Traffic Congestion Recognition Using Ensemble Learning with Metaheuristic OOBO-Based Hyper-Parameter Optimization

Akashdeep Singh[1], Shruti Bansal[2], Kulwinder Singh[3]
[1,2,3] Bhai Maha Singh College of Enginnering, Sri Muktsar Sahib.

Abstract: In this study, we have presented a traffic congestion recognition method based on ensemble learning and hyper-parameter setting based on the metaheuristic OOBO algorithm. Initially, the standard dataset was collected from the open-source Kaggle repository. Thereafter, the preprocessing of the dataset was done by finding the null/missing value in the dataset, normalization, and the k-means clustering algorithm. The pre-processed dataset was split into training and testing sets. Next, an ensemble learning approach was developed in which five ML algorithms (logistic regression, decision tree, random forest, support vector regressor, and gradient boosting) were trained using the training set and validated using the testing set. Based on their individual performance, the minimum voting classifier was used to determine which ML algorithm outperforms the others. Finally, the hyper-parameter setting of the selected ML algorithm was done using the metaheuristic OOBO algorithm based on the RMSE objective function. Finally, the proposed method was evaluated against the existing method by determining the various performance indices, namely, Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Determination of Coefficient. The result indicates that the GB-OOBO algorithm achieves an RMSE value of 754.43, an MAE value of 543.78, and an R-squared value of 0.49.

Keywords: Ensemble Learning, Hyper-Parameter Tuning, Machine Learning, Metaheuristic, Traffic Congestion

1. Introduction

After industrialization, the population is growing day by day, resulting in an increase in vehicles on the road as well. In today's time, vehicles are the most essential things in fast-growing societies. As a result, there is a significant impact of vehicles on societies and the environment. Urban areas have a higher number of vehicles compared to rural areas, and this number is continuously increasing [1]. This results in high traffic congestion on roads. This means that, at a particular time, a part of the road has too many vehicles, due to which the road is blocked and no one is able to move their vehicles. Moreover, higher traffic results in several problems like time loss, pollution, stress, and anger. Traffic congestion on the road requires more time to clear and thus results in time delays, emission of extra pollution from vehicles that get stuck in congestion, and economic waste as well. A 2010 study found that cars generally emit 16% of the world's CO_2. Other harmful gases like reactive hydrocarbons and nitrogen oxides are also released by cars. These gases are responsible for the increase in temperature. For example, in 2017, the increase in temperature was 0.71C. One serious problem is poor air quality, which is a major threat worldwide. It causes several diseases related to the respiratory, like asthma and COPD (chronic obstructive pulmonary disease). A report by the US Department of Transportation states that traffic congestion and its causes are differentiated based on their sources. It is also evident that we cannot completely eliminate congestion; we can just try to reduce it. Two methods of traffic prediction, short-term and long-term traffic prediction, help drivers avoid the route that has higher traffic. We need to build infrastructure to handle the need for more computing power and more accurate models, and we also need to handle the variations of different traffic control systems in different countries. It is more critical to make better use of the current transportation infrastructure than to build new extensions to it [2].

However, controlling the traffic manually is not very effective and is very time-consuming and tiresome [3-4]. But with the use of automated and adaptive technologies, we can overcome certain drawbacks of manual techniques. Also, predicting traffic congestion is not always accurate, so the system might result in unpredictable events. Some barriers in predicting traffic congestion include lack of capacity, fixed time of lights, and unlimited user demand. As a result, it is necessary to use real-time solutions and algorithms that can solve the problems of congestion. For routing traffic based on density, reinforcement methods can be used that are based on rewards and punishment. Also, it is required to continuously update and make the real-time system learn from traffic patterns for more efficient predictions. However, in recent times, the use of ML algorithms in this field has gained popularity [5-6]. In this study, the main focus is to

enhance the performance of the traffic congestion recognition method by overcoming the individual machine learning algorithms' challenges using ensemble learning and hyper-tuning the parameters of it.

