Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Vehicle Density Prediction from Historical Traffic Dataset using Machine Learning Techniques

Md. Mazaherul Islam
Department of Computer Science and Engineering,
Jadavpur University
Kolkata, India

Zeenat Rehena
Department of Computer Science and Engineering,
Aliah University
Kolkata, India

Nandini Mukherjee
Department of Computer Science and Engineering,
Jadavpur University
Kolkata, India

***Abstract* - Traffic congestion is a serious problem for urban areas around the world, negatively impacting the economy and quality of life while increasing carbon emissions. Accurate prediction of vehicle density plays a crucial role in the development of Intelligent Transportation Systems (ITS), as accurate vehicle density predictions are essential for proactively managing traffic, enabling adaptive traffic signal control, and providing real-time route recommendations for vehicles. This study compares the performance of six different Machine Learning (ML) models in vehicle density prediction based on historical spatiotemporal datasets. The selected models are Linear Regression (LR), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Support Vector Regressor (SVR), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM). The historical dataset contains vehicle density measured by hourly vehicle counts, as well as relevant features such as date, weekday, and journey hour (Peak hour and off-peak hour), which are used to conduct a comparative analysis of each machine learning model. The trained models are evaluated based on Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R^2 Score). The results demonstrate the potential of machine learning in improving the accuracy of vehicle density predictions. XGBoost, among the six models, achieves the highest predictive accuracy.**

Keywords* - *Intelligent Transportation System (ITS); Traffic Congestion; Machine Learning

I. INTRODUCTION

Growing populations, vehicle ownership, and traffic congestion have accelerated the pace of urbanization, but there is a need to make cities more sustainable and livable. According to the International Transport Forum, even if existing commitments to decarbonize the transport system are fully implemented, global vehicle numbers are projected to double by 2050, and transport-related emissions will increase by 16% compared to 2015 [1].

Predicting traffic congestion density is inherently complex due to nonlinear spatio-temporal dependencies, sudden congestion, weather disruptions, and travel-time variations (e.g., peak and off-peak hours and weekends). Historically, parametric statistical models such as Autoregressive Integrated Moving Average (ARIMA) have dominated this field. However, this model fails to capture complex nonlinear patterns and the interactions of multidimensional features.

Vehicle density, defined as the number of vehicles occupying a given length of roadway or lane at a specific instant, serves as a fundamental metric in traffic flow theory. Accurate prediction of this variable helps transportation systems to optimize traffic light timing, dispatch emergency services efficiently, and instruct navigational apps to reroute vehicles. This paper presents a comprehensive evaluation of Machine Learning (ML) models applied to historical spatio-temporal traffic datasets to predict vehicle density. Further, this predicted vehicle density may be a useful operator for several on-road services for smooth control of the overall road traffic system. One of the prime services of an Intelligent Transportation System is traffic congestion detection and its proper control, which is our main objective in this research work.

The remainder of this paper is organized as follows. Section II describes the related works. Section III provides the methodology and dataset description. Section IV explains the experimental setup and evaluation Metrics. Section V describes the model implementation and result analysis. Finally, Section VI provides the conclusion of this work.

II. RELATED WORK

The development of vehicle density prediction methods can be broadly divided into two distinct eras: Classical machine learning and Deep learning.

A. Classical Machine Learning

With the advent of big data in the transportation system, data-driven models have gained popularity. Support Vector Regression (SVR) has emerged as a powerful tool, as it can map inputs into high-dimensional space using kernel functions and effectively handle nonlinear traffic relationships. Further Tree-Based models, such as random forests (RF) and eXtreme Gradient Boosting (XGBoost), have advanced this field [2-3]. Researchers have found that tree-based methods inherently

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

handle missing values and perform well with tabular data containing categorical variables [4-5].

B. Deep Learning

Recently, deep learning models have revolutionized traffic state forecasting. Traffic data collected from various sensors is largely sequential. So, recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) networks and gated recurrent units (GRUs), have been widely adopted. The vanishing gradient problem of conventional RNNs is solved by LSTM, which makes RNNs very effective in retaining long-term historical memories, such as identifying weekly travel patterns.

In the past, many research works on vehicle density prediction were accomplished using ML and DL models, such as:

Usman, A. et al. [6] applied five ML models to predict traffic density to improve bus scheduling and reduce passenger delays. The dataset used to train the models has more than 1,048,576 instances with attributes like date, time, vehicle count, average speed, etc. The result shows that the Logistic Regression model is strongly reliable and outperforms the other models with an accuracy of 99.91%.

Authors in [7] focus on vehicle detection and traffic density estimation using deep learning to improve traffic management systems through adaptive control of traffic lights. For preparing datasets, they used various open-source libraries that are FLIR (thermal & RGB), MB7500 (drone images), and KITTI (urban & highway driving) with varied conditions (day/night, weather, urban/rural). The used models, Faster R-CNN and SSD (Single Shot Detector), were ensemble to form a hybrid model using a majority voting classifier to improve robustness and resolve duplicate detections. As per their result analysis, this hybrid model outperformed the individual models. On the FLIR thermal dataset, the ensemble achieved 94% mAP, which was 34% better than SSD and 6% better than Faster R-CNN.

In [8], the authors show how LSTM-based deep learning models outperform traditional machine learning models like decision trees, random forest, linear regression, and even classical deep learning in traffic density prediction based on MAE and RMSE values. The model was configured with 20 neurons in the hidden layer and the Adam optimizer. They use Istanbul's open traffic dataset and find that LSTM can capture long-term dependencies and time-based variations more effectively. This makes LSTM particularly well-suited for intelligent transportation systems, helping reduce traffic congestion at intersections and supporting smart city initiatives by improving urban traffic management.

The authors in [9] shows SVR model can be applied to achieve more accurate predictions of traffic density using real-world traffic data collected from California highways. The study compares Linear Regression, Linear SVR, Quadratic SVR, and Gaussian SVR models. The results demonstrate that the Gaussian SVR model consistently outperforms the others, achieving higher accuracy and lower error rates. This approach highlights the potential of machine learning to enhance Intelligent Transportation Systems by avoiding traffic congestion, improving road safety, and reducing pollution.

