National College of Ireland

A Machine Learning approach for Short-Term Traffic Flow Prediction

MSc Research Project
Data Analytics

Aaditya Ravindra Gajendragadkar
Student ID: 22158758

School of Computing
National College of Ireland

Supervisor: Furqan Rustam

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aaditya Ravindra Gajendragadkar
Student ID:	22158758
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Furqan Rustam
Submission Due Date:	14/12/2023
Project Title:	A Machine Learning approach for Short-Term Traffic Flow Prediction
Word Count:	9000
Page Count:	24

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:	Aaditya Ravindra Gajendragadkar
Date:	30th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

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

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

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

A Machine Learning approach for Short-Term Traffic Flow Prediction

Aaditya Ravindra Gajendragadkar
22158758

Abstract

Traffic congestion presents a complex problem, impacting not just the convenience of commuters, but also imposing significant economic and environmental consequences. While various models have been employed in traffic prediction, there remains a need for accurate, robust, and practical solutions to aid traffic managers in understanding patterns, reducing congestion, and optimizing traffic management strategies. This research focuses on predicting short-term traffic flow using statistical, machine learning, and deep learning models on the PEMS-08 Dataset from San Bernardino, covering July to August 2016 in five-minute intervals. In three case studies, deep learning models of LSTM, CNN, RNN, and machine learning models like KNN, Random Forest, Gradient Boosting, and Decision Tree demonstrate commendable performance, especially Random Forest, Decision Tree, and KNN, outshining others and making them first choice with R2 values of 0.99,0.98 and 0.97 respectively with extremely low RMSE and MAE values. Deep learning models LSTM, CNN, and RNN follow closely with R2 values of 0.958,0.957, and 0.91 respectively but with slightly higher RMSE and MAE, and statistical models of SARIMA and ARIMA Performed well with R2 of 0.95 and 0.90 but with an extremely high RMSE and MAE values. K cross-validation is performed on each machine learning and deep learning model that confirms the model's performance, robustness, and reliability. This research offers valuable insights to policymakers, presenting optimal models for developing proactive strategies. The findings contribute to fostering sustainable and efficient urban transportation systems by addressing dynamic traffic patterns.

1 Introduction

Today in the world of rapid urbanization every smart city faces a massive challenge that brings a city's infrastructure to a slowdown traffic congestion. The urban population continues to grow exponentially resulting in massive road traffic on roads and taking a toll on the city's roadways. The economic impact is very severe. As per the report from the World Bank, global traffic congestion costs huge losses worth $1 trillion annually, draining businesses through wasted time and inflated transportation costsNakat et al. (2014). Environmentally, the toll is equally worrisome. The EPA estimates that traffic accounts for 29% of nitrogen oxide emissions and 27% of volatile organic compounds in the United States aloneHockstad and Hanel (2018). These pollutants poison the air we breathe, fueling respiratory problems, acid rain, and a great threat to climate change Willetts et al. (n.d.). The adverse impact on health due to prolonged exposure to polluted

air and rising levels of stress among commuters underscores the urgency to solve the complex issue as explained by Gomes et al. (2023).

Hence in the context of sustainable urban development addressing traffic congestion and the need for innovative solutions becomes paramount. The research envisions contributing to curbing traffic congestion by predicting traffic flow at various locations offering a comprehensive approach by developing innovative traffic flow prediction models that offer statistical models, Machine learning models, and deep learning models. Through this approach, the research paves the way for a more responsive and efficient transportation system, thereby addressing the challenges posed by traffic congestion in contemporary urban environments. The overall benefits of Traffic flow prediction research include enabling real-time traffic management, optimized route planning, and efficient public transportation, thereby enhancing overall traffic efficiency, and reducing environmental impact. The above problems gives rise to our research question How effectively can a comparative evaluation of statistical, machine learning, and deep learning models for traffic flow prediction at specific locations using the PEMS-08 dataset improve traffic management systems and enhance traffic flow optimization across urban road networks?

While different Machine learning algorithms are explained by Sun et al. (2020) in the paper, "a vital consideration for the Internet of Vehicles (IoVs)". The research project aims to bridge this gap by conducting a comprehensive analysis of the efficiency and accuracy of various statistical, Machine learning, and Deep learning-based prediction models. The main objective of this project is to improve the accuracy of traffic prediction by incorporating statistical methods, machine learning algorithms, and deep learning techniques and offer the best-suited model. Additionally, the project aims to reduce computational time and provide the most optimal model for traffic prediction to enhance the efficiency of transportation networks in urban areas. The research aims to offer valuable insights that can drive the development of reliable and real-time traffic prediction systems, which have the potential to revolutionize modern urban transportation.

1.1 Research question

How effectively can a comparative evaluation of statistical, machine learning, and deep learning models for traffic flow prediction at specific locations using the PEMS-08 dataset improve traffic management systems and enhance traffic flow optimization across urban road networks?

1.2 Project Objective

The main objective of the research is to develop

- an advanced, effective accurate traffic prediction model to forecast traffic flow at a particular location for practical applications including optimizing traffic signals, contributing to more efficient traffic management.
- identify existing literature on different models present for traffic prediction.
- comparing and evaluating different Machine learning, deep learning, and statistical models.

This research contributes to more efficient traffic management, benefiting daily commuters, urban planners, transportation agencies, and emergency responders with timely and informed decision-making capabilities.

1.3 Structure of Report

The paper is structured into six sections. Section 1 serves as the introduction, providing an overview of the paper. Section 2 explores existing literature on traffic prediction and existing approaches. Section 4 outlines the implemented models by segregating them into three parts A, B, and C. An evaluation of the model is presented in Section 5. The paper concludes in Section 6, determining the most suitable model, and offers insights into potential future research directions. Figure 1 shows the overall benefits of Traffic flow prediction Sayed et al. (2023).

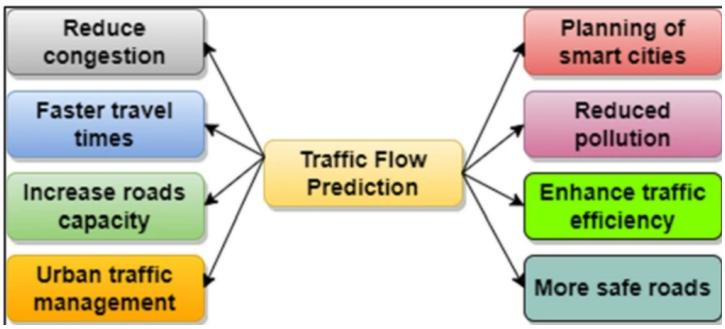

Figure 1: Benefits of Traffic flow Prediction

2 Related Work

Most of the traffic flow prediction models are classified based on Statistical models, Machine learning models, deep learning models, and hybrid models. In this section, we will dive deep into these traffic prediction models.

2.1 Traffic flow prediction using statistical models

In their researchKumar and Vanajakshi (2015) dealt with challenges in predicting traffic flow using a SARIMA model, showing its practicality on a Chennai Road. Their model achieved a 4–10% MAPE, outperforming historical averages and naive methods, making it useful for real-time short-term predictions. They also suggested future research areas, like exploring generalizability and hybrid approaches.

In another study a different approach was proposed by Tan et al. (2009) an aggregation model for traffic flow prediction. They combined MA, ES, ARIMA, and NN models and applied their data aggregation (DA) model to data from National Highway 107 in Guangzhou, China. The DA model, blending predictions from different time series, performed better than individual models, highlighting the benefits of using diverse modeling approaches. ARIMA model performed Fourth best with 12.5% MAPE while the proposed DA model performed best with 5.9% MAPE, while NP and NN performed with 9.5 and 9.7% MAPE, respectively. They also noted the impact of non-recurring events and recommended further research on applying the DA approach in scenarios with multiple detectors.

