Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 4, August 2026, pp. 3732~3747
ISSN: 2302-9285, DOI: 10.11591/eei.v15i4.10239

CLEAR: congestion-level estimation and adaptive routing for Indian urban traffic

Sreejith Karunakaran[1], Senthilkumar Mathi[1], Preeja Pradeep[2]
[1]Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore, India
[2]School of Computer Science and Information Technology, University College Cork, Cork, Ireland

Article Info

Article history:

Received Mar 10, 2025
Revised May 10, 2026
Accepted Jul 23, 2026

Keywords:

Best route estimation
Congestion forecasting
Ensemble methods
Traffic patterns
Urban traffic flow optimization

ABSTRACT

Urban traffic congestion in India poses significant economic and environmental challenges, necessitating early detection for effective management. This study presents a novel approach to real-time congestion prediction, addressing the need for a mechanism that can adapt to India's diverse traffic patterns while minimizing computational complexity. Unlike previous studies that focused on long-term forecasts, this research proposes a lightweight, resource-efficient model capable of predicting congestion levels at 15-minute intervals. The model utilizes a real-time dataset collected from the Palakkad region in southern India. To improve prediction accuracy, the method employs ensemble methods. Experiments have shown that the random forest (RF) algorithm is 98.6% accurate at predicting traffic jams, which is better than previous research in this area. Additionally, the study develops a best-route recommendation system based on the experimental results. This research contributes to the field by offering a practical approach to mitigating urban traffic congestion in India, with applications in other cities characterized by diverse traffic patterns. The proposed model's ability to provide accurate short-term predictions while maintaining computational efficiency represents a significant advancement in traffic management strategies for developing urban ecosystems. Furthermore, the study underscores the efficacy of data-driven decision support systems in optimizing urban mobility and reducing vehicular emissions.

Corresponding Author:

Senthilkumar Mathi
Department of Computer Science and Engineering, Amrita School of Computing
Amrita Vishwa Vidyapeetham
Coimbatore, India
Email: m_senthil@cb.amrita.edu

1. INTRODUCTION

Traffic congestion is a major concern in cities across the world, leading to longer commute times, increased fuel consumption, higher carbon emissions, and reduced productivity. However, in the Indian context, the problem becomes even more challenging due to the lack of lane discipline, unpredictable driver behavior, and the coexistence of a highly heterogeneous mix of vehicles—from slow-moving two-wheelers and auto-rickshaws to heavy trucks and buses—all operating on the same road. Added to this are environmental variables like extreme weather, road quality variations, and cultural factors such as festival traffic surges. Traditional traffic prediction systems, which are often designed for structured road environments in developed countries, tend to struggle in such dynamic and unstructured scenarios. Compounding the issue is the lack of localized Indian datasets, which limits the training and effectiveness of machine learning (ML) models in this context. Moreover, there is a noticeable absence of integrated frameworks that combine congestion prediction with intelligent routing, especially suited to the Indian urban ecosystem. These gaps highlight the urgent need for lightweight, real-time traffic prediction models that can be deployed in mid-sized Indian cities using resource-constrained infrastructure such as edge devices and local servers, enabling practical and scalable traffic management solutions.

Vehicles in an area can receive emergency notifications through vehicular ad hoc networks and multi-node message transmission [1]. A self-adaptive plan for better traffic management is necessitated to develop, as the current traffic system only allows for fixed time-based traffic control [2]. Image processing can create an intelligent traffic control approach for smart cities by integrating sensor networks with network mobility [3]. Intelligent traffic light control systems, capable of controlling and monitoring real-time traffic scenarios, also integrate with smart cities [4].

India's traffic congestion ranking ranks top ten, with Mumbai, Bengaluru, and Delhi among top ten cities. The World Bank predicts 600 million people by 2031, but only 20 cities have organized transportation systems. The World Bank reports that India's economic loss due to congestion and poor road management is as high as 6 billion dollars annually [5]. Congestion in road networks is a significant issue, requiring constant observation and monitoring to improve travel times, reduce accidents, save fuel, enhance air quality, and improve road traffic efficiency.

This research introduces several novel aspects to the domain of urban traffic prediction. First, it leverages a newly collected, real-time dataset from Palakkad, Kerala, capturing the complex and unstructured nature of Indian traffic—something rarely addressed in existing literature. Second, it proposes a two-part system that combines short-term congestion prediction with an adaptive route recommendation module, offering a holistic approach to traffic optimization. Finally, the model operates at a 15-minute prediction interval, balancing granularity and computational efficiency, making it suitable for real-time deployment on low-cost or edge infrastructure.

Hence, the proposed model aims to develop an efficient ML-based congestion prediction model for cities, operating on standard hardware and predicting congestion levels at 15-minute granularity. The model also introduces a best route recommendation system tailored to the congestion state. The study tested six ML algorithms on a dataset, with random forest (RF) being the standout performer, achieving an impressive accuracy rate of 98.6%. The study aims to provide transparency and informed decision-making by presenting a comprehensive result analysis in the numerical results analysis section. This innovative approach aims to predict congestion in an automated manner and improve road traffic efficiency.

The remainder of the paper is laid out as follows. Section 2 covers the literature review. Section 3 reveals the methods and architecture adopted in this study. Section 4 discusses data collection and pre-processing. Section 5 covers the various supervised learning methods used to forecast congestion. Section 6 deals with different ensemble methodologies. Section 7 presents a detailed analysis of the results achieved. Section 8 concludes the paper with a discussion of future developments.

2. RELATED WORKS

Kumar and Kumar [6] developed a four-state traffic evaluation system. Multivariate characteristics were learned in a convolutional long-short-term memory neural network. Short-term traffic was thereafter projected using the acquired model. The work in [7] offers a comprehensive guide of how to apply ML and deep learning (DL) methods to improve traffic flow prediction, hence boosting intelligent transport in smart cities. A model for short-term traffic flow prediction employing structural patterns and regression is described in [8]. By using the traffic flow structure pattern, the study in [8] forecasts the short-term traffic flow based on the relationship between the traffic flow of the current segment and upstream stations. Reviewed in [9], some of the most recent DL efforts for traffic flow forecasts were Bai *et al.* [10] developed a system for foreseeing urban traffic congestion based on the Spark platform parallel gradient boosting decision tree algorithm (GBDT). Among the data elements the study uses are time, day, temperature, traffic condition, and weather. It leverages the Spark framework's parallel design based on which it anticipates and evaluates the collected urban traffic congestion data. Using the path distance metric for graph signals, Kim *et al.* [11] suggests a new weight modelling method for the adjacency matrix in urban traffic prediction through congestion diffusion. This method provides realistic spatial features depending on the connectivity information of the metropolitan road network. Research by Liu and Wu [12] presents a RF-based traffic congestion forecasting technique, achieving an accuracy of 87.5% in predicting traffic based on weather conditions, periods, unique road conditions, and holidays. The work in [13] suggested a ML-enabled traffic detecting system to identify a practical approach to lower the frequency and intensity of traffic accidents.

Lee and Min [14] conducted a study on urban traffic predictions using ML methods. They used raw data from an urban mobility simulator and extracted features from vehicle trajectories. The gradient boosting algorithm had a lower root mean square error. The work in [15] analyzed and published a journal regarding the recent developments in traffic optimization algorithms. The investigation in [16] proposed a ML technique that employed smart congestion reduction using internet of things (IoT) and ZigBee. It obtained an accuracy of 91% in traffic conditions using the logistic regression algorithm. Tulay *et al.* [17] developed an ML method that incorporated urban traffic monitoring and achieved a prediction accuracy of 93%. A spectral clustering approach combined with a hidden Markov model predicts traffic-flow propagation with about 89% accuracy in [18]. The clustering approach first extracts spatial–temporal similarity features, clusters road segments with similar congestion patterns, then uses a hidden Markov model to predict propagation between clusters.