In a review paper [10], the authors emphasize the growing application of machine learning, deep learning, and IoT technologies in traffic prediction and congestion control. They pointed out that conventional static traffic management approaches are inadequate to address modern urban challenges, while AI-driven approaches such as SVM, KNN, CNN-based detectors such as YOLO, genetic algorithms, fuzzy logic, swarm intelligence, and reinforcement learning provide more adaptable and effective solutions. The study also highlights challenges such as data quality, infrastructure limitations, sensor dependency, and privacy concerns, but concludes that the combination of ML and IoT can significantly improve traffic flow, reduce congestion, and enhance passenger safety, paving the way for a smarter and more resilient transportation system.

P. Shetty et al. [11] explore how machine learning models can be applied to predict traffic patterns using historical data. The study compares multiple models, including Random Forest, SVM, Gradient Boosting (XGBoost, LightGBM, CatBoost), ElasticNet, LSTM, CNN-LSTM, SVR-LSTM, stacking, and a Combined Ensemble. The results show that ensemble methods such as Random Forest, Stacking, and Combined Ensemble achieved the highest accuracy ($R^2 = 1.0$) with very low error rates, while individual models such as LSTM and SVR struggled to capture complex traffic behaviour. The study highlights that hybrid and ensemble methods are more reliable and generalizable, making them suitable for real-world smart city traffic management.

H. Xin et al. [12] use data from five administrative districts of New York City to present a multiple linear regression model built in R language that predicts urban traffic jam durations and waiting times. The authors identified key influencing factors like average travel speed, peak hours, travel time, delay time, and day (weekday or weekend), and created a regression equation using these variables. Their model achieved an R^2 score of 0.9458, showing a strong correlation between travel time and delay time. This also confirms the significant impact of peak hours on traffic congestion. Overall, the model shows that multiple linear regressions can effectively process big traffic data to forecast congestion durations and enhance the efficiency of route planning.

The conventional methods like time series analysis, non-parametric regression, and neural networks have been used for traffic prediction, but Random Forest provides faster computation, higher robustness, and strong applicability [13]. The model includes variables like time of day, weather, special road conditions, holidays, road quality, and classifies congestion levels as smooth, congested, or blocked using a congestion index. By creating multiple pruned and unpruned CART decision trees through bootstrap sampling and combining their results via majority voting, Random Forest achieves a prediction accuracy of 87.5% with low generalization error. The study also shows that the model can identify the relative importance of environmental factors affecting congestion, with weather and time duration being particularly significant.

A cloud-based traffic management framework designed to alleviate urban traffic congestion using real-time data from GPS trackers, traffic sensors, and weather APIs [14]. The proposed system applies Random Forest to estimate traffic density and optimize both routes and traffic signals. The system demonstrates a 30% reduction in vehicle density and an

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

18% increase in speed limits, proving its effectiveness in improving urban mobility.

As an ensemble learning method, the Random Forest model combines multiple decision trees with additional randomness to improve classification and regression tasks [15]. Standard trees split nodes using all predictors, whereas Random Forest selects a random subset of predictors at each node. Thus, the Random Forest model improves performance and reduces overfitting. This study explains the following steps: bootstrap sampling, tree construction, and out-of-bag (OOB) error estimation to achieve an accurate measure of prediction error. It also emphasizes the additional outputs such as variable importance measures and proximity matrices, which are useful for model reduction and unsupervised learning. The authors illustrate the Random Forest model as a powerful and user-friendly tool for data analysis for its robustness, ease of use with few parameters, and accuracy compared to support vector machines.

S. Vajeeda et al. [16] present a route optimization and real-time traffic prediction framework using a hybrid machine learning model called XatBoost, which combines the strengths of XGBoost and CatBoost. This study focuses on Chennai traffic data with features like vehicle count, average speed, day type, time of day, and weather. They have tested several models such as logistic regression, decision tree, random forest, CatBoost, and XGBoost. The XGBoost model is the most effective, achieving an accuracy of 87.2% and demonstrating strong robustness across various validation techniques. The analysis also revealed that rainfall increases the probability of congestion by about 20%. The proposed XatBoost model increases the stability, interpretability, and accuracy of the predictions and provides a scalable solution for route optimization and urban traffic management.

R. Rathore et al. [17] present a machine learning and IoT-based framework for traffic flow prediction in ITS, focusing on the growing problem of urban traffic congestion, which causes time loss, fuel consumption, and pollution. After preprocessing the historical traffic data at hourly intervals, they applied linear regression with min-max scaling, which was then extended to Support Vector Machine (SVM) models for better accuracy.

The authors in [18] proposed density prediction in the transportation system of Bandung city in Indonesia by applying Support Vector Machine (SVM) on GPS time series data. They collected more than 35,320 records from the public transportation dataset, processed them through normalization and encoding, and divided them into training (70%) and test (30%) sets. The SVM model was chosen due to its strong performance in classification tasks compared to other methods such as neural networks or particle swarm optimization. Performance metrics such as precision, recall, and F1-score also ensured high reliability, with precision and recall values close to 1.0. Overall, this study concludes that SVM is highly effective in predicting traffic density from GPS data and provides a robust solution for urban traffic control and congestion management.

H. Verma et al. [19] propose an effective, scalable, and accurate deep learning-based traffic density prediction system, which integrates YOLOv8 for real-time vehicle detection with a stacked LSTM model for time-series prediction. The proposed system architecture has three main units: data collection (via fixed cameras at road junctions), data transmission (wired/wireless), and data processing (vehicle detection using YOLOv8 and then density prediction using LSTM). YOLOv8 was pre-trained on the COCO dataset, and it achieved 93.36% accuracy with high precision (98.91%), making it useful for real-time vehicle counting. The stacked LSTM model was trained on the traffic dataset and compared with RNN models. The results show that the stacked LSTM achieves better prediction accuracy with a lower RMSE value (average 0.59) than the RNN (0.73).