In the study Shekhar and Williams (2007) tackled the limitations of static traffic forecasting models, introducing a novel approach with the Kalman filter, recursive least

squares, and least mean squares for automated parameter estimation in the SARIMA model. Although the ML model performs slightly better with 7.19% as MAPE as compared to 7.31% of KF in real-time systems, KF performs much better also with a lower RMSE score the study suggests future research directions, including the application of adaptive techniques to other models and cross-dataset validation.

2.2 Traffic flow prediction using Machine Learning models.

In the Li and Xu (2021) research investigated improvements in video vehicle detection and proposed an SVR-based traffic flow prediction model. By evaluating current algorithms, they demonstrated that the SVR model had better accuracy, especially during busy peak hours, with lower Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE). The MAPE of SVR shows a reduction of 19.94% and 42.86%, while the RMSE demonstrates a decrease of 29.71% and 47.22%, respectively. respectively also introduced a new method for counting pedestrians using Histogram of Oriented Gradients (HOG) features. The study recommends refining calculations and pointing toward future possibilities, making use of the PeMS dataset.

In the study Meena et al. (2020) took a thorough approach to predict traffic flow by combining machine learning, genetic algorithms, and soft computing. They opted not to use deep learning due to limitations in data availability. Their algorithm, which utilized Decision Tree with an accuracy of 88%, Support Vector Machines with an accuracy of 88%, and Random Forest, achieved an impressive 91% accuracy. The study emphasizes the need for advanced methods to handle big data and improve intelligent transportation systems. In their conclusion, they highlight future avenues, such as integrating their approach with web servers. In their study on smart city traffic management.

In the studyMohammed and Kianfar (2018) used machine learning techniques for short-term traffic prediction. They applied Deep Neural Networks (DNN), Distributed Random Forest (DRF), Gradient Boosting Machines (GBM), and Generalized Linear Model (GLM) to Interstate 64 data. The results indicated similar performance among these models, and r2 values of all four model lies between 0.91 and 0.93 % with DRF slightly outperforming the others. Notably, the inclusion of upstream traffic data did not significantly improve accuracy, providing valuable insights for further exploration in various contexts.

2.3 Traffic flow Prediction using Deep Learning models.

In their pursuit of better traffic flow prediction, Shao and Soong (2016) proposed a Long Short-Term Memory (LSTM) model, demonstrating its effectiveness with a 5.4% Mean Absolute Percentage Error (MAPE) and with RMSE as 40.3 They highlighted the importance of smart transportation within a broader smart nation program. The study explores the optimization of hyperparameters, underscoring the efficiency, while the second best-performing model was SAE with MAPE of 6.7% and with RMSE of 47.3 of the LSTM models in achieving accurate predictions for traffic flow.

In the study Zheng and Huang (2020) explored the prediction of the traffic flow by comparing several types of models, including statistical, machine learning, and deep learning approaches. Analyzing data from Open ITS, they discovered that Long Short-Term Memory (LSTM) models performed better than the other methods. Out of the tested models of ARIMA and BPNN, LSTM with RMSE of 14.4438 and MAPE of 4.82% yielded

the most accurate traffic flow prediction. They emphasized the significance of employing advanced models, like LSTM, to effectively handle the unpredictable nature of urban congestion. Lv et al. (2014) explored by using deep learning for traffic flow prediction, employing Stacked Autoencoders (SAEs). They achieved an impressive accuracy rate of over 93% on the PeMS dataset. By addressing challenges related to big data, their model successfully captured intricate features, indicating the promising potential of deep learning in practical applications for predicting real-world traffic patterns.

In the study Yang et al. (2016) introduced the SAE-LM (Stacked Autoencoder Levenberg–Marquardt)model for traffic flow prediction, attaining a high accuracy rate of 90% when applied to M6 freeway data. The model proves effective in handling irregular traffic conditions, showing potential for reducing congestion. However, it is worth noting that the model has limitations when dealing with smoother traffic patterns.

In the study Yi et al. (2017) used TensorFlow™ Deep Neural Network (DNN) for predicting traffic flow, achieving an impressive 99% accuracy, especially in congested and non-congested conditions. This study is a pioneer in applying TensorFlow™ in the field of transportation engineering. It underlines the importance of refining the model and extending its application to broader datasets to enhance its effectiveness.

In another studyChen et al. (2018) introduced FDCN, a fuzzy deep-learning approach designed for predicting citywide traffic flow. FDCN recorded the least RMSE of 0.336. This method stands out for its effectiveness in handling uncertain and extensive datasets. By combining fuzzy theory with a deep residual network, the model surpasses existing methods, highlighting the promise of using fuzzy representation in traffic flow prediction.

In his study Polson and Sokolov (2017) presented a deep learning model designed for accurate short-term traffic flow predictions, surpassing the performance of sparse linear methods. R2 and MSE recorded were 0.79 and 9.14. While stating certain limitations such as concerns about interpretability and the absence of comparisons with more advanced neural network architectures, the study suggests that future research should explore alternative models suitable for diverse traffic conditions

2.4 Summary and limitation of work

In the review of existing literature, various methods for predicting traffic flow are explored. This includes statistical models, machine learning approaches, deep learning models. Some notable statistical models, like SARIMA and an adaptive approach using the Kalman filter, are effective in dealing with challenges related to the availability of data. They have proven to be practical for making real-time short-term predictionsKumar and Vanajakshi (2015)and Shekhar and Williams (2007) and Tan et al. (2009). Machine learning models, such as Support Vector Regression (SVR), and ensemble techniques like Decision Trees, Support Vector Machines, and Random Forests, have shown improved accuracy, especially during peak traffic hours. These models highlight the importance of using advanced methods to handle large datasets and contribute to intelligent transportation systems Li and Xu (2021) and Meena et al. (2020) and Mohammed and Kianfar (2018). In the world of deep learning for traffic prediction, certain models like LSTM and Stacked Autoencoders (SAEs) have proven to be effective. They do a great job of understanding complex patterns and handling a large amount of data, making them valuable for predicting traffic flow Shao and Soong (2016) Lv et al. (2014) Chen et al. (2018)Yang et al. (2016) Despite their effectiveness, some of these models face challenges in terms of understanding how they make predictions, using evaluation metrics that aren't consist-

ent, and not thoroughly comparing with more advanced neural network methods. Some models have difficulty in applying different models in different situations and at different traffic conditions. Hence in the future requirement, it is necessary to improve this model's interpretability. Using this method this can make the models more flexible and useful in a wide range of traffic scenarios.

By comparing and evaluating different models, our research provides a clear and very simple guide on the appropriate selection of traffic prediction models with ease of understanding evaluation metrics. The research focused on how well these methods could capture the complex patterns that develop over both time and space. By applying advanced techniques like machine learning and deep learning models with great accuracy and robust models the research becomes more practical, especially in managing and planning traffic. The research not only enhances our understanding of the PEMS-08 Dataset but also provides insights to improve current models, tackle concerns about how easily they can be interpreted, and guide future research in the field of traffic prediction.Table 1 shows summary of related work.

