-traffic parameters. The increase in accuracy of 1% indicates that the non-traffic temporal feature has a significant effect on traffic parameters. Similarly, experiments 3 and 4 on traffic speed classification using the CNN model displays traffic data with non-traffic parameters provides better accuracy and lesser RMSE value.

Also, the dataset obtained from the UK government website has non-linear components such as various counties, different road types, road categories, link length and vehicles travelling in different directions. So, the model built on top of the non-linear data has better generalization. But, this research is limited to traffic flow and speed prediction for weekdays. Also, the spatial correlation between the neighbouring roads has not considered. However, this research contributes to the Intelligent Transportation System (ITS) to plan public transportation services. It helps to control the traffic parameters (flow, speed) effectively during peak and non-peak hours.

In future work, the spatial correlation between adjacent roads can be considered along with non-traffic parameters. Also, the traffic parameters of the individual vehicle type can be clustered, and traffic flow and speed limit can be predicted based on vehicle type.

Acknowledgement

Firstly, I would like to show my sincere gratitude to the department of the School of Computing, National College of Ireland, Dublin, to fulfil my dream as a Data Analytics student. I would like to thank my research supervisor Mr. Hicham Rifai for his consistent support by guiding on technical expertise, report writing and formatting throughout the research work. Finally, I would like to thank my parents, data analytics professors and friends for continuous encouragement and inspiration.

References

Alghamdi, T., Elgazzar, K., Bayoumi, M., Sharaf, T. and Shah, S. (2019). Forecasting traffic congestion using ARIMA modeling, *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, IEEE, Tangier, pp. 1227–1232.

Arif, M., Wang, G. and Chen, S. (2018). Deep learning with non-parametric regression model for traffic flow prediction, *Proceedings - IEEE 16th International Conference on Dependable, Autonomic and Secure Computing*, IEEE, Athens, pp. 681–688.

Bartlett, Z., Han, L., Nguyen, T. T. and Johnson, P. (2019). A Machine Learning Based Approach for the Prediction of Road Traffic Flow on Urbanised Arterial Roads, *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, IEEE, Exeter, pp. 1285–1292.

Chen, M., Yu, X. and Liu, Y. (2018). PCNN: Deep Convolutional Networks for Short-Term Traffic Congestion Prediction, *IEEE Transactions on Intelligent Transportation Systems* **19**(11): 3550–3559.

Chi, Z. and Shi, L. (2019). Short-Term Traffic Flow Forecasting Using ARIMA-SVM Algorithm and R, *Proceedings - 2018 5th International Conference on Information Science and Control Engineering, ICISCE 2018*, IEEE, Zhengzhou, pp. 517–522.

Essien, A., Petrounias, I., Sampaio, P. and Sampaio, S. (2019). Improving Urban Traffic Speed Prediction Using Data Source Fusion and Deep Learning, *2019 IEEE International Conference on Big Data and Smart Computing, BigComp 2019 - Proceedings*, IEEE, Kyoto.

Garate-Escamilla, A. K., Hassani, A. H. E. and Andres, E. (2019). Big data scalability based on spark machine learning libraries, *ACM International Conference Proceeding Series*, IEEE, Strasbourg, pp. 166–171.

Gu, Z. and Zhou, S. (2019). Short-Term Traffic Flow Prediction and Its Application Based on the Basis-Prediction Model and Local Weighted Partial Least Squares Method, *14th International Conference on Computer Science and Education, ICCSE 2019*, IEEE, Toronto, pp. 992–997.

Guerreiro, G., Figueiras, P., Silva, R., Costa, R. and Jardim-Goncalves, R. (2016). An architecture for big data processing on intelligent transportation systems. An application scenario on highway traffic flows, *2016 IEEE 8th International Conference on Intelligent Systems, IS 2016 - Proceedings*, IEEE, Caparica, pp. 65–71.

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.

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.

Jia, Y., Wu, J. and Du, Y. (2016). Traffic speed prediction using deep learning method, *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, IEEE, Rio de Janeiro, pp. 1217–1222.

Kang, D., Lv, Y. and Chen, Y.-y. (2018). Short-term traffic flow prediction with LSTM recurrent neural network, *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, IEEE, Yokohama, pp. 1–6.

Leite, A. F., Weigang, L., Fregnani, J. A. and De Oliveira, I. R. (2018). Big data management and processing in the context of the system wide information management, *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, IEEE, Yokohama, pp. 1–8.

Li, K.-L., Zhai, C.-J. and Xu, J.-M. (2017). Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN, *Proceedings - 2017 Chinese Automation Congress, CAC 2017*, IEEE, Jinan, pp. 2804–2807.

Liu, Y., Zheng, H., Feng, X. and Chen, Z. (2017). Short-term traffic flow prediction with Conv-LSTM, *2017 9^{th} International Conference on Wireless Communications and Signal Processing, WCSP 2017 - Proceedings*, IEEE, Nanjing, pp. 1–6.

Lounes, N., Oudghiri, H., Chalal, R. and Hidouci, W.-K. (2018). From KDD to KUBD: Big data characteristics within the KDD process steps, *Advances in Intelligent Systems and Computing*, Springer Verlag, Naples, pp. 931–937.

Luo, X., Niu, L. and Zhang, S. (2018). An Algorithm for Traffic Flow Prediction Based on Improved SARIMA and GA, *KSCE Journal of Civil Engineering* **10**(1): 4107–4115.

Lv, Y., Duan, Y., Kang, W., Li, Z. and Wang, F.-Y. (2015). Traffic Flow Prediction With Big Data: A Deep Learning Approach, *IEEE Transactions on Intelligent Transportation Systems* **16**(2): 865—873.

Ma, C., Wang, Y., Chang, X., Li, Y. and Zhu, H. (2019). *Green Intelligent Transportation Systems*, Lecture Notes in Electrical Engineering, Springer Singapore.

Ren, S. and Yang, B. (2018). Traffic speed prediction with convolutional neural network adapted for non-linear spatio-temporal dynamics, *Proceedings of the 7^{th} ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial 2018*, Association for Computing Machinery, Inc, Seattle, pp. 32–41.

Wang, Z. and Thulasiraman, P. (2019). Foreseeing congestion using LSTM on urban traffic flow clusters, *2019 6^{th} International Conference on Systems and Informatics, ICSAI 2019*, IEEE, Shanghai, pp. 768–774.

Zhene, Z., Hao, P., Lin, L., Guixi, X., Du, B., Bhuiyan, Alam, M. Z., Long, Y. and Li, D. (2018). Deep convolutional mesh RNN for urban traffic passenger flows prediction, *Proceedings - 2018 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovations, SmartWorld/UIC/ATC/ScalCom/CBDCom/IoP/SCI 2018*, IEEE, Guangzhou, pp. 1305–1310.

Zheng, Z., Yang, Y., Liu, J., Dai, H. N. and Zhang, Y. (2019). Deep and Embedded Learning Approach for Traffic Flow Prediction in Urban Informatics, *IEEE Transactions on Intelligent Transportation Systems,* **20**(10): 3927–3939.

Zhu, L., Yu, F. R., Wang, Y., Ning, B. and Tang, T. (2019). Big Data Analytics in Intelligent Transportation Systems: A Survey, *Institute of Electrical and Electronics Engineers Inc,* **20**(1): 383–398.