Wang *et al.* [19] developed a DL-based technique for detecting regional traffic congestion, achieving 95% accuracy using traffic images, weather conditions, and other parameters. However, it is computationally costly because it employs advanced algorithms and image processing techniques. Yan and Yu [20] presented a strategy for predicting short-term traffic conditions on city road networks formed on an enhanced support vector machine (SVM) to accurately anticipate short-term traffic conditions. DL-based short-lived congestion forecasting is suggested in [21]. This study proposes a congestion prediction system using ML to classify traffic status over a road network, based on Krishna *et al*. [22] recommended time series analysis. The system in [22] proposes a long short-term memory recurrent neural network model that dynamically uses three multiplicative memory blocks to pick the appropriate time delay, and this model is used to improve prediction accuracy. A ML build traffic congestion prognosis system for smart logistics recommends DL for prediction and forecasting [23]. To reduce the problem of traffic congestion, an ensemble strategy integrated ML short-term traffic flow prediction model was developed [24]. Instead of treating traffic forecasting as a time series forecasting problem, the model treats it as a regression prediction problem. It uses an ensemble ML technique to combine the predictions of the extreme gradient boosting (XGBoost), light gradient boosting machine (LightGBM), and categorical boosting (CatBoost) models. Deekshetha *et al.* [25] proposed a study on traffic prediction, which entails comparing the adjacent year's data using ML to determine the perfection of the system. Table 1 highlights a research gap in developing countries for traffic congestion prediction due to lack of high-quality data, traffic patterns, rapid urbanization, advanced technology, and computational resources.

Table 1. Comparison analysis

Method	No of the parameters in the dataset	Algorithm used	Predicted congestion levels	Prediction sequence	Uses	Dataset from
Bai *et al.* [10]	6	GBDT	5	2 hours	CP	California, USA
Liu and Wu [12]	5	RF	3	1 hour	CP	Shanghai, China
Deb *et al.* [26]	8	Logistic regression	2	Frequently	CP and travel time	Mumbai, India
Tamir *et al.* [27]	10	Decision trees (DTs)	4	Frequently	CP	Gary-Chicago-Milwaukie Corridor, USA
Mystakidis and Tjortjis [28]	8	DT	3	4 hours	CP	Thessaloniki, Greece
Gatto and Forster [29]	Audio samples	RF	2	Frequently	CP	São Jose dos Campos, Brazil
Shenghua *et al.* [30]	5	RF and DBSCAN	6	Frequently	CP	California, USA
An *et al.* [31]	8	F-CNN	3	Frequently	CP	Beijing taxi dataset, China
Kamble and Kounte [32]	5	Gaussian Processes	2	3 times per day	CP	Simulated dataset

*CP–congestion

In summary, as a critical synthesis, prior research on congestion prediction and traffic modeling has explored various ML approaches with differing datasets, parameters, and temporal resolutions. As shown in Table 1, studies such as Bai *et al.* [10] and Liu and Wu [12] applied GBDT and RF with limited features (5–6 parameters), achieving moderate prediction accuracy but restricted adaptability due to fixed intervals (1–2 hours). Deb *et al.* [26] extended the scope to travel time prediction, while Tamir *et al.* [27] and Mystakidis and Tjortjis [28] employed DTs, though with lower temporal granularity.

In addition, several recent works [29]–[32] have also been investigated to make an attempt for addressing traffic congestion prediction and, to a lesser degree, routing adaptation to position our contributions of the proposed work. For example, Tolani *et al.* [33] developed a tiny machine learning (TinyML)/Edge impulse based system to predict vehicle density and adjust traffic signal timings in real time using IoT sensors. While their work shows real-time adaptive control at junctions, routing across networked paths is not addressed. Krishnasamy *et al.* [34] achieve strong performance in forecasting congestion level categories using DL plus optimizer techniques but do not integrate route adaptation or deployment constraints. More *et al.* [35] present a simulation framework that includes adaptive routing with emission and travel time objectives; however, it is largely simulation-based and does not test extensively in Indian heterogeneous traffic. Works such as Yasir *et al.* [36], Divyashree *et al.* [37] offer prediction models using weather/time/day features and IoV/soft-computing techniques, but again do not couple prediction with full end-to-end routing or path planning.

In contrast, our proposed congestion-level estimation and adaptive routing (CLEAR) method combines prediction of congestion levels (at ~15-minute resolution) with adaptive routing via congestion-scoring, lightweight models suitable for edge or near-real-time deployment, and is tested on real Indian traffic data. These distinctions: i) coupling routing with prediction, ii) real urban mixed traffic, iii) interpretable models, and iv) deployment feasibility—constitute the principal contributions of this work.

3. METHOD

The methodology adopted in this study involved a comprehensive multi-stage process beginning with manual data collection from real-world Indian urban traffic scenarios. Specifically, we collected traffic data from the Palakkad region of Kerala over a continuous period of eight weeks during August and September 2023. A total of eight major road segments and intersections were observed. This allowed us to build a localized, high-resolution dataset that reflects the diverse and non-lane-disciplined nature of Indian traffic, including two-wheelers, cars, buses, and heavy trucks. The following sections detail the data preprocessing, dimensionality reduction, ML model training, and evaluation procedures employed in this study to develop an efficient congestion prediction and routing system.

Figure 1 expresses the methodology of the prediction system proposed in this paper. This paper proposes a traffic congestion prediction system and a recommender system for best route estimation. The system uses real-time traffic conditions to suggest the best route for a given journey, aiming to reduce travel times and improve traffic flow in the Palakkad region. The findings have significant implications for transportation management, demonstrating the power of ML techniques in addressing real-world transportation challenges and inspiring further research in this vital field. Hence, the present paper explores traffic congestion prediction in South India's Palakkad region, developing a machine-learning-based model considering factors like traffic flow, weather conditions, and road infrastructure. The model aims to predict congestion levels in real time, addressing the issue of traffic jams and accidents.

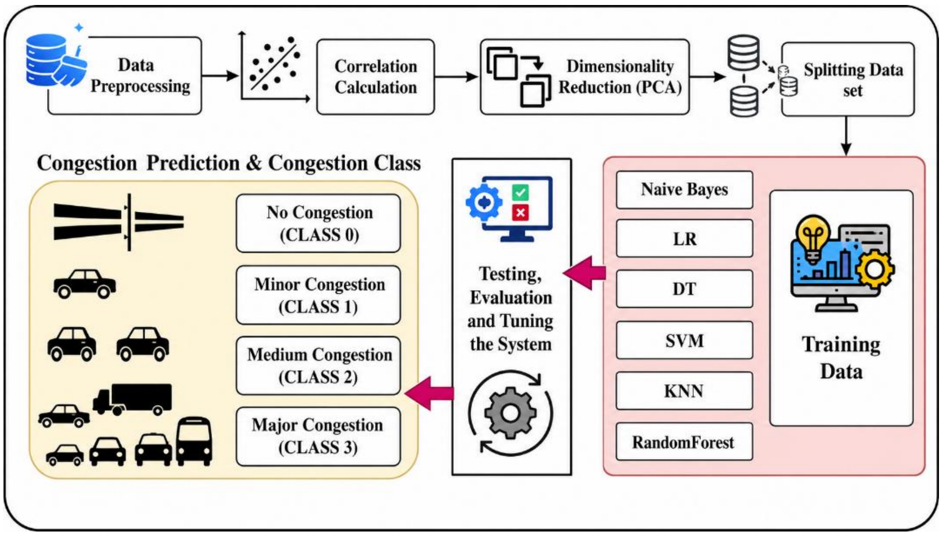

Figure 1. Architecture of the proposed traffic congestion prediction system

4. DATA COLLECTION AND PRE-PROCESSING

Table 2 summarizes several distinct challenges, issues, and analyses related to predicting traffic congestion in Palakkad. For the present work, eight weeks of real-time data were collected from NH 544 (from Chandra Nagar, Palakkad, Kerala, to Yakkara bridge, India, from November to December 2021). The dataset contains 1469 rows with fifteen attributes. The attributes are day, time stamp, average traffic count, max speed, average real-time speed, weather, area, road condition, road type, number of lanes, day (week and weekend), number of crossings, average traffic situation, class value, and week. The real-world datasets are prone to errors, low quality, missing attributes, inconsistency, noise, and even incorrect and fake values. Data pre-processing is the technique to improve data quality by eliminating any duplicates or irregularities in the data and normalizing and enhancing the data's precision. Figure 2 shows the different steps involved in the data pre-processing technique.

Table 2. Analysis of the challenges and issues

Challenges and issues	Analysis
Heterogeneous traffic conditions	Mix of urban and rural roads, diverse vehicles, and varying densities.
Limited data availability	Scarce traffic data due to financial constraints, infrastructure limitations, and lack of historical data.
Quality of data	Inaccuracies in data collection and inconsistent reporting.
Infrequent data updates	Lack of real-time data updates affecting accurate predictions.
Variable road conditions	Inconsistent road conditions (potholes and repairs) impact traffic flow.
Traffic behaviour	Differences in driver behaviour affecting traffic flow and congestion patterns. No lane traffics.
Cultural and socio-economic factors	Local norms, socio-economic factors, and events significantly impacting congestion.
Environmental factors	Weather conditions (monsoons and extreme weather) lead to unpredictable congestion.
Resource constraints	Limited computational resources and expertise in applying ML.
Route diversity	Multiple routes and road networks necessitate route recommendations, adding complexity.