Table 1: summary of related work.

sr no	**Refrence**	**Technique**	**Data**	**score**
1	Kumar and Vanajakshi (2015)	Arima Sarima	Chennai roadway	4-10 Mape
2	Tan et al. (2009)	Arima,DA,NN,MA ES	PMS	5.9 MAPE
3	Shekhar and Williams (2007)	ML,sarima,kf	ITS	7.19 MAPE
4	Li and Xu (2021)	SVR SVM RF	Random	3.2 MAPE
5	Meena et al. (2020)	svr RF decision tree	ITS	0.99 R2
6	Mohammed and Kianfar (2018)	DNN DRF GBM GLM	Interstate 64	0.92 R2
7	Shao and Soong (2016)	LSTM	ITS	5.4 MAPE
8	Zheng and Huang (2020)	LSTM BPNN ARIMA	ITS	12.9 MAPE
9	Lv et al. (2014)	SAE BPNN SVM RBF	Random	34 MAE
10	Yang et al. (2016)	saelm psonn rbfnn	UK roadway	0.90 R2
11	Yi et al. (2017)	DNN	obd	R2 0.99
12	Chen et al. (2018)	cnn fdcn fcnn Arima	random	21.126 RMSE
13	Polson and Sokolov (2017)	LSTM RNN	chcihago highway	0.79 R2

3 Methodology

In this section, we delve into to structured framework for predicting traffic flow for the first five locations using PEMS-08. The four methodology phases involve PEMS-08 Dataset collection, Dataset preparation and Exploratory Data Analysis, Applying ML models with training and testing along with Hyperparameter Tuning, evaluating model performance with metrics such as R2 MSE and MAE. The research focuses on predicting traffic flow across 5 different locations in San Bernardino using the PEMS-08 Dataset from July to August 2016.The dataset records traffic data at 5-minute intervals with features like flow, occupy, and speed. Employing statistical, machine learning, and deep learning, models. The aim is to provide insights into urban planning and develop a suitable and optimum model for short-term traffic flow predictions. The goal is to make traffic management in urban areas more efficient and effective. Figure 2 shows methodology diagram.

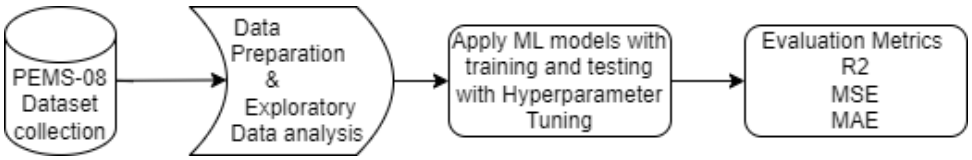

Figure 2: Methodology diagram

3.1 PEMS-08 Dataset Collection

The dataset contains the traffic data in San Bernardino from July to August 2016. There are 170 locations with detectors recording every 5-minute interval of traffic information. The dataset includes 3 features: flow, occupy, and speed. The details of the features are as follows: The flow variable in the PEMS08 dataset represents the number of vehicles that pass through the loop detector per time interval (5 minutes in this case). It is measured in vehicles per 5-minute interval. The occupancy variable represents the proportion of time during the time interval (5 minutes) that the detector was occupied by a vehicle. It is measured as a percentage. The speed variable represents the average speed of the vehicles passing through the loop detector during the time interval (5 minutes). It is measured in miles per hour (mph). For the experiment considering the huge size of data out of 170 locations, we have only predicted the first 5 locations of the dataset.

3.2 Data Preparation and Exploratory Data Analysis

In the data preparation stage, the jupyter notebook loads a traffic dataset, specifically the PEMS-08 dataset, which contains information on traffic flow, occupancy, and speed recorded at 170 different locations every 5 minutes over a specific period. The data is initially stored in CSV format, and the notebook converts it into a more familiar format, a pandas Data Frame, for ease of manipulation and analysis. The dataset is then analysed, resulting in a data frame with over three million rows and five columns: timestep, location, flow, occupy, and speed. The dataset is inspected for any missing values or anomalies, and fortunately,there are no null values. The distribution of the three main features (flow, occupy, and speed) is visualized to understand their spread. Subsequently, the notebook selects data from a random location (in this case, location 50) and explores the trends in occupancy, flow, and speed over the first 1000 timesteps. Data is then understood with the help of Summary Statistic: Descriptive statistics (mean, std, min, max) are computed and printed for numerical columns in the DataFrame. Visualization: Line plots illustrate traffic flow over time for the first five locations, while boxplots and violin plots depict the distribution of traffic flow at these locations exploration provides insights into the temporal patterns and behaviors of the chosen location, which can inform the subsequent steps in building predictive models for traffic-related tasks.

3.3 Applying ML model along with Hyperparameter Tuning

• In the applied phase of this study, determining the most effective model for predicting traffic flow in the PEMS-08 Dataset involves a comprehensive model selection. The primary prediction target is the traffic flow at the first five 5 locations. Overall, nine models are applied with hyperparameter tuning. Nine models include Arima, Sarima, KNN, Decision Tree, Random Forest, LSTM, CNN, and RNN.

• In ARIMA model: The code employs a grid search to optimize three ARIMA hyperparameters (p, d, q) for traffic flow prediction. It iterates over predefined ranges, training and testing ARIMA models for each combination. The best hyperparameters are chosen based on the highest R-squared score, and a final ARIMA model is trained on the entire dataset, making predictions on the test set.

• In SARIMA model: The code conducts SARIMA hyperparameter tuning through a grid search over p, d, and q values using itertools. product. It fits SARIMA models for each parameter combination, fixed at a seasonal order of (1, 0, 1, 12) for monthly data, with relaxed constraints. The best hyperparameters are chosen based on the highest R-squared score, and a final SARIMA model is trained on the entire dataset, making predictions on the test set.

• In KNN: The code utilizes K-Nearest Neighbors (KNN) regression with a hyperparameter, n neighbors, set to 5. This parameter controls the influence of neighboring data points on predictions and can be adjusted based on data characteristics. KNN is a non-parametric, supervised algorithm applied here for traffic flow prediction, leveraging time step and location features for each location.

• In Decision Tree: The code utilizes a Decision Tree regression model with a hyperparameter, max depth, set to 35. This parameter controls the depth of the tree, influencing model complexity. Decision Trees are non-linear, supervised algorithms employed here for traffic flow prediction, creating a tree structure where internal nodes make decisions based on features, and leaf nodes represent predictions.

• In Random Forest Regressor, this model, comprising 100 decision trees, excels in time series forecasting, combining 'timestep' and 'location' features for robust predictions. Feature Importances: Highlighting 'timestep' and 'location,' the model discerns feature importance is crucial for accurate predictions. Hyperparameter Setting: With n estimators fixed at 100, this vital parameter shapes the forest's size, impacting the model's complexity. In time series forecasting, Random Forest, an ensemble algorithm, merges predictions from diverse decision trees, ensuring resilient predictions for traffic flow.

• The RNN model is configured with 50 units in the SimpleRNN layer, a dropout rate of 0.2, trained for 100 epochs with a batch size of 32. Technique/Algorithm: The RNN is employed for time series forecasting, utilizing SimpleRNN for feature extraction and a Dense layer for prediction. It employs the Adam optimizer with mean squared error loss and incorporates early stopping to prevent overfitting. Sequences of length 10 are used for training and capturing temporal dependencies.

• The LSTM model is configured with 50 units in the LSTM layer and trained for 100 epochs with a batch size of 32. The model employs the Adam optimizer with mean squared error loss and incorporates early stopping to prevent overfitting's (Long Short-Term Memory) is used for time series forecasting. The model includes an LSTM layer for capturing long-term dependencies and a Dense layer for prediction. Sequences of length 10 are employed for training, and the model is evaluated using the R2 score.

• The CNN model has a convolutional layer with 64 filters, a kernel size of 3, ReLU activation, a max-pooling layer with a pool size of 2, a dense layer with 64 units and ReLU activation, a dropout layer with a rate of 0.3, and an output layer with 1 unit using a linear activation function. The model employs a 1D Convolutional Neural Network (CNN) for time series forecasting. Sequences of length 10 are used, reshaped to fit the input of the convolutional layer. The architecture includes convolutional layers with max pooling, a flattening layer, and fully connected dense layers. Training involves mean squared error

loss, Adam optimizer, and early stopping to prevent overfitting.