The rest of the structure is as follows. Section 2 presents the related work on traffic congestion. Section 3 illustrates the proposed methodology. Section 4 shows the results and analysis. Finally, the conclusion is drawn in Section 5.

2. Related Work

Gupta et al. [8], proposed a hybrid approach to improve the prediction of traffic congestion. The proposed model uses K-means clustering with ML models like DT, LR, RF, GB, and SVR to recognize unique traffic patterns. The traffic features included traffic volume, road length, and time of day. The study showed the hybrid approach achieved significantly higher performance than the single ML models. Several performance metrics were estimated to show the efficacy of the hybrid approach, such as RMSE, MAE, and R^2. The study demonstrated that the RMSE value of the GB algorithm significantly decreased from 2180.85 (without clustering) to 620.84 (with clustering). For the RF model, it decreased from 2186.32 to 632.07. Similarly, other models also showed significant improvements. The higher R2 value was observed for the hybrid model, reaching 0.92, which was significantly higher than the R2 value of the single model. Additionally, the SVR model also resulted in decreased RMSE and an R2 improvement from 0.065 (without clustering) to 0.713 (with clustering). The lowest MAE was observed for SVR with a value of 301.46, which shows the model's capability to reduce prediction error after clustering. **Alsubai et al. [9],** introduced the IAOADL-TCC (Improved Arithmetic Optimization with Deep Learning Driven Traffic Congestion Control) system for ITS (Intelligent Transportation System) in smart cities. The proposed model collects traffic data and uses existing routes to divert traffic so that traffic congestion can be avoided. The IAOADL-TCC model uses HCNN-ALSTM (hybrid CNN attention long short-term memory) for controlling the traffic congestion problem. After that, the research used an IAOA-based strategy for tuning hyperparameters to modify the parameters of the HCNN-ALSTM model optimally. The research demonstrated that the proposed model results in improvement in the traffic flow and congestion reduction. For validation, the research used the Kaggle dataset of road traffic. On this dataset, the proposed model achieved an accuracy of 98.03%, which is higher than other DL approaches. **N. Deepika and G. Pandove [10],** utilized four ML models to predict congestion in US cities. Different regression-based models were used in this research to predict congestion. The research presented the performance of various ML models using common performance metrics like MAE, RMSE, and R2. The study applied XGBoost with optimization and detrending to improve the prediction capabilities. The proposed model achieved the lowest RMSE value of 10.99 and the lowest MAE value of 6.39. The results showed the variability in the model's effectiveness. **Zhang et al. [11],** proposed a model for traffic congestion control by researching the traffic conditions and using urban traffic and congestion conditions. The study first proposed a fusion model using IPSO (improved PSO), RBF (optimized radial basis function), LSTM/SVM (long and short-term memory network/support vector machine) techniques to predict traffic state in urban areas. The traffic sample dataset for this study was acquired from some regions of Shenyang. Using this data, an experiment was designed, and models were used to evaluate prediction results and fitting values. The results of the error comparison analysis revealed the lower RMSE value of 13.2653, MAE value of 11.0113, MAPE value of 13.6345, BIC value of 7.511, and AIC value of 5.991. **M. Anjaneyulu and M. Kubendiran [12]**, aimed to predict short-term traffic congestion (STTC) every 5 minutes over 1 hour. For this, the research proposed a hybrid XPSVM (Xception support vector machine) model to forecast STTC. Basically, the exception model uses ReLU, separable convolution, and convolution methods to forecast feature detection in the dataset. Moreover, the SVM model predicts the result efficiently using a fine-tuned binary hyperplane and weight regularization method. The data for this research was collected from Google Maps. The dataset included screenshots that were taken using the Selenium automation tool from Karnataka and Bangalore. The proposed model achieved an accuracy of 97.16% for predicting traffic congestion. **Mehdi et al. [13],** proposed a labelling system for short-time congestion state using vehicle count and speed, followed by a CNN forecast based on meta-parameters. This research was based on an urban smart city project that aims to make decisions on different modes. First, this study models a unique entropy-based technique with traffic parameters like traffic density and vehicle speed. These entropy-based techniques are able to make accurate predictions even with random traffic parameters. After that, the research used a DNN for predicting the mean speed from parameters. The proposed strategy achieved superior performance over traditional methods and was able to predict congestion flow with improved accuracy. **Gollapalli et al. [14],** proposed an intelligent traffic congestion