A comprehensive survey of recent advances in AI-based traffic forecasting and emerging opportunities in intelligent transportation systems is well explored in [20]. The study highlights the global challenge of traffic congestion, which wastes travel time, leads to fuel consumption, and emphasizes the importance of accurate traffic forecasting for intelligent traffic management. The review classifies traffic data into different types like sensor data, GPS, trip records, network infrastructure, etc. Also, discusses pre-processing techniques and compares classical approaches (e.g., ARIMA, Kalman filter, decision trees) with modern machine learning and deep learning methods such as CNNs, RNNs, and multivariate traffic time series models, which capture complex spatio-temporal dependencies.

The authors in [21] present a comparative study of three machine learning algorithms: decision trees, logistic regression, and neural networks, for predicting traffic congestion using large-scale sensor data from the GCM (Gary-Chicago-Milwaukee) corridor. The dataset included more than 1.2 million samples collected over multiple days. The models were implemented and tested in both Python (TensorFlow) and Clementine environments. The results show that the decision tree consistently outperforms the other methods, achieving the highest accuracy (97% in Python and 97.65% in Clementine).

In the above-mentioned existing research works, some studies focus only on a limited number of models without providing a broader comparison. Some studies have limited real-world validation, scalability issues, and underutilization of multimodal data. Few studies highlight advanced and hybrid models but have high computational overhead. In contrast, our selected six machine learning models are simple and lightweight in computational tasks. Moreover, our historical traffic dataset is classified into three different categories based on day of week and hour of day, which enables the models to increase the accuracy level in vehicle density prediction.

III. METHODOLOGY AND DATASET DESCRIPTION

We aim to predict vehicle density on the lanes connected to the different intersections in an urban road network shown as a graph in Fig. 1. In this road network, there are 16 intersections, and during the journey of any vehicle from intersection 1 to the last intersection 16, the vehicle has a total of 24 available routes to choose from.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

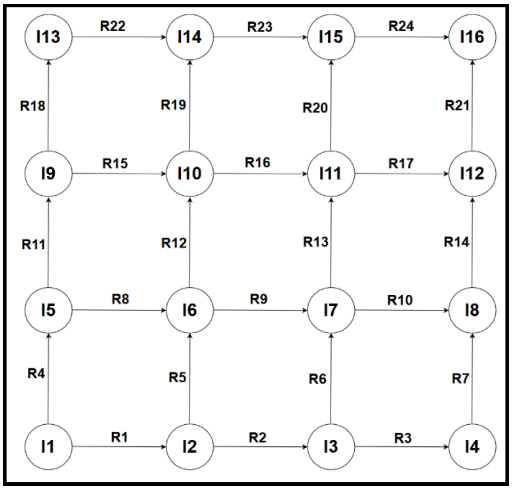


Fig. 1: Graph of an urban road map

A. *Dataset Generation*

This section describes how the dataset is prepared for analysis of vehicle density prediction by the selected six machine learning models. The dataset utilized in this research consists of continuous, historical traffic state records across multiple lanes of the urban road network over six years.

Now, to prepare a dataset on vehicle density (VD) along with other parameters for those 24 lanes, an early-prepared large dataset from Kaggle [22] has been used.

The Kaggle dataset was prepared considering six different geographical areas. As this dataset does not meet our requirement for a lane-wise vehicle density dataset, we have modified it as follows. First, we prepare three dataset formats considering day and time: peak hours on weekdays, off-peak hours on weekdays, and weekends 24 hours. Then the date field is added as an index in the modified dataset for each category. From the beginning of 2020 to the end of 2025, a total of 1566 different dates have been selected for the first two category datasets, and 784 different dates have been selected for the third category dataset. A specific area from the Kaggle dataset is represented as a single lane from our selected urban road network in the modified datasets. The values of vehicle type, weather, economic condition, hour of day, speed, energy consumption, and vehicle density fields remain unchanged, and other irrelevant fields are not considered in the modified datasets. Thus, a total of 24 lane-wise modified datasets for the said three different categories are prepared for vehicle density prediction for each lane. Table I summarizes the attributes of the dataset.

After necessary modifications to the original datasets, finally we get desired datasets for each of the 24 lanes of each category of the three different types of datasets. Table II for lane 5, Table III for lane 12, and Table IV for lane 17 show examples of lane-wise historical vehicle density (HVD) on a particular lane collected on different dates and times. The entire dataset is divided into three portions: (i) data collected during peak hours on weekdays (07:00 -- 9:00 and 17:00 -- 19:00), (ii) data collected during off-peak hours on weekdays (00:00 -- 6:00, 10:00 -- 16:00, and 20:00 -- 23:00), and (iii) data collected throughout the day (24 hours) on weekends.

TABLE I. DATASET COLUMNS AND DESCRIPTIONS

Column Name	Description
Date	Date of recording information mentioned below
Day of week	Day of the week
Lane	Specific lane between two intersections
Vehicle type	Type of vehicle (e.g., Car, Autonomous vehicle)
Weather	Current weather (e.g., Clear, Rainy)
Economic condition	Economic state of the city (e.g., Stable, Recession)
Hour of day	Hour of the day when the data was recorded
Speed	Recorded vehicle speed
Energy consumption	Estimated energy consumption based on vehicle type and speed
Vehicle density	Density of traffic at the time of recording

TABLE II. EXAMPLE OF HVD COLLECTED IN WEEKDAY PEAK HOURS

Date	Day of Week	Lane	Peak Hours	HVD
01.01.2020	Wednesday	Lane 5	18:00	0.1821
02.01.2020	Thursday	Lane 5	08:00	0.3704
03.01.2020	Friday	Lane 5	07:00	0.2398
06.01.2020	Monday	Lane 5	08:00	0.3388
07.01.2020	Tuesday	Lane 5	08:00	0.5689
08.01.2020	Wednesday	Lane 5	08:00	0.3532
.....	.....	.....	.....	.....
31.12.2025	Wednesday	Lane 5	7:00	0.1937

TABLE III. EXAMPLE OF HVD COLLECTED IN WEEKDAY OFF-PEAK HOURS

Date	Day of Week	Lane	Off-Peak Hours	HVD
01.01.2020	Wednesday	Lane 12	09:00	0.1637
02.01.2020	Thursday	Lane 12	00:00	0.3232
03.01.2020	Friday	Lane 12	08:00	0.2721
06.01.2020	Monday	Lane 12	06:00	0.0495
07.01.2020	Tuesday	Lane 12	23:00	0.2535
08.01.2020	Wednesday	Lane 12	00:00	0.2723
.....	.....	.....	.....	.....
31.12.2025	Wednesday	Lane 12	10:00	0.3501