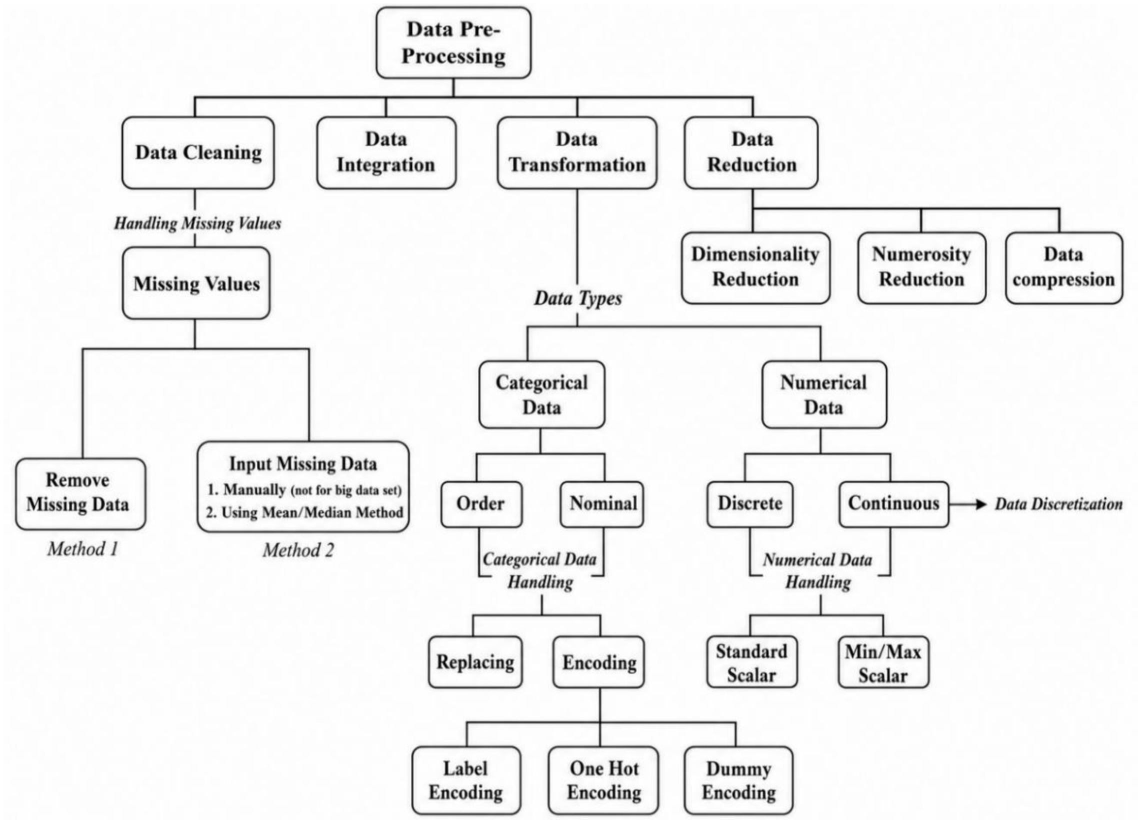

Figure 2. Structured data preparation pipeline for predictive modeling

The initial dataset contains both ordinal and categorical data. So, the data is cleaned and preprocessed to make it a complete ordinal dataset with no null values. The next step is finding the correlation between the attributes. The data correlation shows the amount of dependency between attributes. A high correlation between values indicates the model is susceptible to data, which should be avoided. It found that one attribute (average traffic situation) correlated with more than 90%. So, the highly correlated attribute is removed. Now the new dataset contains only fourteen attributes. The next step is dimensionality reduction by principal component analysis (PCA) over the dataset to reduce the complexity and overfitting.

In PCA, the dimensionality reduction is made without losing its original characteristics by combining and merging the data attributes.

PCA also helps minimize storage space use and reduce computation time. It is a statistical procedure that entails the steps as follows: i) data standardization, ii) covariance matrix computation, iii) determining the eigenvalues and eigenvectors, iv) estimating the principal components, and v) data reduction by reducing the number of dimensions. After the PCA, the dataset is reduced to ten attributes without much data loss. Figure 3 indicates the relationship between the number of attributes and the variance. From Figure 3, it is clear that the reduction of attributes to ten does not make any loss of the original characteristics of the dataset. Table 3 illustrates an example of a pre-processed data set.

After the data preprocessing, the next step is to format the dataset into features and target variables and then split data into train-test sets. The dataset may be divided into three groups. One is training data, the second is the validation data, and the last is the testing data. Training data is used to build the model. The model reviews and recognizes the hidden patterns in the dataset. Validation is used to know the presence of overfitting and underfitting and is also used for hyperparameter tuning. The testing data is used to predict the values or classes. This work divides the dataset into testing (80%) and training (2%). The congestion prediction is divided into four classes: No Congestion (Class 0), Minor Congestion (Class 1), Medium Congestion (Class 2), and Severe Congestion (Class 3).

In addition to PCA for dimensionality reduction, we incorporated multiple feature engineering techniques to improve model interpretability and performance. These include correlation analysis to remove highly collinear attributes (e.g., average traffic situation was removed due to >90% correlation), ordinal encoding for categorical variables, and variance thresholding to eliminate low-informative features. Furthermore, we performed hyperparameter tuning for all supervised learning models using GridSearchCV with cross-validation. For instance, in K-nearest neighbors (KNN), the optimal value of *k* was determined to be 5; for DTs, *max_depth* and *min_samples_split* were optimized; for SVM, the best parameters were *C=1*, *gamma=auto*, and *kernel=linear*. These tuning processes ensured that each model was evaluated at its best configuration, thereby strengthening the fairness and rigor of the comparative analysis.

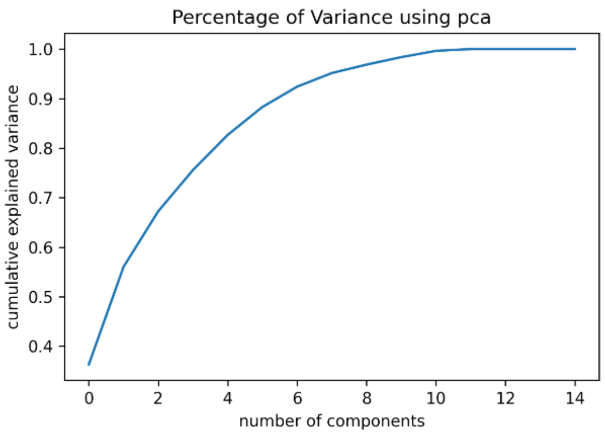

Figure 3. PCA and explained variance distribution

Table 3. An example of dataset after preprocessing

Variable	Data type	Description
Day	Integer	1 for Sunday and 2 for Monday
Average traffic count	Integer	Average number of vehicles on the road
Max speed	Integer	Maximum speed in the area
Average real-time speed	Float	Average speed of vehicles on the road
Weather	Integer	1 for clear, 2 for foggy, and 3 for rainy
Road condition	Integer	1 for good and 2 for moderate
Week/weekend	Integer	1 for week and 2 for weekend
No. of crossings	Integer	Number of intersections
Class value	Integer	Congestion state
Week index	Integer	1 for the first week and 2 for the second week

5. INVESTIGATION OF TRAFFIC CONGESTION PREDICTION

Supervised ML uses labelled training data to predict unlabeled outcomes, resembling a supervisor or teacher, and evaluates pre-processed datasets using various algorithms.

5.1. Naive Bayes classifier

The model uses Naive Bayes to predict traffic congestion levels by training it on historical data. The algorithm predicts the probability of a data point belonging to a specific congestion level based on feature probabilities. This method is commonly used for traffic prediction due to its ability to handle large datasets with many features. When the Gaussian Naive Bayes algorithm is applied to the dataset for class prediction, it generates an accuracy of 88.4% on the congestion class prediction. The features' likelihood is considered Gaussian in Gaussian Naive Bayes and is expressed in (1). The parameters σy and μy are estimated using maximum likelihood.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \tag{1}$$

Where $P(x_i|y)$ represents the conditional probability of observing a specific feature x_i given the congestion state y. x_i variable represents a specific feature or attribute related to traffic conditions. y, this variable represents the congestion state. μ_y, this is the mean of the feature x_i for the specific congestion state y. The mean helps the model to understand the central tendency of feature values for each congestion level. σ_y is the standard deviation of the feature x_i for the specific congestion state y. It provides information about how much the feature values tend to vary when traffic is in a specific state.