All models are fine-tuned and retrained to enhance their predictive performance, ensuring robust and accurate traffic flow predictions. The data is split into varying percentages of 90,80, 70, 60, and 50 used for training and 10,20, 30, 40, and 50 for testing for each model to check its robustness. After that K cross-validation is performed on machine learning and deep learning models to enhance model reliability, optimize hyperparameters, maximize dataset utility, detect overfitting or underfitting, assess model performance comprehensively, and reduce bias by iteratively splitting the data into K subsets for training and testing. Table 2 shows hyperparameter tuning for all models

Table 2: Hyperparameter tuning of all models

Model	**Hyperparameter Tuning**
ARIMA	Nested loop grid search;Values: p=2, d=1, q=3.
SARIMA	nested loop grid search with itertool product;
SARIMA	Values : PDQm = (1, 1, 2)(1, 1, 1, 12)
KNN	n_neighbors=5
Random Forest	n_estimators=100
Decision Tree	max_depth=35
Gradient Boost	n_estimators=100, learning rate=0.1
LSTM	units=50, sequence length=10, optimizer=adam, .
LSTM	loss=mean squared error, batch size=32
RNN	units=50, sequence length=10, dropout=0.2, optimizer='adam',
RNN	Loss= mean squared error, batch size=32
CNN	Conv 1D filters = 64, Conv 1D kernel size = 3, Conv 1D activation = relu
CNN	Max Pooling 1D pool size = 2, Dense1 units = 64, Dense1 activation = relu,
CNN	loss = mean squared error, epochs = 100, batch size = 16

3.4 Model Evaluation and Presentation

The models are evaluated based on insights from the Evaluation metrics, including R-squared (R2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) which are employed to provide a comprehensive assessment of the model performance.This helps to get a detailed picture, showing where each model shines and where it might struggle. The study wraps up by suggesting how this research could push traffic flow prediction methods forward and help with managing traffic in the real world. The results are displayed in a clear table and visualizations that dive into the details of how we got the best outcome.

4 Design Specification

In the initial stages of the project, the design specification is crucial for explaining the requirements, limitations, and objectives of a machine learning system. The chosen techniques and algorithms for the project are mentioned below:

4.1 Modelling Technique

• Random forest, an ensemble learning method for classification and regression, combines multiple decision trees to make predictions. It injects randomness into the training process by randomly selecting a subset of features and training examples to consider when splitting a node. This reduces overfitting and improves generalization performance. The prediction for a new instance is the average of the predictions from all trees. Breiman (2001). Mathematically, random forest can be represented as: y pred = 1/n estimators * sum(T i(x)) where y pred is the predicted value, n estimators is the number of trees, T i is the i-th decision tree, and x is the new instance.

• A decision tree regressor is a supervised machine learning algorithm used for regression tasks. It constructs a tree-like model that makes predictions by recursively splitting the data into subsets based on certain decision rules. Each split is determined by the feature that best reduces the variance or impurity of the data within the node. The resulting tree structure represents a set of rules that map input features to the predicted output. (The equation for predicting the output of a decision tree regressor is: y pred = leaf value where y pred is the predicted output and leaf value is the average or median of the target values for the data points belonging to the leaf node.

• K-nearest neighbors (KNN) regressor is a non-parametric machine learning algorithm used for regression tasks. It operates by identifying the k nearest neighbors in the training data to a new instance and predicting the average value of the target variable for those neighbors. The number of neighbors (k) is a hyperparameter that needs to be tuned. Altman (1992)The equation for predicting the output of a KNN regressor is: y pred = (1/k) * sum(y i)where y pred is the predicted output, k is the number of neighbors, y i is the target value of the i-th neighbor, and the sum is over the k nearest neighbors.

• Gradient boosting is an ensemble learning technique that builds an additive model in a series of stages. Each stage consists of training a weak learner on the residuals of the previous stage, where the residuals are the differences between the actual target values and the predictions of the previous stage. The predictions of the weak learners are then added to the predictions of the previous stages to obtain the final prediction. Friedman (2001)The algorithm continues to build stages until a stopping criterion is met, such as a maximum number of stages or a minimum error. y pred = F

$$\sum_{i=1}^{n} x_i$$

This is an equation for predicting the output of a gradient-boosting regressor.

• ARIMA (Autoregressive Integrated Moving Average) is a statistical method used to forecast time series data. It assumes that the time series can be modeled as a combination of autoregressive (AR) terms, which represent the dependence of the current value on past values, integrated (I) terms, which account for non-stationarity by differencing the data, and moving average (MA) terms, which represent the dependence of the current value on past forecast errors. Box and Jenkins (1976) The ARIMA model is represented by the equation: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$ where y_t is the value of the time series at time t, c is the constant term, ϕ_i are the autoregressive parameters, θ_i are the moving average parameters, ε_t is the white noise error term, p is the order of the AR model, d is the degree of differencing, and q is the order of the MA model.

• SARIMA (Seasonal Autoregressive Integrated Moving Average) is an extension of the ARIMA model specifically designed to forecast time series data with seasonal patterns. It incorporates seasonal parameters to capture recurring patterns in the data, making it particularly useful for forecasting time series with regular seasonal cycles, such as monthly or quarterly sales figures. The SARIMA equation is:

$$y_t = c + \phi_1 y_{t-s} + \phi_2 y_{t-2s} + \ldots + \phi_p y_{t-ps} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t.$$

Here, y_t is the time series value at time t, c is the constant term, ϕ_i are the seasonal autoregressive parameters, θ_i are the seasonal moving average parameters, ε_t is the white noise error term, s is the seasonal period, p is the order of the seasonal AR model, d is the degree of seasonal differencing, and q is the order of the seasonal MA model. The SARIMA model's parameters are estimated by minimizing the sum of squared errors (SSE) between the predicted y_t values and the actual y_t values.

• Recurrent Neural Networks (RNNs) are a type of artificial neural network (ANN) specifically designed to handle sequential data, such as text, speech, and time series data. Unlike traditional feedforward ANNs, which treat each input independently, RNNs incorporate a feedback loop that allows them to consider the context of previous inputs when processing new data. This makes RNNs well-suited for tasks that require an understanding of temporal dependencies, such as machine translation, natural language processing, and speech recognition. The core concept of an RNN is the use of hidden states, which represent the network's understanding of the input sequence at a given point in time. The hidden state is updated as the network processes each input, allowing it to capture the evolving context of the sequence learning. The equation of RNN is as follows:

$$h_t = \sigma(W_{hh} h_{t-1} + W_{xh} x_t + b_h).$$

• CNNs are a powerful class of artificial neural networks that have revolutionized the field of computer vision. At the heart of CNNs lie the convolutional layers, which are responsible for extracting features from images. These layers apply filters, also known as kernels, to the image pixels, sliding them across the image to generate feature maps. The extracted features are then pooled using pooling layers, which downsample the feature maps while preserving the most important features. This helps to reduce the dimensionality of the data and makes the network more efficient LeCun et al. (1998). The equation for CNN is as following y pred = softmax(z).y pred is the predicted output is the output of the final fully connected layer softmax is the softmax activation function, which normalizes the outputs of the fully connected layer to sum to 1, making them suitable for representing probabilities.

4.2 Evaluation Technique

The definitions for R-squared (R2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are as follows: R-squared (R2):

• R-squared is a statistical measure that represents the proportion of the variance in the dependent variable (traffic flow, in this case) that is predictable from the independent variables used in the model. It ranges from 0 to 1, where 1 indicates a perfect fit, meaning the model explains all the variability in the traffic flow data. Following is the equation for R2, R2 = 1 - (SSR/SST).where:SSR (Sum of Squared Residuals) and SST (Total Sum of Squares)

• Mean Absolute Error is a metric that calculates the average of the absolute differences between predicted and actual values. It represents the average magnitude of errors without considering their direction. MAE provides a clear measure of the model's accuracy in predicting traffic flow.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

n is the number of data points,
where:

n is the number of data points,

y_i is the actual value of the dependent variable for the$i$-th data point,

$\hat{y}_i$ is the predicted value of the dependent variable for the i-th data point.