TABLE IV. EXAMPLE OF HVD COLLECTED IN WEEKEND 24 HOURS

Date	Day of Week	Lane	24 Hours	HVD
04.01.2020	Saturday	Lane 17	17:00	0.2332
05.01.2020	Sunday	Lane 17	23:00	0.2498
11.01.2020	Saturday	Lane 17	06:00	0.1923
12.01.2020	Sunday	Lane 17	18:00	0.3871
18.01.2020	Saturday	Lane 17	17:00	0.2582
19.01.2020	Sunday	Lane 17	23:00	0.3919
.....	.....	.....	.....	.....
28.12.2025	Sunday	Lane 17	04:00	**0.0816**

Next, to train the selected models and predict vehicle density from the prepared historical dataset, the following steps are performed as shown in Fig. 2.

B. *Pre-processing and Feature Engineering*

Raw traffic data may contain missing values, noise, and cyclical artefacts, which require specialized transformations.

1) Handling missing values: The selected models were implemented with null value checking and handling. Specifically, for all datasets, rows with missing values in the target variable (Vehicle Density) are dropped.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

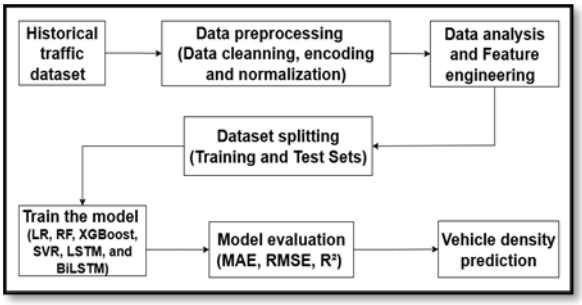


Fig. 2: Vehicle density prediction workflow

2) Data normalization: For the selected models, features were scaled to a range of [0, 1] using Min-Max normalization to prevent gradient explosion and optimize convergence.

$$X_{scaled} = \frac{X - X_{min}}{X_{\max} - X_{min}} \quad (1)$$

Where X is the original value, and X_{min} and X_{max} are the minimum and maximum values of the feature X in the dataset.

IV. EXPERIMENTAL SETUP AND EVALUATION METRICS

In this section, the experimental setup and the performance evaluation metrics used for the selected models are discussed.

A. Experimental Setup

The processed dataset is divided chronologically into 80% training and 20% testing subsets to predict the vehicle density from the processed dataset.

B. Hyperparameter Configuration

Hyperparameter tuning for ML models is performed manually as well as automatically using a specific tuner as required. Some models use Grid-SearchCV and Keras Tuner for optimizing the hyperparameters on the basis of the validation performance. Table V shows the key hyperparameters used for each model.

TABLE V. HYPERPARAMETERS USED FOR THE MODELS

Model	Key Hyperparameters
Linear Regression	random_state=42, fit_intercept=True
Random Forest	n_estimators=200, learning_rate=0.05, max_depth=6, subsample=0.8, colsample_bytree=0.8, random_state=42, objective='reg:squarederror',
SVR	kernel='rbf', C=1.0, epsilon=0.1, n_estimators=200, random_state=42
LSTM	layers=2, units=32, dropout=0.3, dense units= 16, learning_rate=0.01, activation='tanh', optimizer='adam', validation_split=0.2, epochs=100, batch_size=32, early stopping(patience=10), verbose=1
BiLSTM	Bidirectional layers=2, units=64, activation='tanh' and 'relu', return_sequences=True, optimizer=Adam, learning_rate=0.01, validation_split=0.2, epochs=100, batch_size=32, verbose=1
XGBoost	n_estimators=200, learning_rate=0.1, max_depth=3, subsample=0.9, min_child_weight=5, colsample_bytree=0.9, random_state=42

C. Evaluation Metrics

To assess the prediction accuracy and evaluate the performance of the selected models, the accuracy was rigorously quantified using four standard statistical scores: R-squared (R^2), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE). These metrics are obtained as follows:

1) Coefficient of Determination (R^2 Score): Indicates the proportion of variance predictable from the independent variables.

$$R^2 = 1 - \frac{\sum_{k=1}^{n} (p_k - \hat{p}_k)^2}{\sum_{k=1}^{n} (p_k - \bar{p})^2}$$

2) Mean Absolute Error (MAE): Calculates the average magnitude of absolute errors.

$$MAE = \frac{1}{n} \sum_{k=1}^{n} |p_k - \hat{p}_k|$$

3) Root Mean Squared Error (RMSE): Measures the accuracy and reliability of the prediction model and penalizes larger outliers more heavily than MAE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (p_k - \hat{p}_k)^2}$$

Where n denotes the number of observations (sample size of the dataset). For each observation k, $\hat{p}_k$ represents the actual density, while p_k denotes the predicted density. The mean of all observed density values is expressed as $\bar{p}$. These notations are fundamental for defining evaluation metrics. An accurate prediction is characterized by an R^2 score closer to 1 and lower values of other metrics.

V. MODEL IMPLEMENTATION AND RESULTS

Experimental evaluations of linear regression, random forest, XGBoost, SVR, LSTM, and BiLSTM methods have been conducted using modified real data from Kaggle. To assess model performance, R^2 score, MAE, and RMSE are used as evaluation metrics.

A. Experimental results

Following exhaustive training cycles, the experimental quantitative results across the test sets of the models on the three different datasets are summarized in Tables VI, VII, and VIII. Experimentally evaluated results are presented using R^2 score, MAE, and RMSE. The values shown in bold represent the lowest values for MAE and RMSE, and the highest value for R^2.

Figs. 3, 4, and 5 show the comparative visualization of the R^2, MAE, and RMSE values achieved by the six models in the three different datasets. It is found that the XGBoost model achieved the highest R^2 score and the lowest MAE and RMSE values for the three different types of datasets.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

TABLE VI. R^2, MAE, AND RMSE VALUES ACHIEVED FOR THE WEEKDAY PEAK HOURS DATASET.