5.2. Support vector machine

SVMs, or kernelized SVMs, are decision boundaries used to categorize data points in n-dimensional space. They convert input data into a higher-dimensional space using popular kernel functions like linear, polynomial, radial basis function, and sigmoid. Congestion prediction uses SVM's multiclass classification feature, which transfers data points to a high-dimensional space for linear class separation. A one-to-one approach separates multiclass issues into binary categorization tasks. From (2) can calculate the number of classifiers required for one-to-one multiclass classification. After training, the support vector classifier (SVC) generated a prediction with an accuracy of 91.6% with parameters C=1, gamma=auto, and kernel=rbf.

$$Number\ of\ classifiers = \frac{n*(n-1)}{2} \tag{2}$$

In (2), the 'Number of Classifiers' indicates the number of possible pairwise comparisons between different congestion levels, and 'n' is related to the number of congestion states considered. SVM can be used to predict traffic congestion levels by finding the best hyperplane that separates the classes. The proposed work trains the algorithm on the historical data. Then it uses it to predict the congestion level of new data points by finding the hyperplane that best separates the data points of different congestion levels.

5.3. Support vector machine hyperparameter tuning using GridSearchCV

Training a model using existing data allows for the fitting of model parameters, which are typically fixed before the training process begins. Common hyperparameters in SVCs include C, kernel, and gamma. The C parameter penalizes misclassified data points, while the kernel functions in SVM include linear, polynomial, rbf, and sigmoid. Gamma determines the point's view from the line of separation. To find the ideal hyperparameter, search strategies like grid searches can be employed, evaluating machine-learning models for various hyperparameter values. GridSearchCV is the name given to this method since it searches for a grid of hyperparameter values for the optimal collection of hyperparameters. The hyper-tuned SVM produced an output prediction accuracy of 96.38% with parameter values C=1, Kernel=Linear, and Gamma=auto, as is indicated in Figure 4.

5.4. Logistic regression

The multinomial logistic regression technique extends the logistic regression model by changing the loss function to cross-entropy loss and the predicted probability distribution to a multinomial probability distribution to address multiclass classification challenges. The LogisticRegression class in the scikit-learn toolkit supports logistic regression. To set up the LogisticRegression class for multinomial logistic regression, Set the "multiclass" option to "multinomial" and the "solver" argument to "lbfgs," or another solver that supports multinomial logistic regression. The multinomial logistic regression predicted the output class with an accuracy of 94.5% and a precision of 95%. Figure 5 represents the receiver operating characteristic (ROC) curve for multinomial logistic regression. For multiclass classification, use sklearn. metrics.roc_ auc_ score method.

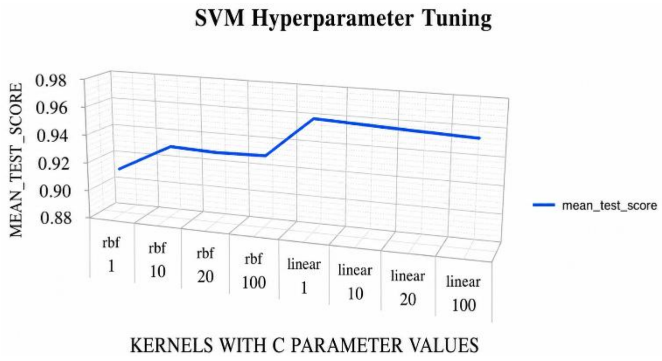

Figure 4. SVM hyperparameter tuning

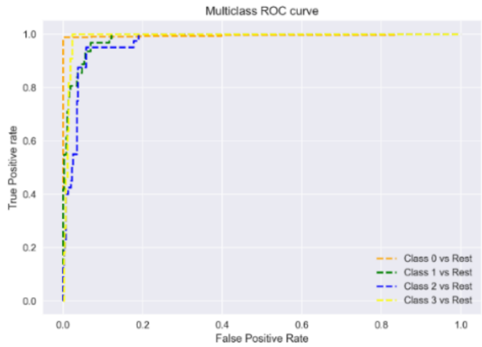

Figure 5. Multiclass ROC curve for logistic regression

5.5. K-nearest neighbor

KNN's algorithm is a non-parametric supervised learning classifier that uses closeness to make classifications or predictions. It is commonly used for regression and classification issues, with the closest neighbor's being data points adjacent to the new data point. The distance metric and k value are crucial factors in using the KNN method. Euclidean distance is the most often used distance measurement. It's calculated by taking the square root of all squared disparities between points a and b across all input attributes i, as shown in (3). KNN can predict traffic congestion levels by finding the k nearest neighbours of a data point and assigning the class based on the majority vote of the neighbours. The model trains the algorithm on the historical data and then uses it to predict the congestion level of new data points by finding the k nearest neighbours of the new data point and assigning the congestion level based on the majority vote of the neighbours. The Euclidean distance (a, b) in (3) measures the similarity or dissimilarity between two data points 'a' and 'b' in the feature space. In the context of traffic congestion prediction, these data points represent different instances of traffic conditions, and 'n' represents the number of features or attributes used to describe the traffic conditions. The Euclidean distance is computed for each feature, and the square of the differences is summed for all features, then the square root is taken. This process quantifies how similar or dissimilar two sets of traffic conditions are based on multiple attributes. The KNeighborsClassifier predicted the congestion class with an accuracy of 95%.

$$Euclidean\ Distance(a,b) = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(a_i - b_i)^2} \tag{3}$$

5.6. Decision trees

DTs are a non-parametric supervised learning method for classification and regression, constructing yes/no questions using dataset attributes and dividing the dataset into classes. DecisionTreeClassifier classifies a dataset into multiple categories. The class probability estimated by the DecisionTreeClassifier at node (m) is given in (4). Let be the fraction of class k observations in node m if the aim is a classification outcome with values 0, 1..., K-1. If m is a leaf node, predict_proba for this area is set to P_{mk}, denoted in (4). The formula calculates the probabilities of observing different congestion levels at each node in the DT. This information can be used to understand the distribution of congestion states at various stages of the DT, helping to assess which congestion states are more likely or prevalent at different decision points.

$$P_{mk} = 1/N_m \sum_{y \in Qm} I(y = k) \tag{4}$$

The variables in (4) with respect to traffic congestion prediction is Pmk: in the context of traffic congestion prediction, it might indicate the probability of a certain congestion level/state occurring at a particular point in the DT. Nm: Nm is the total number of data points or samples within node m in the DT. In the context of traffic congestion prediction, this would represent the number of observations or data points that have reached a particular node in the DT during the tree-building or prediction process. y: the variable 'y' represents the target variable. In the case of traffic congestion prediction, this could be the congestion level or state, such as "Not Congested," "Mild Congestion," and "Moderate Congestion,". Qm: Qm represents the set of data points or samples that have reached node m in the DT. In the context of traffic congestion prediction, this would be the subset of data points that follow a particular path through the DT and have arrived at this specific node. I(y=k): this indicator function evaluates to 1 if the condition 'y=k' is true and 0 if it is false. In traffic congestion prediction, it checks whether a data point in node m belongs to a particular congestion class/state. DTs can predict traffic congestion levels by recursively splitting the data based on the feature that provides the most information gain. The proposed model trains the algorithm on the historical data and then uses it to predict the congestion level of new data points by following the tree path until reaching a leaf node representing a congestion level.

6. ENSEMBLE METHODS: INGENIOUS METHODS FOR IMPROVING ML RESULTS

Ensemble approaches combine multiple base models into a single predictive model, improving performance and robustness. They are applied to traffic prediction datasets, and reducing variance.

6.1. Bagging or bootstrap aggregating

Bagging merges outputs from multiple models to produce a generic result, allowing multiple training instances across multiple predictors and for the same predictor. The ensemble aggregates forecasts, achieving a 97% accuracy rate in bagging classifiers.