• The root mean squared error (RMSE) is a popular metric for evaluating the performance of regression models. It measures the average magnitude of the errors in a set of predictions, where an error is the difference between the actual value and the predicted value. The lower the RMSE, the better the model is at predicting the target variable.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

where:

n is the number of data points,

y_i is the actual value of the dependent variable for the i-th data point,

$\hat{y}_i$ is the predicted value of the dependent variable for the i-th data point.

5 Implementation

5.1 Tools Utilized

The software tools involved from the initial stage through to the final prediction are listed below: 1. Language Python – Jupyter Notebook for code. 2. Microsoft Excel - Initial data analysis and filtering of data.

5.2 Data Exploration

It is necessary to understand the nature of the data, hence it becomes important to understand all variables in the dataset. To understand the data heatmap scatter plot, histogram, and summary statistics were analyzed to understand the nature of the data. The exploration starts with three scattered plots, explaining inter-variable relationships: 'Occupancy vs. Flow,' 'Speed vs. Flow,' and 'Speed vs. Occupancy.' These visualizations collectively offer us complete understanding of the traffic data's statistical properties and relationships, important for analytical and modeling endeavors. Below you can find in figure 3 the scatter plot, Scatter Plot 1: Occupancy vs. Flow.

The first scatterplot shows the relationship between occupancy and flow. Occupancy is a measure of how full the road is, while flow is a measure of the number of vehicles

passing a point on the road per minute. The scatterplot shows that there is a positive correlation between occupancy and flow, concerning location which means that as occupancy increases, flow also increases similarly speed and flow scatter plots can be analysed concerning location. The scatterplot shows that there is a negative correlation between speed and flow, which means that as flow increases, speed decreases. On the other hand, from Fig 4, we can get an idea about the distribution of flow speed and occupancy. The distribution of flow is skewed to the right, with a median of 2,200 vehicles per minute. The distribution of occupancy is also skewed to the right, with a median of 0.6. The distribution of speed is skewed to the left, with a median of 30 miles per hour. Along with this Heatmap is studied to understand the relationship between variables below is a summary from Fig 5. Flow and occupancy have a strong positive correlation. Speed and flow have a strong negative correlation. Location and the other three traffic metrics have moderate correlations.

From Figure 6 overall flow at the first five locations can be understood through a line plot. Location 0: Location 0 experiences the highest traffic flow during the morning rush hour, and the lowest traffic flow during the late-night hours. Location 1 experiences a moderate amount of traffic throughout the day, with slightly higher traffic flow during the morning and evening rush hours. Location 2 experiences the highest traffic flow during the evening rush hour, and the lowest traffic flow during the early morning hours. Location 3 experiences a consistent traffic flow throughout the day, with slightly higher traffic flow during the morning and evening rush hours. Location 4 experiences a moderate amount of traffic throughout the day, with slightly higher traffic flow during the midday hours.

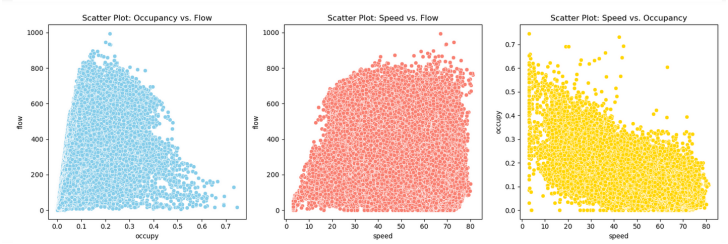

Figure 3: scatter plot Occupancy vs. Flow,' 'Speed vs. Flow,' and 'Speed vs. Occupancy

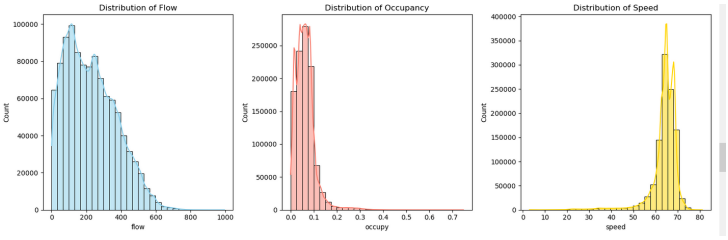

Figure 4: Distributionof flow speed occupancy

5.3 Implementation of Models

After careful exploration of data and understanding, implementation of the model takes place to determine which model suits best and is effective for traffic prediction. Overall,

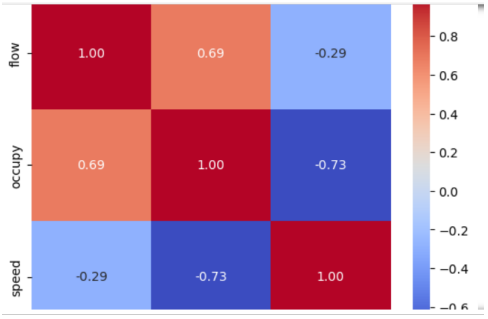

Figure 5: Heat map of flow speed occupancy

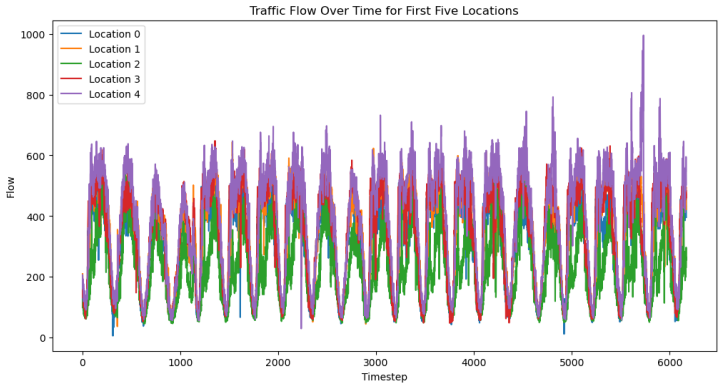

Figure 6: Traffic flow for five location

the traffic dataset PEMS-08 contains 170 locations but for the experiment, we have chosen the first five locations of the dataset to find the best-suited model for traffic prediction at a particular location. Models have been categorized into three groups, Model A, B, and C.

• Model A Utilizing Statistical Model

Model A consists of statistical models such as ARIMA and SARIMA. For ARIMA model performs a grid search for optimal ARIMA hyperparameters, and the ARIMA model is trained on the training data for each set of hyperparameters. The data is split with varying percentages of 80, 70, 60, and 50% used for training and 20, 30, 40, and 50% for testing. Traffic flow predictions are made at the first five locations using the test data, and the R-squared (R2), MSE, and MAE scores are calculated to evaluate the accuracy of the predictions. A similar experiment was performed utilizing the SARIMA model. The model demonstrates time series forecasting highlighting hyperparameter tuning, model fitting, and performance evaluation. It also includes error handling for potential model fitting issues. The data is split into varying percentages of 90,80, 70, 60, and 50% used for training and 10,20, 30, 40, and 50% for testing. Traffic flow predictions are made at the first five locations using the test data, and the R-squared (R2), MSE, and MAE scores are calculated to evaluate the accuracy of the predictions.

Model B utilizing ML Model.

• Four machine learning models were implemented Random Forest, Decision Tree, KNN, and gradient boosting. A similar approach was built on implementing these four models. The first five locations' traffic flow are predicted considering timestep and location as the independent variable and flow as the dependent variable. The hyperparameters are tuned to produce the best results. For KNN n neighbors are set to five, decision trees max depth is set to 35, gradient boosting and random forest n estimators are set to 100. Various train-test splits, including 80, 70, 60, and 50%, with corresponding test set sizes of 20, 30, 40, and 50% on the dataset. Traffic flow predictions are made at the first five locations using the test data and the model is evaluated using R-squared (R2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

• Model C utilizing DL Model.

Three Deep-learning models were implemented namely LSTM, simple RNN, and CNN. In our research model choices, and hyperparameter tuning, and compare performance using metrics, emphasizing insights from visualizations. Various train-test splits, including 90,80, 70, 60, and 50%, with corresponding test set sizes of 20, 30, 40, and 50% on the dataset. Traffic flow predictions are made at the first five locations using the test data and the model is evaluated using R-squared (R2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

To validate performance robustness and reliability K cross-validation is performed on ML and deep learning models where the parameter of k is set to 5 which divides the dataset into 5 folds.