Model	R^2	MAE	RMSE
Linear Regression	0.6634	0.0946	0.1354
Random Forest	0.7599	0.0726	0.1143
SVR	0.7328	0.0826	0.1206
LSTM	0.7584	0.0728	0.1147
BiLSTM	0.7672	0.0716	0.1126
XGBoost	**0.8013**	**0.0678**	**0.1023**

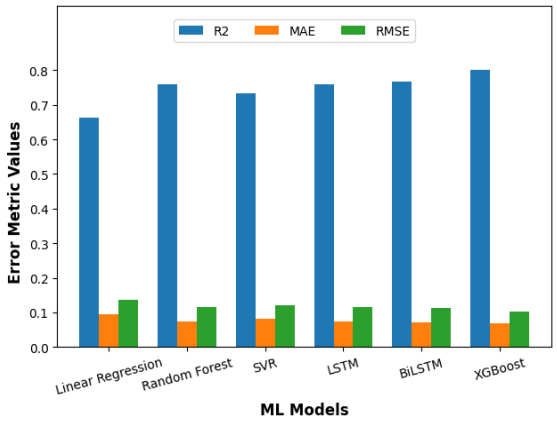


Fig. 3: R^2, MAE, and RMSE values achieved for the weekday peak hours dataset

TABLE VII. R^2, MAE, AND RMSE VALUES ACHIEVED FOR THE WEEKDAY OFF-PEAK HOURS DATASET.

Model	R^2	MAE	RMSE
Linear Regression	0.6134	0.0968	0.1318
Random Forest	0.7204	0.0728	0.1121
SVR	0.6769	0.0861	0.1205
LSTM	0.7112	0.0749	0.1139
BiLSTM	0.7170	0.0736	0.1127
XGBoost	**0.7497**	**0.0719**	**0.1084**

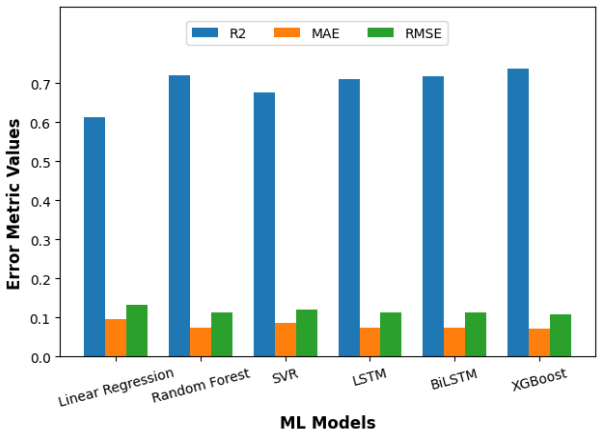


Fig. 4: R^2, MAE, and RMSE values achieved for the weekday off-peak hours dataset

TABLE VIII. R^2, MAE, AND RMSE VALUES ACHIEVED FOR THE WEEKEND 24 HOURS DATASET.

Model	R^2	MAE	RMSE
Linear Regression	0.6435	0.0907	0.1389
Random Forest	0.6838	0.0755	0.1242
SVR	0.6559	0.0840	0.1295
LSTM	0.6855	0.0753	0.1238
BiLSTM	0.7050	0.0764	0.1199
XGBoost	**0.7466**	**0.0731**	**0.1105**

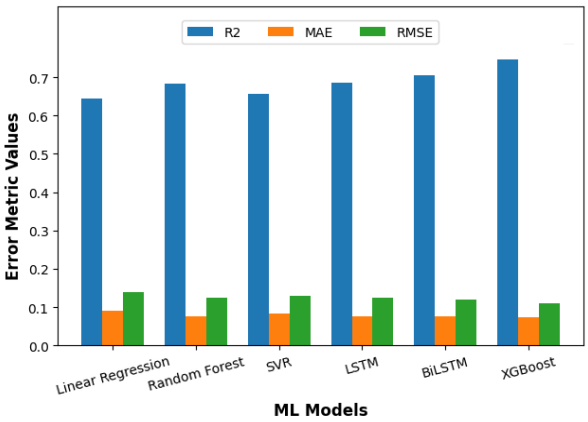


Fig. 5: R^2, MAE, and RMSE values achieved for the weekend 24-hour dataset

B. *Comparative Analysis of Actual and Predicted Values across the Models*

The comparison of actual vs. predicted traffic density for all the models we have trained is presented in separate subplots within a single figure for the three different category datasets. That helps to visualize each model's performance while acknowledging their differing test data splits where applicable.

The scatter plots comparing actual vs. predicted traffic density for all models are shown in Figs. 6, 7, and 8. This comprehensive visualization in the figures allows comparison of how the models, including Linear Regression, Random Forest, SVR, LSTM, BiLSTM, and XGBoost, perform predictions against the actual values in weekday peak hour, weekday off-peak hour, and weekend datasets, respectively. The red dashed line in each subplot represents a perfect prediction, helping to assess each model's accuracy quickly. The deviation of the points from the diagonal line exhibits the model's prediction errors and reflects the challenges of accurately capturing variations in actual vehicle density.

C. *Performance Comparison of the Six Models*

In this section, we discuss the trained models on the basis of their used hyperparameters and achieved predictive performance on the test set of the three different historical traffic datasets.

1) Linear Regression: This model is implemented using the LinearRegression class of Scikit-learn, where fit_intercept=True and random_state=42 have been used as the hyperparameters shown in Table 5. Before training, the dataset is divided into 80% training and 20% testing sets. The training process of the model minimizes the sum of squared residuals to fit the best linear equation to the data.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