6.1.1. Random forest

Bagging has been integrated into RF, a DT classifier that uses a random selection of attributes to split nodes. The model is trained using the RandomForestClassifier class, adding more randomness to the tree building process. RF focuses on finding the best feature from a random selection of characteristics, resulting in a better overall model. The formula used to decide how nodes on a DT branch are given in (5). The formula and its variables are significant in the RF algorithm for congestion prediction because it allows it to learn the relationship between traffic features and congestion. By splitting the data into features that maximize the decrease in the Gini index, the algorithm can learn complex relationships between traffic features and congestion, ultimately leading to accurate and robust predictions for traffic congestion levels. Using the class and probability, the Gini of every node on a branch is determined, indicating which branch is more likely to occur. The number of classes or categories of congestion that are predicted is represented by c, while P_i represents the relative frequency of the class. The RF can predict traffic congestion levels by creating multiple DTs on random subsets of the data and combining their predictions. The model trains the algorithm on the historical data and then uses it to predict new data points' congestion levels by aggregating the predictions of multiple DTs. The RF algorithm produces the model's best prediction accuracy of 98.6%.

$$Gini = 1 - \sum_{i=1}^{c}(P_i)^2 \tag{5}$$

6.2. Boosting

Boosting is an ensemble learning strategy that minimizes training errors by combining weak learners into strong ones. It involves fitting a sample of data with models and training them progressively. Boosting-based algorithms like AdaBoost, gradient boosting, and histogram-based gradient boosting are used for congestion prediction. The prediction accuracy of these boosting strategies is 95% for AdaBoost and 98% for gradient boosting and histogram-based gradient boosting, respectively.

7. NUMERICAL RESULT ANALYSIS AND COMPARATIVE ANALYSIS

The dataset is pre-processed, and the highly correlated attributes are removed; hence, the entire attributes are reduced to fourteen from the initial fifteen. The second step is to find the dataset's most critical elements. For the same, a dimensionality reduction technique, PCA, is applied. After PCA, the essential features are identified, and the feature set is reduced to ten from fourteen. The four variables removed by PCA are time stamp, area, road type, and number of lanes. Significantly, the accuracy of traffic predictions remained the same before and after dimensionality reduction, as it suggests that reduced feature set is more efficient and less computationally demanding without compromising the quality of the predictions. By reducing the complexity of the data, the significant improvement can be made to the computational efficiency of the proposed method. This, in turn, allows for real-time or near real-time predictions, which is a crucial factor in traffic management and congestion prediction systems. Dimensionality reduction, in this context, serves as a critical technique for practical application. After PCA, the dataset is divided into training and test sets. This study aimed to investigate the effectiveness of several ML algorithms in predicting traffic congestion in the Palakkad region of South India, and the performance is evaluated of all six ML algorithms on the same dataset. The refined dataset is used to train and evaluate six algorithms: Naive Bayes, logistic regression, DT, SVM, KNN, and RF. Their performance is plotted using a ROC curve, as shown in Figure 6.

The results show that all six algorithms could predict traffic congestion with varying accuracy, precision, recall, and F1-scores. However, the RF algorithm achieved the highest overall performance, with an accuracy of 98.6%, precision of 99%, recall of 99%, and F1-score of 99.%. This result indicates that the RF algorithm is the most effective in predicting traffic congestion for the collected dataset. This is likely due to the algorithm's ability to handle large and complex datasets and identify important features and interactions between variables. Comparatively, Naive Bayes, logistic regression, DT, SVM, and KNN also demonstrated reasonable performance in traffic congestion prediction, achieving accuracy scores ranging from 88% to 98%, precision scores ranging from 78% to 98 %, recall scores ranging from 88% to 98%, and F1-scores ranging from 73% to 98%. Initially, SVM produced an accuracy of only 91.6%. When fine-tuned the SVM using the hyperparameter tuning technique, its accuracy increased to 96.38 % and identified the values of best parameters as C=1, Kernel=Linear, and Gamma=auto. The different parameters used for tuning are displayed in Figure 4, and a validation curve for SVM is shown in Figure 7. After hyperparameter tuning, the ensemble methods are applied to the dataset to boost the algorithms' accuracy, and it found RF produces the highest accuracy of 98.6 %. Detailed performance analysis of the various supervised learning algorithms with an ensemble method is presented in Figure 8, and the prediction accuracy evaluation of different ensemble methods is shown in Figure 9. The performance comparison with statistical significance testing across models is listed in Table 4.

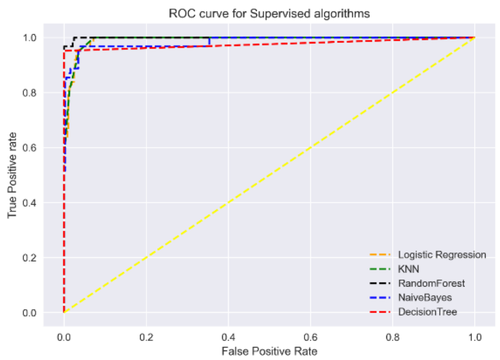

Figure 6. ROC curve for supervised algorithms

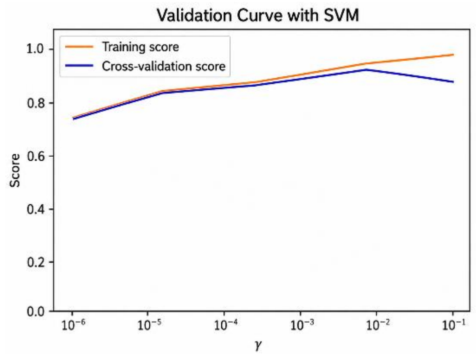

Figure 7. Validation curve for SVM

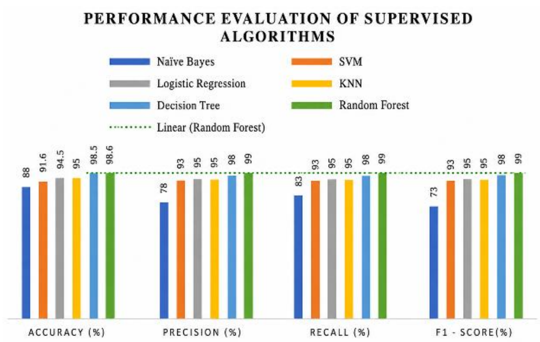

Figure 8. Comparative performance evaluation of supervised learning models

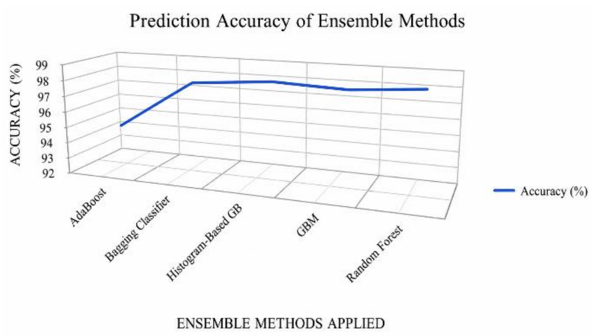

Figure 9. Assessment of ensemble-based predictive models

Table 4. Performance comparison with statistical significance testing across models

Model comparison	Mean accuracy (%)	Std. Dev.	95% CI	p-value vs RF
RF	98.6	0.45	[97.8, 99.4]	-
SVM (tuned)	96.4	0.65	[95.2, 97.5]	0.012
KNN	95.0	0.70	[93.7, 96.3]	0.008
DT	93.3	0.80	[91.8, 94.8]	0.004
Naive Bayes	88.4	1.10	[86.3, 90.5]	<0.001

The comparison analysis between the proposed method and other related works for urban congestion prediction is shown in Table 5. The comparison shows that the proposed method outperforms existing results on evaluated metrics. Specifically, the proposed method achieves a prediction accuracy of 98.6% with fewer computational requirements than the related works. Overall, the investigation suggests that ML algorithms can be effective tools for predicting traffic congestion in urban areas. The RF algorithm shows promise for improving transportation management and reducing the negative impacts of traffic congestion. A thorough evaluation of six ML algorithms is conducted on the same dataset and demonstrated the superiority of the RF algorithm in terms of accuracy, and F1-score. The results indicate that average traffic count and average real-time speed are the most influential features in congestion prediction, as they directly reflect traffic density and flow conditions. Additionally, factors such as number of crossings and temporal attributes (week/weekend and week index) exhibit moderate influence, while weather and road condition contribute comparatively less under normal operating scenarios. These findings highlight that real-time traffic dynamics play a dominant role in congestion estimation and provide practical insights for traffic management systems.