6 Evaluation

Evaluation plays a key role in machine learning, containing crucial activities such as measuring model performance, selecting the optimal model, fine-tuning parameters, ensuring generalization to new data, interpreting results, quantifying business impact, guiding continuous improvement, and communicating insights to stakeholders. It serves as a crucial

point in the machine learning lifecycle, guiding decision-making, and ensuring models align with objectives.

6.1 Case Study -1 Evaluation of Model A

The ARIMA model produces strong predictive capabilities across locations, capturing a major variance in traffic flow data. Notably, location 0 stands out with the highest R^2 at 0.9582%, demonstrating its great fit, while location 2 also performs exceptionally well. Despite achieving consistent results across training and testing accuracies (ranging from 50% to 90%), variations in R-squared values and MSE are observed. Specifically, locations 1, 3, and 4 display moderate R-squared values with higher MSE compared to locations 0 and 2. ARIMA performs well in predictive accuracy, particularly in locations with higher R-squared and lower MSE, to enhance the need for better performance in specific locations.

SARIMA: The SARIMA model impressively predicts traffic flow variations across locations, boasting R-squared values from approximately 0.8869% to 0.9628%—indicating an excellent fit and exceptional pattern capture. Mean Squared Error (MSE) values are moderate to low, signifying solid accuracy in predictions. Notable performances include Location 3 with the highest R-squared (0.9628%) and lowest MSE (894.0896), and Location 1 showing an incredibly good fit (R-squared of 0.9497%) with low MSE (1183.4562). Locations 0 and 2 also demonstrate impressive performance. Overall, SARIMA emerges as a highly effective tool for predicting traffic flow, earning a positive evaluation.

Both ARIMA and SARIMA models demonstrate strong predictive capabilities, with SARIMA showing slightly better performance.

Table 2 shows the overall performance of ARIMA and SARIMA across all locations with various train and test split. Figures 7 and 8 show actual vs predicted values for traffic flow at location 0 for ARIMA and SARIMA.

Table 3: Perfromance of statistcal models.

Location	**Model**	**R^2**	**RMSE**	**MAE**
0	ARIMA	0.9582	784	21.3
1	ARIMA	0.804	4590	57
2	ARIMA	0.8786	1492.47	27.29
3	ARIMA	0.803	4725	57
4	ARIMA	0.771	7186	69.3
0	SARIMA	0.987	982.3	24.3
1	SARIMA	0.9497	1183.4	24.7
2	SARIMA	0.886	1389	25.8
3	SARIMA	0.962	894.0	22.85
4	SARIMA	0.937	2018	31.2

6.2 Case study -2 Evaluation of Model B

Evaluation of Model B consists of all ML models implemented.

• Gradient Boost Regression Model: R-squared Values: Ranged are from 0.8928% to 0.9397%, indicating a strong ability to explain the variance in traffic flow. RMSE

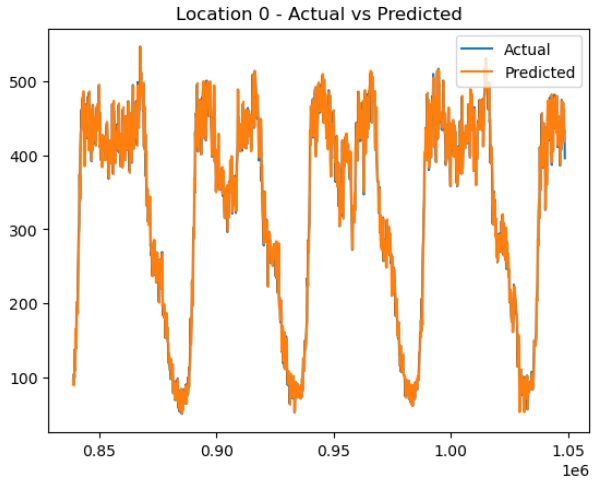

Figure 7: Actual vs predicted values for traffic flow at location 0 for SARIMA

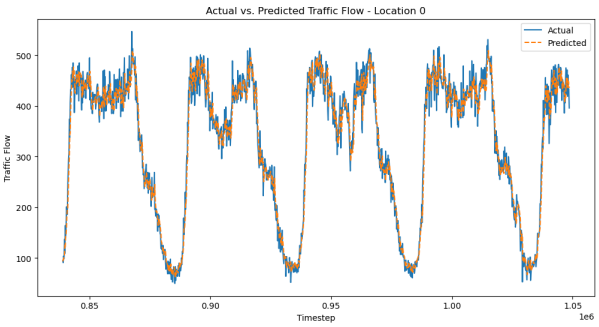

Figure 8: Actual vs predicted values for traffic flow at location 0 for SARIMA

Values: Ranged are from 35 to 43, suggesting reasonable to high accuracy. MAE Values: Ranged are from 28 to 33, indicating reasonable accuracy. Location 3: Highest R-squared value (0.9397%) and lowest RMSE (38.0564), indicating excellent fit and high prediction accuracy. Overall: gradient boost is Reliable tool for predicting traffic flow with superior performance across all five locations.

• KNN Regression Model: R-squared Values recorded are Consistently high, ranging from 0.9333% to 0.9721%. RMSE Values are Moderate, ranging from 24 to 30, highlighting accurate traffic flow prediction. MAE Values: are Moderate, ranging from 19 to 24, indicating reasonable accuracy. Overall: Reliable and accurate asset in traffic flow prediction. Specific characteristics and challenges in each location should be considered for further improvements.

• Decision Tree Regression Model: R-squared Values: are extremely high, ranging from approximately 0.9398% to 0.9999%. RMSE Values: recorded are Low, ranging from 1.3 to 38, suggesting excellent accuracy. MAE Values: are Consistently low, indicating exceptional prediction accuracy. Location 2: is Near-perfect fit with an exceptionally high R-squared (0.9999%) and extremely low RMSE (1.3556). Overall: Highly effective tool for capturing temporal patterns in traffic flow with exceptional accuracy.

• Random Forest Regression Model: R-squared Values: Near-perfect, ranging from approximately 0.9846% to 0.9958%. RMSE Values are Consistently low, varying from 9 to 12. MAE Values: are low, indicating high accuracy. Feature Importance: 'Timestep' is crucial (100% importance), while 'location' does not contribute significantly. Overall: Exceptional performance with 'timestep' is identified as the key driver of traffic flow predictions. Consistent accuracy across locations indicates reliability.

In summary, all three models Gradient Boost, KNN, and Random Forest show robust performance in predicting traffic flow, each with its strengths and areas of exceptional accuracy. The Decision Tree and Random Forest models show outstanding fits to the data. Continuous monitoring and potential adjustments are recommended for sustained performance, especially in specific locations with unique characteristics.

Figure 9 shows a line plot of mapping of actual and predicted data with random forest covering data exceptionally well.