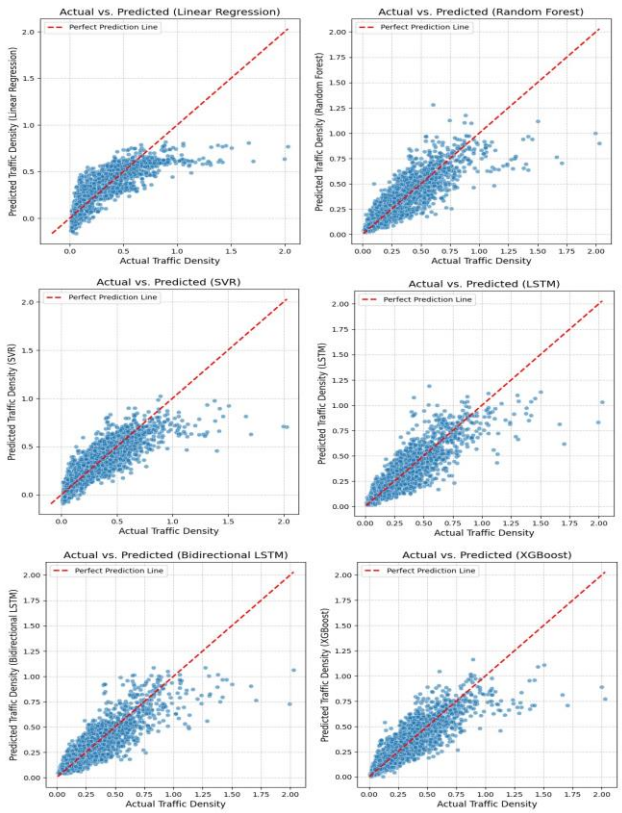


Fig. 6: Actual vs. predicted traffic density scatter plots for all models in the weekday peak hours dataset

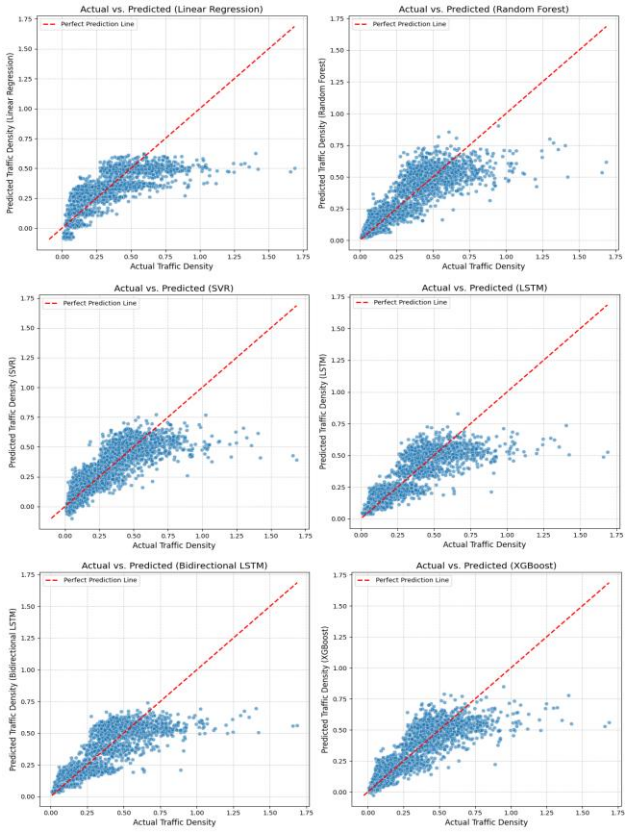


Fig. 7: Actual vs. predicted traffic density scatter plots for all models in the weekday off-peak hours dataset

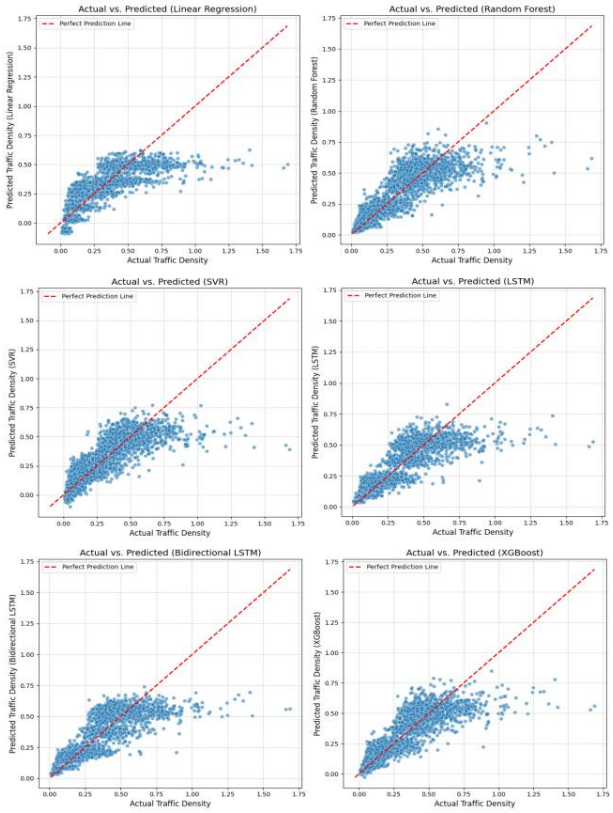


Fig. 8: Actual vs. predicted traffic density scatter plots for all models in the weekday peak hours dataset

Tables VI, VII, and VIII for three different types of datasets summarize the performance of the model across training and testing, where R^2 scores are 0.6634, 0.6134, and 0.6435, MAE values are 0.0946, 0.0968, and 0.0907, and the corresponding RMSE values are 0.1354, 0,1318, and 0.1389. These results state that the model explains only 66%, 61%, and 64% of the variance in vehicle density for the three different datasets separately.

2) Random Forest: This model has been implemented using scikit-learn's RandomForestRegressor and GridSearchCV with 3-fold cross-validation. The best hyperparameter set is n_estimators=200, learning_rate=0.05, max_depth=6, subsample=0.8, colsample_bytree=0.8, bjective='reg:squarederror', random_state=42. On the test set of three datasets, the model has achieved R^2 scores of 0.7599, 0.7204, and 0.6838, MAE values of 0.0726, 0.0728, and 0.469, and RMSE values of 0.1143, 0.1121, and 0.1242. The results indicate its better performance than linear regression.

3) SVR: The SVR model has been implemented using the Scikit-learn library and trained to predict vehicle density on historical traffic datasets divided into 80% training and 20% testing sets. GridSearchCV with 3-fold cross-validation has been applied to tune the hyperparameters as kernel='rbf', C=1.0, epsilon=0.1, gamma='scale', n_estimators=200, and random_state=42.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Tables VI, VII, and VIII show that the model obtained the values of performance evaluation metrics on the test datasets as R^2 scores of 0.7328, 0.6769, and 0.6559, MAE values of 0.0826, 0.0861, and 0.084, and RMSE values of 0.1206, 0.1205, and 0.1295. The result shows that the model can explain 73%, 67%, and 65% of the variance in vehicle density for the three datasets.