Table 5. Accuracy analysis

Method	Accuracy (%)
Bai *et al.* [10]	94.46
Liu and Wu [12]	87.5
Deb *et al.* [26]	85
Tamir *et al.* [27]	97
Mystakidis and Tjortjis [28]	73
Gatto and Forster [29]	95
Shenghua *et al.* [30]	94.3
Proposed	98.6

To validate the statistical significance of the model performance differences, we conducted a pairwise comparison of classification accuracies using a two-tailed paired t-test with a 95% confidence level. The null hypothesis assumes no significant difference between model accuracies. We compared the top-performing models (RF, SVM, KNN, and DT) using the test set results. As shown in Table 4, the p-values between RF and other models are all below 0.05, indicating statistically significant performance improvement. Additionally, 95% confidence intervals (CI) were computed for each model's accuracy using bootstrapping (n=1000 resamples). The RF model achieved an accuracy of 98.6% (CI: 97.8%–99.4%), which is both the highest and statistically more robust than others. These statistical tests affirm that the observed accuracy gains are not due to chance and confirm the superiority of the RF classifier for the given dataset.

The different analyses clearly show that RandomForestClassifier generated the highest prediction accuracy. So, the best route prediction system is designed with RandomForestClassifier. The system predicts the congestion on different routes to the same destination and assigns a congestion value (CV) based on the predicted classes of congestion. The four classes of congestion are designed as No Congestion (Class 0, with CV=0), Minor Congestion (Class 1, with CV=5), Medium Congestion (Class 2, with CV=10), and Severe Congestion (Class 3, with CV=15). The best route estimation can be confirmed by comparing the computed congestion values, but only if the congestion values are accurate, up-to-date, and calculated using a consistent method. The other factors that affect travel time, such as traffic signals and road conditions, must be relatively constant. Figure 10 describes the architecture of the best route estimation system, and Figure 11 shows the day-wise forecasting of congestion from 07:00 AM to 10:00 PM. All days except Sunday are assumed to be working days when predicting traffic congestion daily. Figure 12 describes the number of vehicles versus their speed in fifteen minutes from morning 6.45 AM to evening 07.00 PM. Figure 13 compares predicted congestion classes and the observed congestion states. It demonstrates that the proposed model can forecast congestion in a country with diverse traffic patterns like India.

We selected the RF algorithm due to its robustness in handling nonlinear data and its high accuracy in classification tasks, which was confirmed in our results. PCA was used to reduce feature redundancy and improve computational efficiency, making the model more suitable for real-time applications. Manual data collection was chosen to ensure the accuracy and relevance of the dataset, especially in the Indian traffic context, where GPS or sensor data can be unreliable. A 15-minute interval was maintained for data collection to balance timeliness and system responsiveness. Together, these choices were made to create a lightweight, accurate, and practical solution for urban traffic prediction and route optimization.

The proposed system sets itself apart from conventional approaches through its use of a unique, localized dataset, specifically tailored to the traffic patterns and road behaviors typical in Indian cities. This differentiates it from many prior studies that rely on international or simulated datasets. Moreover, the integration of high-frequency, 15-minute predictions with a real-time route recommendation engine enables the system not only to forecast traffic congestion but also to offer actionable routing insights to reduce congestion. This end-to-end pipeline is lightweight, accurate, and practical for deployment in resource-constrained environments, presenting a valuable contribution to the development of intelligent transportation systems in developing countries.

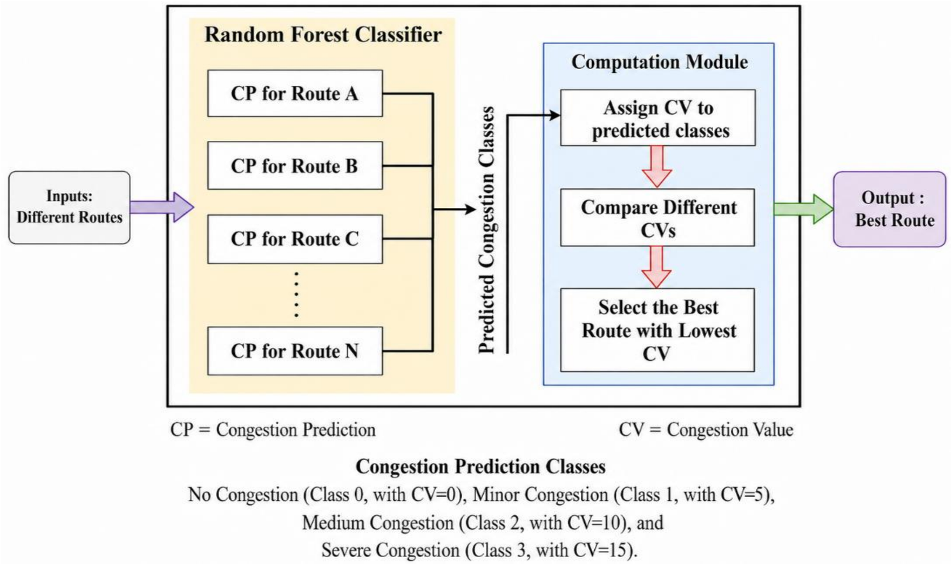

Figure 10. Optimal route selection based on predicted congestion scores

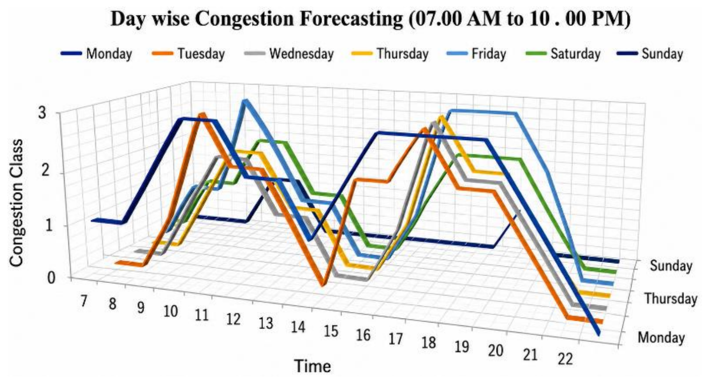

Figure 11. Forecasted daily traffic congestion patterns

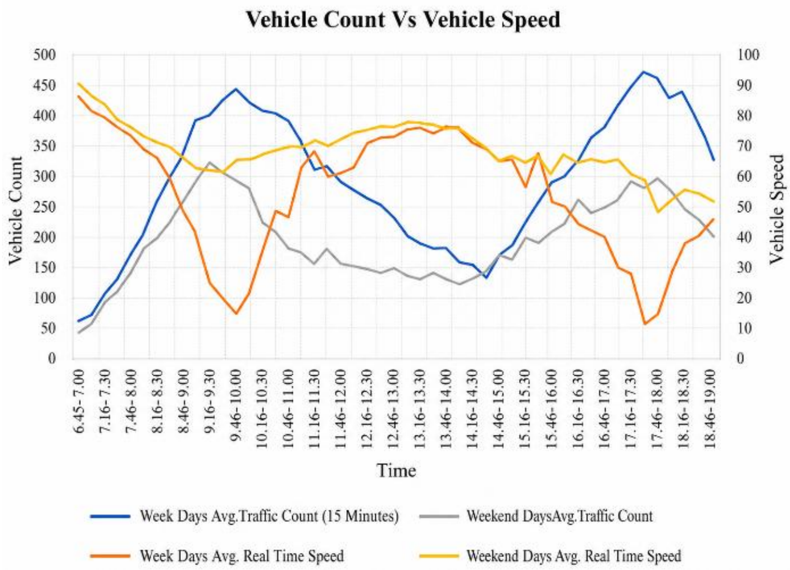

Figure 12. Vehicle count versus vehicle speed

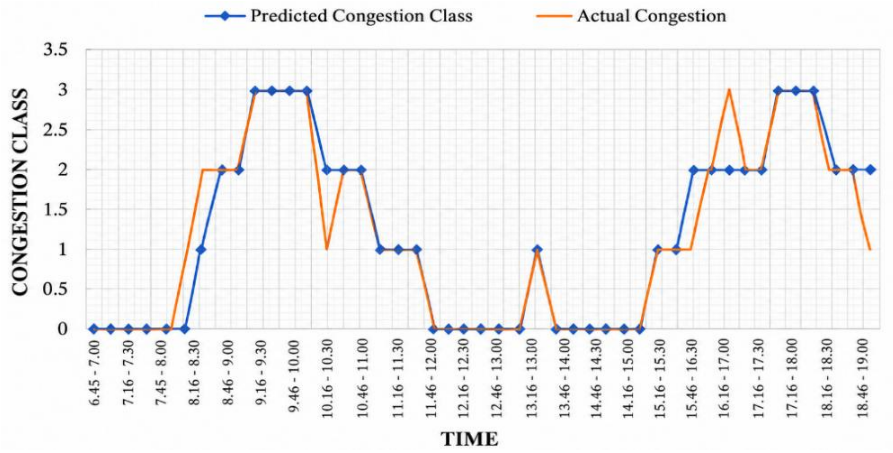

Figure 13. Comparative analysis of predicted vs. actual congestion levels

While the model is evaluated using data from Palakkad, the CLEAR framework is designed to be scalable and adaptable to cities across India. Its modular architecture supports real-time input from any traffic source, and the congestion scoring mechanism is data-driven and not location-specific. This ensures that the system can be easily replicated in urban areas with varying traffic dynamics, including metro cities and tier-2 regions, by retraining the model on local traffic data. In addition, the comparative analysis of the proposed work and the existing works has also been tabulated in Table 6.