prediction model using cloud technology and a hybrid Neuro-Fuzzy technique. The study also aims to reduce the queue delay at different road junctions in the city. The model proposed was also useful for controlling automated traffic systems by reducing congestion, especially in smart cities where the data is gathered from various IoT sensors on the road. The first step is pre-processing, which is performed on the cloud server. Then, this data is used by the neuro-fuzzy engine. The proposed approach achieved a higher level of accuracy with a minimum error rate. The experimental findings showed that the model achieved an accuracy of 98.72% during validation. This is higher than the accuracies achieved by existing studies such as 95.84%, 90.6%, and 98.03%. The proposed system shows an accuracy of 99.214% in the training phase analysis. The proposed prediction model has the ability to improve the environment of smart cities. **Bokaba et al. [15],** developed a model for predicting road traffic congestion using several ensemble methods. These ensemble methods were DT, RF, AdaBoost, and SVM. The results of the study showed that the AdaBoost model achieved an accuracy of 99.7% with a misclassification cost of 24. The results of the AdaBoost model were better than those of the other models in terms of recall, precision, and F1-score. It was assumed that the proposed model could reduce the high number of traffic collisions and congestion. AdaBoost performed very well because it was able to handle the dataset effectively. This proved the suitability of the AdaBoost model for this type of dataset and problem. But this data didn't include road collision and weather data. This could be one reason for AdaBoost's significant performance. **Z. Miao and Q. Liao [7],** employed the CNN algorithm for recognizing traffic congestion and hyper-tuned its parameters using the metaheuristic PSO algorithm. The dataset in their work was collected from the open-source Kaggle database. The database target variable is available in the categorical format. Therefore, the classification approach was used, and they have achieved an accuracy value of 92.0%.

In the literature, several machine learning algorithms, namely, Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM), are successfully utilized for recognizing traffic congestion [7-15]. However, individual machine learning algorithms have bias and variance issues, and their performance is also dependent on the size of the data. Therefore, in this research, these challenges will be taken into consideration to design an enhanced traffic congestion recognition method that combines the multiple machine learning algorithms using the ensemble learning approach and hyper-tunes the parameters of it using the metaheuristic algorithms. Besides that, a preprocessing method will be designed to cluster the similar traffic attributes in the database.

3. Proposed Methodology

This section presents the traffic congestion recognition method is designed using machine learning algorithms for effective traffic management purposes. The block diagram of the proposed traffic congestion method is shown in Figure 1.