4) LSTM: The LSTM model has been implemented using the TensorFlow library on the historical traffic Datasets for vehicle density prediction. The model has been employed to tune hyperparameters, with the best configuration obtained as layers=2, units=32, dropout=0.3, dense units= 16, learning_rate=0.01, activation='tanh', optimizer='adam', validation_split=0.2, epochs=100, batch_size=32, early stopping(patience=10), verbose=1.

Across the three datasets, the model achieved test R^2 scores of 0.7584, 0.7112, and 0.6855, MAE values of 0.0728, 0.0749, and 0.0753, and RMSE values of 0.1126, 0.1139, and 0.1238. The results indicate that the LSTM model effectively captures the temporal patterns in traffic data. Also, the model is outperforming Linear Regression, Random Forest, and SVR models.

5) BiLSTM: The BiLSTM model has been implemented using the TensorFlow library to predict vehicle density based on the three different datasets. Keras tuner provides best hyperparameter set is Bidirectional layers=2, units=64, activation='tanh' and 'relu', return_sequences=True, optimizer=Adam, learning_rate=0.01, validation_split=0.2, epochs=100, batch_size=32, verbose=1.

Tables VI, VII, and VIII show that the model obtained the values of the evaluation metrics on the test datasets as R^2 scores of 0.7672, 0.7170, and 0.7050, MAE values of 0.0716, 0.0736, and 0.0764, and RMSE values of 0.1126, 0.1127, and 0.1199. According to the results, the model explains 76%, 71%, and 70% of the variance in vehicle density across the three datasets.

6) XGBoost: The XGBoost model has been implemented using the Scikit-learn library and trained to predict vehicle density on three different historical traffic datasets separately. The datasets are divided into 80% training and 20% testing sets. GridSearchCV with 3-fold cross-validation tuned the hyperparameters and found the best values as n_estimators=200, learning_rate=0.1, max_depth=3, subsample=0.9, min_child_weight=5, colsample_bytree=0.9, random_state=42.

This model achieved the performance with R^2 scores of 0.8013, 0.7497, and 0.7466, MAE values of 0.0678, 0.0719, and 0.0731, and RMSE values of 0.1023, 0.1084, and 0.1105. The results indicate that the model can explain 80%, 75%, and 74% of the variance in vehicle density for the three datasets. This model achieved the highest R^2 score and the lowest error metrics among all the trained models discussed earlier. So, it is outperforming all five other models considered in this study.

D. Vehicle Density Prediction for the Six Models

The selected six ML models were trained over our three different types of modified datasets separately to calculate predicted vehicle density (PVD). Based on the results discussed earlier, the XGBoost model outperforms all other models. Tables IX, X, and XI present the PVD values achieved by the XGBoost model on the three different datasets. Each table shows the first 20 rows from the PVD calculation table generated on the test dataset.

TABLE IX. *PVD OBTAINED IN WEEKDAY PEAK HOURS*

Index	Actual Vehicle Density	Predicted Vehicle Density (PVD)
1	0.2498	0.2419
2	0.3914	0.3538
3	0.1688	0.2308
4	0.262	0.2424
5	0.1147	0.2675
6	0.052	0.0391
7	0.3902	0.4767
8	0.5731	0.4466
9	0.2032	0.1812
10	0.204	0.2697
11	0.3703	0.5839
12	0.311	0.3054
13	0.0815	0.0785
14	0.5248	0.4677
15	0.4167	0.4995
16	0.1369	0.1555
17	0.2271	0.2119
18	0.2835	0.2745
19	0.1533	0.1672
20	0.1663	0.2545

TABLE X. *PVD OBTAINED IN WEEKDAY OFF-PEAK HOURS*

Index	Actual Vehicle Density	Predicted Vehicle Density (PVD)
1	0.5026	0.5406
2	0.136	0.2129
3	0.8157	0.5585
4	0.526	0.5664
5	0.1381	0.078
6	0.0511	0.0641
7	0.1235	0.2294
8	0.2898	0.2228
9	0.4976	0.5059
10	0.2126	0.22
11	0.1584	0.1835
12	0.071	0.1133
13	0.1105	0.0856
14	0.2183	0.219
15	0.0251	0.0755
16	0.5487	0.5943
17	0.0621	0.1141
18	0.6137	0.5843
19	0.0946	0.0845
20	0.0788	0.2427

TABLE XI. *PVD OBTAINED IN WEEKEND 24 HOURS*

Index	Actual Vehicle Density	Predicted Vehicle Density (PVD)
1	0.3652	0.5705

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

2	0.4018	0.3103
3	0.0983	0.0682
4	0.1094	0.0862
5	0.1542	0.138
6	0.3406	0.6191
7	0.0648	0.1701
8	0.9866	0.5
9	0.0901	0.1014
10	0.2451	0.2526
11	0.2081	0.1591
12	0.1462	0.0828
13	0.5383	0.4956
14	0.0966	0.0592
15	0.091	0.1064
16	0.1055	0.0816
17	0.4598	0.4736
18	0.2165	0.23
19	0.5159	0.4331
20	0.289	0.2607

VI. CONCLUSION AND FUTURE WORK

Accurate and reliable vehicle density prediction offers a significant opportunity to avoid many challenges in the transportation system. It is very important to handle the traffic systems properly in places where population density is very high. Predicting vehicle density plays an essential role in ITS applications. In this paper, after appropriate preprocessing, six models, including linear regression, random forest, XGBoost, SVR, LSTM, and BiLSTM models have been trained with the best hyperparameters. These six regression models go through a comparative analysis based on their achieved values of R^2, MAE, and RMSE metrics. Experimental results show that the XGBoost model achieved the highest predictive accuracy (R^2 score: 0.8013, 0.7497, and 0.7466), (MAE values: 0.0678, 0.0719, and 0.0731), and (RMSE values: 0.1023, 0.1084, and 0.1105) separately for weekdays peak hour, weekdays off-peak hour, and weekend 24-hour datasets. In the future, these models may be applied to a larger dataset with more features and seasonal diversity from different geographic regions, which can help validate their effectiveness, scalability, and robustness in real-world scenarios.