Table 6. Comparison of our work with state-of-the-art approaches

Aspect	Tolani *et al.* [33]	Krishnasamy *et al.* [34]	More *et al.* [35]	Yasir *et al.* [36]	Proposed work
Goal	Signal control (SigCtrl)	Congestion prediction (Pred)	Routing optimization (RoutOpt)	Pred	Pred+Rout
Data	Prototype at Indian junction (Proto-IN)	Generic dataset from Kaggle (Kaggle-Global)	Simulated smart city grid (Sim-Grid)	Real Indian traffic data (Real-IN)	Real-IN
Model	Edge-optimized ML (EdgeML)	DL+optimizer (DL+Opt)	Reinforcement learning+multi-objective optimization (RL+MOO)	Regression-based ML (RegML)	RF+congestion value scoring (RF+Score)
Routing	No	No	Yes	No	Yes
Edge/RT	TinyML	Heavy	Simulation only (SimOnly)	Non-real-time prediction (Offline)	Real-time feasible (RT-Feasible)
TempRes	Real-time in seconds (RT-Sec)	High-frequency classification (High-freq)	Dynamic simulation environment (Dyn-Sim)	One-week-ahead prediction (Wk-Ahead)	15-minute interval prediction (15-min)
Traffic	Basic-IN	Generic	Simulated	Heterogeneous Indian traffic (Hetro-IN)	Hetro-IN
Key contributions	IoT-based signal adjustment (IoT+SigAdj)	DL-based prediction (DeepPred)	Smart routing +emission optimization (SmartRout+Emiss)	Prediction using weather/time/external factors (Weather+Pred)	Lightweight model+Routing+Real traffic data (Lite+Rout+Real)

The proposed CLEAR framework demonstrates strong relevance to the field of Electrical Engineering and Informatics by integrating ML, IoT architecture, and real-time informatics to address challenges in smart urban mobility. The system is designed to be lightweight and edge-deployable, making it suitable for implementation on low-cost microcontrollers or embedded platforms commonly used in smart city infrastructure. Moreover, the integration of real-time traffic data with an adaptive routing engine forms an end-to-end pipeline that exemplifies the informatization of urban transportation systems. This cross-disciplinary design shows clear potential for IoT-based traffic signal control, electrical sensor networks, and decision support systems, reinforcing its practical significance across both engineering and informatics domains.

8. CONCLUSION

This research introduces a practical and scalable ML-based framework for urban traffic congestion prediction, specifically designed to address the complexities of Indian traffic environments characterized by heterogeneous vehicles, a lack of lane discipline, and infrastructural limitations. By leveraging a novel, real-time dataset from the Palakkad region, the study achieves a high prediction accuracy of 98.6% using the RF classifier, which outperforms other evaluated algorithms in both efficiency and reliability. The proposed system goes beyond standalone prediction by integrating a high-frequency (15-minute interval) congestion forecasting module with an adaptive best-route recommendation engine, enabling timely and informed routing decisions. Its lightweight architecture ensures feasibility for real-world deployment on edge devices or within smart city infrastructure, even in resource-constrained urban settings. From an environmental perspective, the system holds potential to significantly reduce fuel consumption, vehicular emissions, and idle time by mitigating congestion through predictive interventions. As an extension, the framework can be scaled to multiple cities and integrated with live GPS feeds, traffic signal timing data, and context-aware adaptive models. Future work may also investigate the application of reinforcement learning algorithms for dynamic signal control, further improving the system's ability to optimize traffic flow and enhance urban mobility under dynamic, real-time conditions.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Sri Mata Amritanandamayi Devi (Amma), Chancellor, Amrita Vishwa Vidyapeetham, for her inspiration and for providing financial support for the Article Processing Charges (APC) of this publication.

FUNDING INFORMATION

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sreejith Karunakaran	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Senthilkumar Mathi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Preeja Pradeep	✓									✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest related to this work.

DATA AVAILABILITY

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Kumari, J. J. Janet, and S. Thangam, “CSG_IACO: an efficient stigmergic-based improved ACO routing strategy to determine the effective path based on traffic density in V2V networks,” *International Journal of Information Technology*, vol. 17, no. 3, 2025, pp. 1883-1908, doi: 10.1007/s41870-024-02386-9.

[2] K. Sreejith, S. Mathi, and P. Pradeep, “Beyond sensors: intellisignal’s map-integrated intelligence in traffic flow optimization,” *IEEE Access*, vol. 12, pp. 39028–39040, 2024, doi: 10.1109/ACCESS.2024.3375335.

[3] V. Bhardwaj, Y. Rasamsetti, and V. Valsan, “Traffic control system for smart city using image processing,” *AI and IoT for Smart*