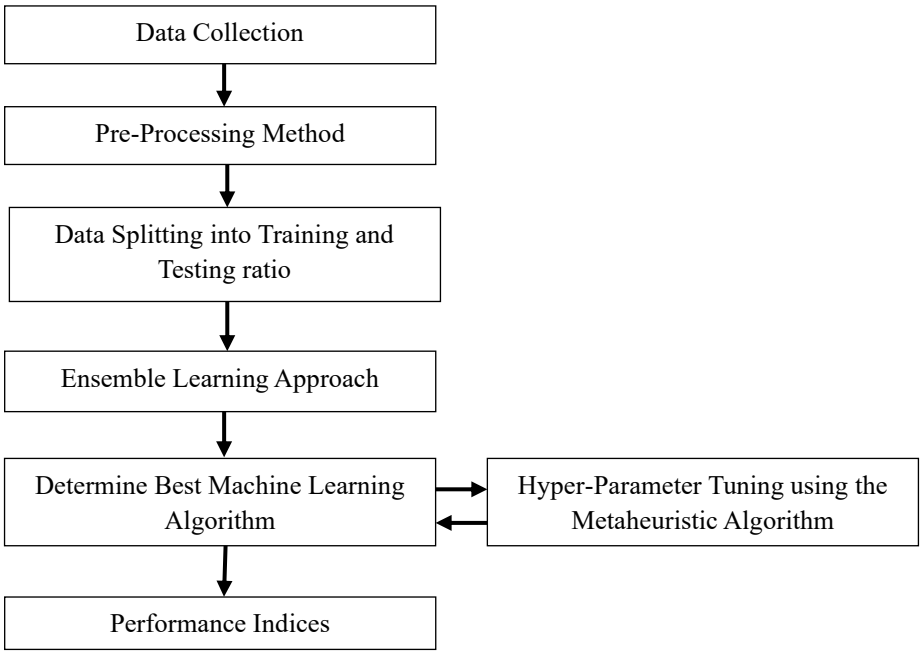


Figure 1: Block Diagram of the Proposed Traffic Congestion Recognition Method

Initially, in the proposed method, the standard dataset of traffic congestion was collected from the secondary database, such as Kaggle. Thereafter, preprocessing of the data was done to find out the missing or null values in the database and to cluster the traffic patterns using clustering methods. Next, the pre-processed database was done to split into training and testing splitting ratios to train and validate the performance of the machine learning algorithms. In the proposed method, multiple machine learning algorithms were evaluated and best algorithm was chosen using the ensemble learning approach and hyper-tuned using the metaheuristic algorithm to enhance congestion recognition compared to the previous approaches. Finally, various performance indices were determined to evaluate the proposed traffic congestion recognition method over the existing approaches. The detailed description of various steps is given below.

3.1 Data Collection and Preparation: In this research, the secondary database of the traffic congestion was taken into consideration to evaluate the proposed method. The secondary database utilized in the literature is available on the Kaggle website [16]. In this step, the attributes of the database was analyzed to find out if there are any missing or null values and if the data is balanced or not. After that, clustering methods, such as K-means clustering, was employed to cluster the traffic attributes. In this step, the pre-processed database was splitted into training and testing ratios to validate the performance of the machine learning algorithm. In the literature, the most preferred training and testing ratios are 60:40, 70:30, 80:20, and 90:10. Out of these, 70:30 ratio was used in this study.

3.2 Traffic Congestion Prediction using Machine Learning Algorithms: In this step, the traffic congestion recognition was done using the machine learning algorithms. The multiple machine learning algorithms were combined using the ensemble learning approach to determine which algorithm outperforms the others through a voting regressor in the proposed method. After determining the most suitable machine learning algorithm, the performance of it was enhanced by hyper-tuning of the parameters.

Hyper-tuning was performed using optimization approaches, such as metaheuristics. The metaheuristic algorithm searches for the most suitable value in the lower and upper bounds of the parameter value based on the objective function. In the proposed method, RMSE was taken as the objective function. After determining the best parameter value, the machine learning algorithm was hyper-tuned with the best parameter value, and the traffic congestion recognition is done.

Figure 2 shows the ensemble learning and hyper-parameter tuning process for the proposed method using the ensemble learning and metaheuristic algorithm. The pre-processed data was given to the individual ML algorithms, and their evaluation was done using various performance metrics, as done in the bagging ensemble learning approach. After that, the minimum voting regressor was used to select the ML algorithm that provides the minimum RMSE value over others. Finally, the selected ML algorithm was hyper-tuned using the metaheuristic OOBO algorithm.