EFERENCES

[1] International Transport Forum, “Worldwide transport activity to double, emissions to rise further,” ITF, 2021. [Online]. Available: https://www.itf-oecd.org/worldwide-transport-activity-double-emissions-rise-further (itf-oecd.org in Bing)

[2] M. Imani, A. Beikmohammadi, and H. R. Arabnia, “Comprehensive analysis of random forest and XGBoost performance with SMOTE, ADASYN, and GNUS under varying imbalance levels,” *Technologies*, vol. 13, no. 3, p. 88, 2025. doi: 10.3390/technologies13030088

[3] S. Fatima, A. Hussain, S. Amir, S. H. Ahmed, and S. Aslam, “XGBoost and random forest algorithms: An in depth analysis,” *Pakistan Journal of Scientific Research*, vol. 3, pp. 26–31, 2023. doi: 10.57041/pjosr.v3i1.946

[4] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. doi: 10.1023/A:1010933404324

[5] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794. doi: 10.1145/2939672.2939785

[6] A. Usman, T. M. Ali, and C. Irawan, “Machine learning approaches for real-time traffic density estimation and public transport optimization,” *Engineering Proceedings*, vol. 107, no. 1, p. 117, 2025. doi: 10.3390/engproc2025107117

[7] U. Mittal and P. Chawla, “Vehicle detection and traffic density estimation using ensemble of deep learning models,” *Multimedia Tools and Applications*, vol. 82, pp. 10397–10419, 2023. doi: 10.1007/s11042-022-13659-5

[8] S. Ünlüleblebici, M. Taşyürek, and C. Öztürk, “Traffic density estimation using machine learning methods,” *Proc. Int. Conf.*, vol. 1, pp. 136–143, 2021.

[9] A. Zeroual, F. Harrou, and Y. Sun, “Predicting road traffic density using a machine learning-driven approach,” in *Proc. Int. Conf. Electrical, Computer and Energy Technologies (ICECET)*, Cape Town, South Africa, 2021, pp. 1–6. doi: 10.1109/ICECET52533.2021.9698639

[10] C. Johny and V. Dahiya, “Machine learning applications in vehicular traffic prediction and congestion control: A systematic review,” in *Proc. 6th Int. Conf. Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2022, pp. 948–954. doi: 10.1109/ICECA55336.2022.10009384

[11] P. Shetty, S. S. Poojary, S. U. Fondekar, Y. Kumar, and S. J. Bhat, “Traffic flow prediction using machine learning,” in *Proc. Int. Conf. Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, Bangalore, India, 2026, pp. 1–4. doi: 10.1109/IITCEE67948.2026.11394647

[12] H. Xin, S. Liu, and T. Li, “A multialgorithm linear regression traffic congestion prediction model based on R language New York City’s five administrative districts,” in *Proc. Int. Conf. Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, Bangkok, Thailand, 2020, pp. 284–287. doi: 10.1109/ICBASE51474.2020.00066

[13] Y. Liu and H. Wu, “Prediction of road traffic congestion based on random forest,” in *Proc. 10th Int. Symp. Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2017, pp. 361–364. doi: 10.1109/ISCID.2017.216

[14] K. Vineetha, M. T. Reddy, M. J. Ali, N. Prathima, and S. M. Rajagopal, “Real-time traffic prediction and optimization using machine learning,” in *Proc. 3rd Int. Conf. Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2025, pp. 1419–1427. doi: 10.1109/IDCIOT64235.2025.10914871

[15] A. Liaw and M. Wiener, “Classification and regression by randomForest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: https://journal.r-project.org/articles/RN-2002-022

[16] S. Vajeeda, D. BJ, N. Michael, I. Saleem, and B. P. Prathaban, “Real-time traffic prediction and route optimization framework using XatBoost algorithm,” in *Proc. 1st Int. Conf. Advancement in Futuristic Technologies (ICAFT)*, Belagavi, India, 2025, pp. 1–6. doi: 10.1109/ICAFT66710.2025.11453027

[17] R. Rathore, P. S. Chauhan, T. Tevatio, K. R. Chythanya, A. Haldorai, and M. Rajamanickam, “Machine learning and IoT-based traffic flow prediction for intelligent transportation systems,” in *Proc. Modern Electronics Devices and Intelligent Communication Systems (MEDCOM)*, Greater Noida, India, 2025, pp. 1032–1035. doi: 10.1109/MEDCOM67532.2025.11405342

[18] N. K. Putri, M. R. Wijaya, and I. Atastina, “Road density prediction based on city transport GPS time series data using a support vector machine,” in *Proc. 11th Int. Conf. Information and Communication Technology (ICoICT)*, Melaka, Malaysia, 2023, pp. 523–527. doi: 10.1109/ICoICT58202.2023.10262577

[19] H. Verma, S. Sarkar, S. Sadhukhan, A. Mishra, and M. A. Mondal, “Traffic density prediction system using deep learning,” in *Proc. 2nd Int. Conf. Data Science and Information System (ICDSIS)*, Hassan, India, 2024, pp. 1–7. doi: 10.1109/ICDSIS61070.2024.10594660

[20] M. Shaygan, C. Meese, W. Li, X. Zhao, and M. Nejad, “Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities,” *Transportation Research Part C: Emerging Technologies*, 2022. doi: 10.1016/j.trc.2022.103921

[21] T. S. Tamir, G. Xiong, Z. Li, H. Tao, Z. Shen, B. Hu, and H. M. Menkir, “Traffic congestion prediction using decision tree, logistic regression and neural networks,” *IFAC-PapersOnLine*, vol. 54, no. 4, pp. 138–143, 2021. doi: 10.1016/j.ifacol.2021.04.138

[22] A3amat02, “Urban traffic EDA and traffic density regression,” Kaggle, 2022. [Online]. Available: https://www.kaggle.com/code/a3amat02/urban-traffic-eda-and-traffic-density-regression

(This work is licensed under a Creative Commons Attribution 4.0 International License.)