City Applications, Springer Nature Singapore, 2022, pp. 83–99, doi: 10.1007/978-981-16-7498-3_6.
[4] D. Poojitha, T. Darak, U. Samaddar, and M. Khanna, "Smart traffic flow optimization with iot and machine learning," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, Jun. 2024, pp. 1–7, doi: 10.1109/ICCCNT61001.2024.10726030.
[5] *India's transport sector : the challenges ahead, volume 2. background papers*. Washington, DC, 2002, doi: 10.1596/15407.
[6] M. Kumar and K. Kumar, "Traffic congestion prediction using categorized vehicular speed data," in *Lecture Notes in Civil Engineering*, Springer Nature Singapore, 2023, vol. 273, pp. 367–384, doi: 10.1007/978-981-19-4204-4_22.
[7] N. A. M. Razali, N. Shamsaimon, K. K. Ishak, S. Ramli, M. F. M. Amran, and S. Sukardi, "Gap, techniques and evaluation: traffic flow prediction using machine learning and deep learning," *Journal of Big Data*, vol. 8, no. 1, pp. 1-25, Dec. 2021, doi: 10.1186/s40537-021-00542-7.
[8] S. Rajendran and B. Ayyasamy, "Short-term traffic prediction model for urban transportation using structure pattern and regression: an indian context," *SN Applied Sciences*, vol. 2, no. 7, pp. 1-11, Jul. 2020, doi: 10.1007/s42452-020-2946-2.
[9] A. A. Kashyap, S. Raviraj, A. Devarakonda, S. R. Nayak K, S. K V, and S. J. Bhat, "Traffic flow prediction models – a review of deep learning techniques," *Cogent Engineering*, vol. 9, no. 1, Dec. 2022, doi: 10.1080/23311916.2021.2010510.
[10] X. Bai, Y. Feng, L. Li, and L. Zhang, "Research on prediction of urban road congestion based on spark-gbdt," in *2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*, Beijing, China, Sep. 2020, pp. 102–106, doi: 10.1109/ICITE50838.2020.9231416.
[11] S.-S. Kim, M. Chung, and Y.-K. Kim, "Urban traffic prediction using congestion diffusion model," in *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, Seoul, Korea (South), Nov. 2020, pp. 1–4, doi: 10.1109/ICCE-Asia49877.2020.9276823.
[12] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, IEEE, Dec. 2017, pp. 361–364, doi: 10.1109/ISCID.2017.216.
[13] N. Dogru and A. Subasi, "Traffic accident detection using random forest classifier," in *2018 15th Learning and Technology Conference (L&T)*, IEEE, Feb. 2018, pp. 40–45, doi: 10.1109/LT.2018.8368509.
[14] Y.-J. Lee and O. Min, "Comparative analysis of machine learning algorithms to urban traffic prediction," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), Oct. 2017, pp. 1034–1036, doi: 10.1109/ICTC.2017.8190846.
[15] R. Aleksander and C. Paweł, "Recent advances in traffic optimisation: systematic literature review of modern models, methods and algorithms," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1740–1758, Dec. 2020, doi: 10.1049/iet-its.2020.0328.
[16] A. Lakshna, K. Ramesh, B. Prabha, D. Sheema, and K. Vijayakumar, "Machine learning smart traffic prediction and congestion reduction," in *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, Sep. 2021, pp. 1–4, doi: 10.1109/ICSES52305.2021.9633949.
[17] H. B. Tulay, F. Barickman, J. Martin, S. Rao, and C. E. Koksal, "Poster: urban traffic monitoring via machine learning," in *2019 IEEE Vehicular Networking Conference (VNC)*, Los Angeles, CA, USA, Dec. 2019, pp. 1–2, doi: 10.1109/VNC48660.2019.9062799.
[18] B. Priambodo, A. Ahmad, and R. A. Kadir, "Predicting traffic flow propagation based on congestion at neighbouring roads using hidden markov model," *IEEE Access*, vol. 9, pp. 85933–85946, 2021, doi: 10.1109/ACCESS.2021.3075911.
[19] P. Wang *et al.*, "Regional detection of traffic congestion using in a large-scale surveillance system via deep residual trafficnet," *IEEE Access*, vol. 6, pp. 68910–68919, 2018, doi: 10.1109/ACCESS.2018.2879809.
[20] H. Yan and D.-J. Yu, "Short-term traffic condition prediction of urban road network based on improved svm," in *2017 International Smart Cities Conference (ISC2)*, Wuxi, China, Sep. 2017, pp. 1–2, doi: 10.1109/ISC2.2017.8090856.
[21] F. Salahdine, S. Aggarwal, and A. Nasipuri, "Short-term traffic congestion prediction with deep learning for lora networks," in *Conference Proceedings - IEEE SOUTHEASTCON*, AL, USA, Mar. 2022, pp. 261–268, doi: 10.1109/SoutheastCon48659.2022.9763927.
[22] B. R. Krishna, M. H. Reddy, P. S. Vaishnavi, and S. V. Reddy, "Traffic flow forecast using time series analysis based on machine learning," in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, Mar. 2022, pp. 943–947, doi: 10.1109/ICCMC53470.2022.9753812.
[23] G. Meena, D. Sharma, and M. Mahrishi, "Traffic prediction for intelligent transportation system using machine learning," in *Proceedings of 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things, ICETCE 2020*, Jaipur, India, Feb. 2020, pp. 145–148, doi: 10.1109/ICETCE48199.2020.9091758.
[24] X. Zeng, Y. Wang, X. Deng, and J. Wang, "Short-term traffic flow prediction based on ensemble machine learning strategies," in *Proceedings of 2021 IEEE 10th Data Driven Control and Learning Systems Conference, DDCLS 2021*, Suzhou, China, May 2021, pp. 333–338, doi: 10.1109/DDCLS52934.2021.9455594.
[25] H. R. Deekshetha, A. V. S. Madhav, and A. K. Tyagi, "Traffic prediction using machine learning," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 116, Springer Singapore, 2022, pp. 969–983, doi: 10.1007/978-981-16-9605-3_68.
[26] B. Deb, S. R. Khan, K. T. Hasan, A. H. Khan, and M. A. Alam, "Travel time prediction using machine learning and weather impact on traffic conditions," in *2019 IEEE 5th International Conference for Convergence in Technology, I2CT 2019*, Bombay, India, Mar. 2019, pp. 1–8, doi: 10.1109/I2CT45611.2019.9033922.
[27] T. S. Tamir *et al.*, "Traffic congestion prediction using decision tree, logistic regression and neural networks," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 512–517, 2020, doi: 10.1016/j.ifacol.2021.04.138.
[28] A. Mystakidis and C. Tjortjis, "Big data mining for smart cities: predicting traffic congestion using classification," in *11th International Conference on Information, Intelligence, Systems and Applications, IISA 2020*, Piraeus, Greece, 2020, pp. 1–8, doi: 10.1109/IISA50023.2020.9284399.
[29] R. C. Gatto and C. H. Q. Forster, "Audio-based machine learning model for traffic congestion detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7200–7207, Nov. 2021, doi: 10.1109/TITS.2020.3003111.
[30] H. Shenghua, N. Zhihua, and H. Jiaxin, "Road traffic congestion prediction based on random forest and dbscan combined model," in *Proceedings - 2020 5th International Conference on Smart Grid and Electrical Automation, ICSGEA 2020*, Zhangjiajie, China, 2020, pp. 323–326, doi: 10.1109/ICSGEA51094.2020.00075.
[31] J. An, L. Fu, M. Hu, W. Chen, and J. Zhan, "A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information," *IEEE Access*, vol. 7, pp. 20708–20722, 2019, doi: 10.1109/ACCESS.2019.2896913.
[32] S. J. Kamble and M. R. Kounte, "Machine learning approach on traffic congestion monitoring system in internet of vehicles," *Procedia Computer Science*, vol. 171, pp. 2235–2241, 2020, doi: 10.1016/j.procs.2020.04.241.
[33] M. Tolani *et al.*, "Machine learning based adaptive traffic prediction and control using edge impulse platform," *Scientific Reports*,

vol. 15, no. 1, pp. 1-18, May 2025, doi: 10.1038/s41598-025-00762-4.
[34] L. Krishnasamy *et al.*, "Intelligent traffic congestion forecasting using bilstm and adaptive secretary bird optimizer for sustainable urban transportation," *Scientific Reports*, vol. 15, no. 1, pp. 1-17, May 2025, doi: 10.1038/s41598-025-02933-9.
[35] J. S. More, V. V. Sarbhukan (Bodade), and R. Lokare, "Adaptive network routing algorithm for intelligent and sustainable smart transportation systems," *International Journal of Computer Networks and Applications*, vol. 12, no. 4, pp. 487–499, Aug. 2025, doi: 10.22247/ijcna/2025/30.
[36] R. M. Yasir, D. N. Nower, and D. M. Shoyaib, "Traffic congestion prediction using machine learning techniques," *arXiv preprint*, 2022, doi: 10.48550/arXiv.2206.10983.
[37] D. M., R. H. G, and R. C. R, "Traffic congestion prediction using soft computing approach in cognitive internet of vehicles," *Journal of Computational Analysis and Applications*, vol. 33, no. 7, p. 2024, 2024.

BIOGRAPHIES OF AUTHORS

Sreejith Karunakaran obtained his Bachelor of Technology in Computer Science and Engineering from Cochin University of Science and Technology (CUSAT). Subsequently, he embarked on an M.Tech. degree specializing in Wireless Networks and Applications at Amrita Vishwa Vidyapeetham. Furthermore, he possesses a Master of Business Administration degree in Human Resources from Sikkim Manipal University. He possesses more than eight years of expertise as an Assistant Professor. He is presently a part-time doctoral candidate in the Amrita School of Computing, Amrita Vishwa Vidyapeetham, in Coimbatore, India. He has authored and published three papers in internationally recognized conferences indexed in Scopus. His research focus primarily includes intelligent transportation systems, artificial intelligence, computer networks, and machine learning. He can be contacted at email: k_sreejith@cb.students.amrita.edu.

Senthilkumar Mathi received his Bachelor of Engineering (Computer Science and Engineering) degree from Bharathiar University, India, Master of Engineering (Computer Science and Engineering) degree from Anna University, India, Master of Business Administration (Human Resources Management) degree from Bharathiar University, India and Doctor of Philosophy (under the Faculty of information and communication engineering) degree in Internet Protocol Mobility and Security, from Anna University, India. He joined the Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore Campus, India, in 2004 and currently serves as a professor. He has around 22 years of teaching and research experience. His areas of interest include IPv4 mobility and IPv6 mobility–next generation internet protocol mobility, vehicular networks, internet of things, wireless sensor networks, mobile computing, cryptography and information security, machine learning and formal languages, and automata theory. He has published research papers in reputed journals and conference proceedings, most of which are indexed in Scopus, SCI, and Web of Science. He has received the best paper awards for his publications. He is a member of IETE. He has guided numerous B.Tech. and M.Tech. students and Ph.D. scholars in their academic projects, theses, and research and continues to supervise students at all three levels. He can be contacted at email: m_senthil@cb.amrita.edu.

Preeja Pradeep is a Postdoctoral Researcher at the School of Computer Science and IT, University College Cork, Ireland. She holds a Ph.D. in Computer Science, M.Tech. in Wireless Networks and Applications, and M.Sc. in Computer Science from Amrita Vishwa Vidyapeetham, Amritapuri Campus, India. Her research focuses on edge-cloud computing, data analysis, explainable AI (XAI), and internet of things (IoT). She has 10 years of technical experience working on various interdisciplinary national and internationally funded research projects in an academic R&D setting. She can be contacted at email: ppradeep@ucc.ie.