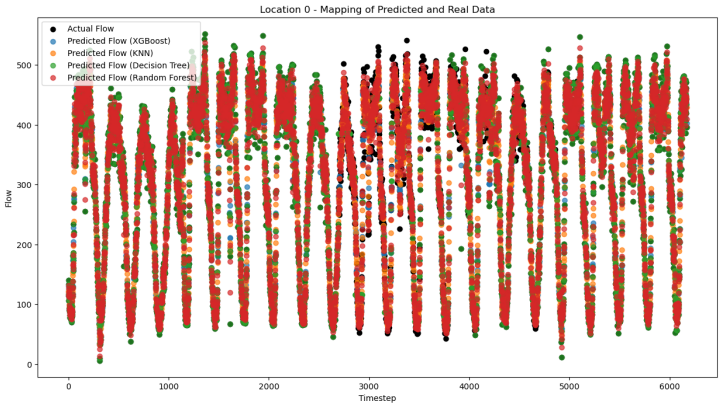

Figure 9: mapping of real and predicted data of ML models

After performing k cross-validation, k-fold results for different regression models demonstrate consistent and robust performance across various locations. Random Forest achieves high average R-squared (0.9615), low RMSE (27.0817), and moderate MAE (20.3550)

Table 4: Perfomance of Machine Learning models

Location	Model	R^2	RMSE	MAE
0	gradient boost	0.9325	35	28
1	gradient boost	0.9255	41	33
2	gradient boost	0.8928	38	29
3	gradient boost	0.9397	38	30
4	gradient boost	0.936	43	33
0	Decision Tree	0.986	15	5
1	Decision Tree	0.985	18.4	5.9
2	Decision Tree	0.99	1.3	0.1
3	Decision Tree	0.9398	38	10.04
4	Decision Tree	0.9957	11	3
0	KNN	0.964	26	19
1	KNN	0.967	27.4	20..8
2	KNN	0.933	21	9.83
3	KNN	0.97	25	19
4	KNN	0.962	33	24
0	Random forest	0.994	11.68	9.20
1	Random forest	0.9954	11.68	7
2	Random forest	0.9899	11.68	10
3	Random forest	0.9958	11.68	7
4	Random forest	0.9947	11.68	9.20

across locations, indicating reliable predictive capabilities. K-Nearest Neighbors: Shows robust performance with high average R-squared (0.9661), low RMSE (25.4110), and moderate MAE (19.1112) across locations, indicating effective capture of temporal patterns. Decision Tree: Performs well with moderate to high average R-squared (0.9454), varying RMSE (32.2417), and MAE (24.0489) across locations, demonstrating solid predictive accuracy. XGBoost: Demonstrates consistent and impressive performance with high average R-squared (0.9632), low RMSE (26.4915), and moderate MAE (19.9808) across locations, highlighting its effectiveness in capturing complex relationships.

6.3 Case Study -3 Evaluation of Model C

Evaluation of Model C consists of analyzing three deep learning models of LSTM, simple RNN, and CNN for traffic prediction of the first five locations.

• LSTM Model: Performance Metrics: R2 scores close to 1 (0.95-0.96), low RMSE between 28 to 35 and MAE values 21 to 26. Consistency: Consistently satisfactory performance across all locations. Training: Loss decreases over epochs, indicating effective learning and generalization. Training Details: Trained for 100 epochs, batch size of 32, 50 LSTM layers. Stability: Consistent performance across epochs suggests the model is not overfitting. Effectiveness: Strong predictive performance for traffic patterns across various locations.

• Simple RNN Model: Performance Metrics: Good R2 scores (0.8786-0.9185), higher RMSE and MAE compared to LSTM. Consistency: Performs consistently across all locations but slightly less than LSTM. Training: Loss decreases with training but does

not match LSTM's overall performance. Comparison: Slightly lower overall performance metrics compared to LSTM.

• CNN Model: Performance Metrics: High R2 values (0.91-0.96), low RMSE between 21 to 25and MAE. Learning: Shows effective learning with decreasing loss. Consistency: Consistent performance across epochs with no significant issues. Optimization: Suggests potential for further optimization in Location 2.

K Cross-Validation Results: LSTM Models: Consistently high performance across locations, average R2 score around 0.958. Comparison: Simple RNN performs well but with slightly lower R2 scores and higher error metrics. CNN Models: Reliable performance, similar R2 scores to LSTM, slightly higher RMSE and MAE.

Overall Recommendation: LSTM is a robust choice, providing a balance of high accuracy and generalization across various locations. In summary, the LSTM model demonstrates strong and consistent performance, making it a recommended choice for the traffic prediction task. followed by CNN and simple RNN models who also perform well but with slight variations in performance metrics. Figure 10 shows plot for Actual vs predicted traffic flow of LSTM for location 0.

Table 5: Perfomance of Deep learning models across locations

Location	**Model**	**R^2**	**RMSE**	**MAE**
0	LSTM	0.958	28.3	21.4
1	LSTM	0.963	29.3	21.97
2	LSTM	0.914	34.8	25.05
3	LSTM	0.966	28	21
4	LSTM	0.9585	35.25	26.32
0	CNN	0.9585	28.1	21.2
1	CNN	0.9623	29	22.5
2	CNN	0.912	35.2	25.4
3	CNN	0.966	28.5	21.6
4	CNN	0.957	35.6	25.3
0	RNN	0.903	43.04	35.7
1	RNN	0.923	42.43	34.8
2	RNN	0.87	42.1	30.9
3	RNN	0.917	44.6	34.8
4	RNN	0.911	51.4	38.9

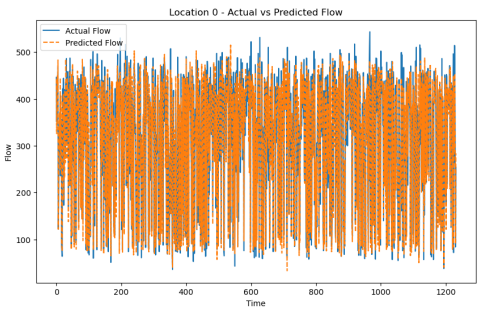

Figure 10: Actual vs predicted traffic flow of LSTM for location 0

6.4 Discussion

Overall, three case studies which included, two statistical Models, four machine learning models, and three deep learning models were implemented and evaluated with varying training and testing data ranging from 50 to 90 for the first locations of the dataset to predict their traffic flow.

Out of the two Statistical models ARIMA and SARIMA.

ARIMA relies on the differencing of past observations to achieve stationarity, making it suitable for linear trends. The simplicity of ARIMA is attributed to its parameterization, comprising three components (p, d, q) for autoregression, differencing, and moving average. ARIMA may struggle in scenarios where traffic flow exhibits non-linear patterns, abrupt changes, or intricate dependencies. SARIMA extends ARIMA by incorporating seasonality components (P, D, Q) to capture periodic patterns. Its success in outperforming ARIMA suggests its superior ability to model time series with both linear trends and seasonal variations. SARIMA models outperformed ARIMA models in terms of R-squared, MSE, and MAE for the given locations. SARIMA's ability to capture seasonality and trends in the data seems to contribute to its better performance, but extremely high MSE values as compared to Machine learning and deep learning models suggest that on average, making larger errors in predicting the traffic flow for the given locations.

Out of four Machine learning models of KNN, Gradient-boosting, and decision tree evaluated based on various training and testing sets, Random Forest performed best model with its accuracy and precision for all locations with the highest R2 RMSE and MSE followed very closely by a decision tree with a small difference.

Ensemble Learning and Complex Relationships: Gradient Boosting builds an ensemble of weak learners sequentially, each correcting error of the previous. This mitigates bias and variance issues, allowing the model to capture complex relationships by focusing on areas with prediction errors.

Decision Tree: Interpretability and Low RMSE/MAE: Decision Trees create a hierarchical structure of decisions based on features, offering interpretability. The tree structure contributes to achieving low RMSE and MAE as it effectively partitions the data into homogeneous groups.

KNN: Proximity-Based Predictions: KNN predicts based on the proximity of data points in feature space. The choice of distance metric (e.g., Euclidean, Manhattan) impacts the model's sensitivity to feature scales and influences performance.

Random Forest: Ensemble Robustness: Random Forest builds multiple decision trees, each trained on a subset of data and features. This ensemble approach enhances robustness by reducing overfitting and capturing diverse patterns in the data. All models perform well, with high R-squared values indicating a good fit to the data but the choice of the best model depends on the specific requirements and characteristics of the data.

Out of three deep learning models LSTM, RNN, and CNN, LSTM is a strong contender, with comparable performance across locations with high R2 low MSE and MAE scores closely followed by CNN as compared to RNN performs well but shows some limitations.