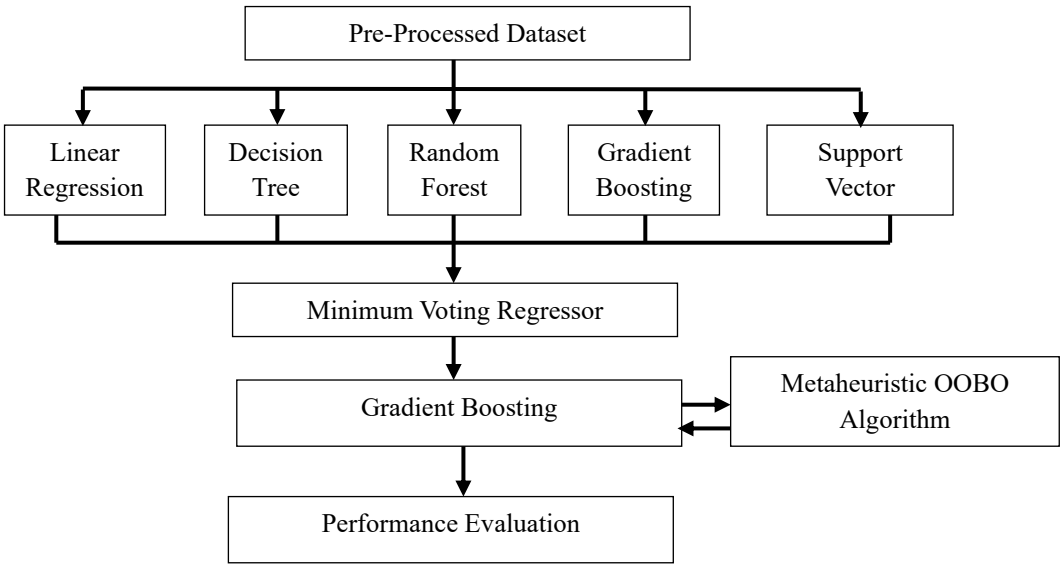


Figure 2: Flowchart of the Ensemble Learning and Hyper-tuning using the Metaheuristic OOBO Algorithm

3.3 Performance Indices

In this step, various performance indices were measured to evaluate the proposed method. In the literature, error indices, such as Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and determination of Coefficient (R^2), are measured. The dataset utilized in the proposed traffic congestion method is available in continuous discrete form. Therefore, the ML algorithms were employed as regression models and evaluated using error parameters, including RMSE, MAE, and the coefficient of determination. These parameters are explained below [17-20].

- **RMSE:** It estimates the square root of the average squared difference between the actual and predicted values. This measure is commonly used in regression problems to assess the model's effectiveness. A lower RMSE value is generally considered good. Using this metric, it is easy to assess the model's effectiveness, especially when the values of the target variable vary significantly.

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

- **MAE:** This parameter measures the mean absolute error between the actual and predicted value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (2)$$

- **R-Squared (R^2):** In regression models, this measure is used to calculate the percentage of variance of the dependent variable that can be explained by one or more independent variables.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (3)$$

4. Results and Analysis

This section presents the simulation evaluation of the proposed traffic congestion method and comparative analysis with the previous approaches. The standard dataset of traffic congestion was taken from the open-source Kaggle database [16]. Thereafter, the proposed method was coded and simulated in MATLAB 2018a software. Table 1 presents the simulation setup configuration of the proposed traffic congestion method. This configuration defines the sample size, training/testing ratio, number of clusters, and parameter values of the ML algorithms utilized in the proposed method. Finally, in this table, the metaheuristic OOBO algorithm is defined as being utilized to hyper-tune the parameters of the best ML algorithm over others.

Table 1: Simulation Setup Configuration of the Traffic Congestion Method

Parameter	Value
Data Sample Size	104,858
Training and Testing Ratio	70:30
Number of Clusters	2
Num-Learning-Cycles for GB	100
Num-Learning-Cycles for RF	100
Population and its dimension	[5 3]
Total Iterations	100
Lower Bound of Parameters	[50 0.01 1]
Upper Bound of the Parameter	[300 0.5 30]
Objective Function	RMSE

4.1 Simulation Results

Figure 3 shows the K-means clustering with K = 2 applied to the traffic data to separate congestion levels in the pre-processing stage to cluster the dataset. The clustering results show two clusters. In Figure 5.1, cluster 1 (red) represents

the high volume of traffic, while cluster 2 (blue) represents low-to-moderate traffic volume. The boundary is cluster is almost constant during the entire day, which shows that traffic volume mainly segments the data. However, high values of volume are observed during peak periods (particularly the rush hours like 7:00, 16:00, and 17:00), but congestion occurs throughout the day,