LSTM: Capturing Long-Term Dependencies excels in capturing long-term dependencies using memory cells that can store and retrieve information over extended periods. The model is well-suited for traffic flow prediction where dependencies span multiple time steps.

CNN: Spatial Feature Extraction: CNN's effectiveness lies in spatial feature extrac-

tion, suitable for capturing spatial patterns in traffic flow. Convolutional layers learn spatial hierarchies, enabling the model to recognize patterns at different scales.

RNN:Long-Term Dependency Challenges' struggles with vanishing or exploding gradient problems, limiting its ability to capture long-term dependencies. This limitation can manifest in scenarios where traffic flow exhibits extended temporal dependencies.

After performing k cross-validation this was further proven that LSTM's and CNN performance.

From the Evaluation of the case study of A, B, and C, ML models have shown consistent and accurate performance lowest RMSE and MAE with proven robustness that can be visualized through charts and tables, this was followed by deep learning models followed by statistical models. This is attributed to ML model's adeptness at capturing complex relationships in the traffic flow data, leveraging the ensemble nature of algorithms of Random Forest Decision Tree, KNN, and gradient boosting demonstrating high interpretability. Out of ML models, there is a clear scope in the future of integrating ML models and performing a highly accurate and robust model. The minor limitation of research includes considering small size of data considering immense size would be interesting to watch the performance of these models.

Overall, the project was extremely successful considering the models' performance of A, B, and C. The research's results are a great encouragement in the field of traffic prediction as ML and DL models show exceptional results that will effectively traffic management strategies, optimize resource allocation, and facilitate timely interventions to improve overall traffic efficiency and reduce congestion. The research meets its objective of developing and implementing a high-performance model that is effective in the practical world of traffic management. Bar graphs of R2 values of all models

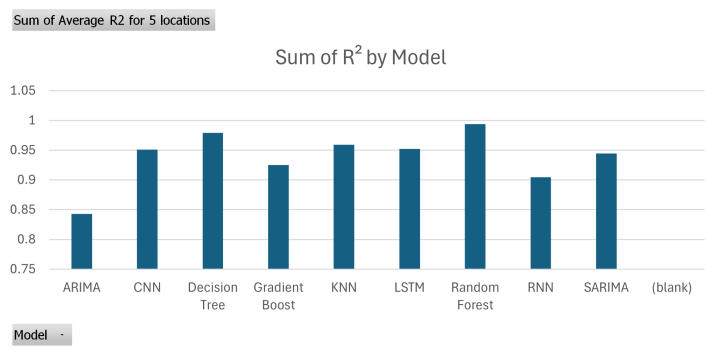

Figure 11: Bar graphs of R2 values of all models

7 Conclusion and Future Work

The overarching goal of the project is to find the potential of various ML, DL, and statistical models in the field of short-term traffic flow prediction and to find the most optimum model suitable for its application in the field of traffic management systems. The research was extremely successful in implementing Statistical, Machine learning, and deep learning traffic prediction with remarkable accuracy and identified the shortcomings of each model. Out of which average performance of ML models in the likes of random forest achieved R2 of 0.99, decision tree with 0.98 and 0.97 performed exceptionally well

followed by LSTM with R2 of 0.95.8, CNN with 0.95.2, and RNN with 0.91 Statistical models like Sarima and Arima performed well too with high accuracy of 0.81, and 0.95 respectively, but a higher RMSE and MAE make them the latter choice. With this research ML models of random forest, Decision tree, and KNN remain the first choice. The models developed in this study provide a solid foundation for real-time traffic monitoring, resource optimization, and data-driven decision-making. Their applications include adaptive signal control and dynamic route planning, contributing to a more responsive and efficient traffic management infrastructure. This study sets the stage for intelligent traffic management solutions to alleviate congestion and enhance overall transportation system performance. The limitation of work would be the sizeable use of dataset, it would be interesting to know if the models that performed well can retain their results with increase in the size of dataset.

In future works with the use of the PEMS 08 dataset with each model having some drawbacks, they can be integrated to find a robust and accurate model which can reduce traffic congestion at a location. In addition to this traffic congestion is a huge topic and several factors come into picture that led to it, and amalgamation of weather datasets and accident datasets could provide us with a more comprehensive way of providing actionable insights for effective traffic management and improved transportation systems

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician* **46**(3): 175–185.

Box, G. E. and Jenkins, G. M. (1976). Time series analysis forecasting and control-rev.

Breiman, L. (2001). Random forests, *Machine learning* **45**: 5–32.

Chen, W., An, J., Li, R., Fu, L., Xie, G., Bhuiyan, M. Z. A. and Li, K. (2018). A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features, *Future generation computer systems* **89**: 78–88.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine, *Annals of statistics* pp. 1189–1232.

Gomes, B., Coelho, J. and Aidos, H. (2023). A survey on traffic flow prediction and classification, *Intelligent Systems with Applications* p. 200268.

Hockstad, L. and Hanel, L. (2018). Inventory of us greenhouse gas emissions and sinks, *Technical report*, Environmental System Science Data Infrastructure for a Virtual Ecosystem ....

Kumar, S. V. and Vanajakshi, L. (2015). Short-term traffic flow prediction using seasonal arima model with limited input data, *European Transport Research Review* **7**(3): 1–9.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE* **86**(11): 2278–2324.

Li, C. and Xu, P. (2021). Application on traffic flow prediction of machine learning in intelligent transportation, *Neural Computing and Applications* **33**: 613–624.

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

Meena, G., Sharma, D. and Mahrishi, M. (2020). Traffic prediction for intelligent transportation system using machine learning, *2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, IEEE, pp. 145–148.

Mohammed, O. and Kianfar, J. (2018). A machine learning approach to short-term traffic flow prediction: A case study of interstate 64 in missouri, *2018 IEEE International Smart Cities Conference (ISC2)*, IEEE, pp. 1–7.

Nakat, Z., Herrera, S. and Cherkaoui, Y. (2014). Cairo traffic congestion study: Executive note, *The World Bank Group* pp. 1–5.

Polson, N. G. and Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction, *Transportation Research Part C: Emerging Technologies* **79**: 1–17.

Sayed, S. A., Abdel-Hamid, Y. and Hefny, H. A. (2023). Artificial intelligence-based traffic flow prediction: a comprehensive review, *Journal of Electrical Systems and Information Technology* **10**(1): 13.

Shao, H. and Soong, B.-H. (2016). Traffic flow prediction with long short-term memory networks (lstms), *2016 IEEE region 10 conference (TENCON)*, IEEE, pp. 2986–2989.

Shekhar, S. and Williams, B. M. (2007). Adaptive seasonal time series models for forecasting short-term traffic flow, *Transportation Research Record* **2024**(1): 116–125.

Sun, P., Aljeri, N. and Boukerche, A. (2020). Machine learning-based models for real-time traffic flow prediction in vehicular networks, *IEEE Network* **34**(3): 178–185.

Tan, M.-C., Wong, S. C., Xu, J.-M., Guan, Z.-R. and Zhang, P. (2009). An aggregation approach to short-term traffic flow prediction, *IEEE Transactions on Intelligent Transportation Systems* **10**(1): 60–69.

Willetts, E., thank Carlos, D. C.-L. W., Corvalan, M. M. and Neville, T. (n.d.). Review of ipcc evidence 2022.

Yang, H.-F., Dillon, T. S. and Chen, Y.-P. P. (2016). Optimized structure of the traffic flow forecasting model with a deep learning approach, *IEEE transactions on neural networks and learning systems* **28**(10): 2371–2381.

Yi, H., Jung, H. and Bae, S. (2017). Deep neural networks for traffic flow prediction, *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)* pp. 328–331.
URL: *https://api.semanticscholar.org/CorpusID:2421567*

Zheng, J. and Huang, M. (2020). Traffic flow forecast through time series analysis based on deep learning, *IEEE Access* **8**: 82562–82570.