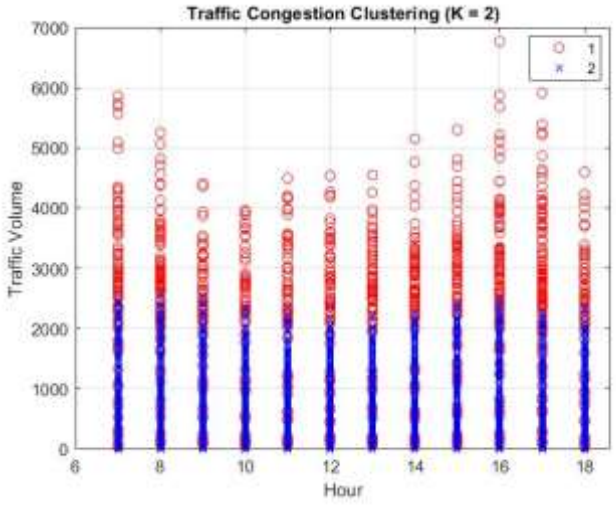


Figure 3: K-Mean Clustering

Table 2 shows the evaluation of five individual ML algorithms used for traffic congestion recognition. These algorithms were compared using three performance metrics, namely, RMSE, MAE, and R^2. The result indicates that the GB algorithm achieves the lowest values of the RMSE and MAE parameters and the highest value of R^2. From the results, it can be seen that the LR and SVR algorithms performed poorly with R^2 values of 0.01 and -0.13, respectively. The lower values of R^2 show that there is non-linear relationship between the traffic volume and predictor variables.

Table 2: Evaluation of Individual ML Algorithms

ML Algorithm	**RMSE**	**MAE**	**R^2**
LR	1036.55	808.34	0.01
DT	834.76	588.96	0.36
GB	**804.87**	**576.59**	**0.41**
RF	832.02	634.42	0.37
SVR	1108.61	751.34	-0.13

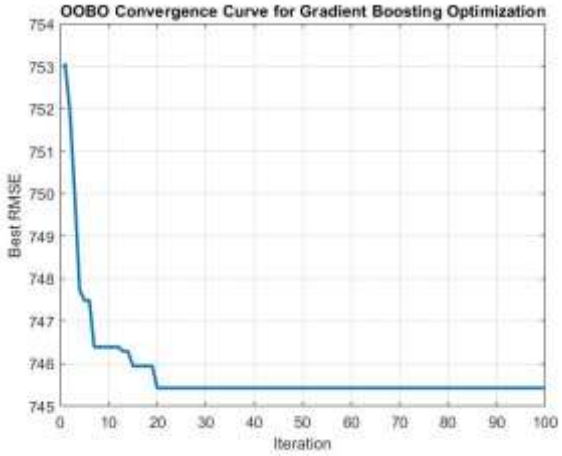


Figure 4: Convergence Rate Graph for OOBO Algorithm

Based on the individual performance of the ML algorithm, the GB algorithm was chosen using the voting regressor in the ensemble learning approach. Furthermore, hyper-parameter tuning using the metaheuristic OOBO algorithm was done for the GB algorithm to enhance its performance. The meta-heuristic algorithm searches the best parameter values based on the objective function. The convergence rate graph for the OOBO algorithm is plotted in Figure 4. It shows that the objective function value is reduced from 753.0735 to 745.4267. Table 3 shows the optimal parameter value of the GB algorithm along with the objective function achieved using the OOBO algorithm. After determining the best values, the GB algorithm was initialized with these values for traffic congestion recognition.

Table 3: Optimal Parameters Value of GB Algorithm

GB Parameter	**Value**
Number of Trees	94
Learning Rate	0.1813
Minimum Leaf Size	12
Objective Function	745.4267

Table 4 presents the evaluation of the individual ML algorithms with respect to the GB algorithm is hyper-tuned using the metaheuristic OOBO algorithm. The proposed model GB-OOBO results in reduced values of RMSE and MAE and an increased R^2 value as compared to the individual GB (Gradient Boosting) model. The RMSE value decreased by 6.27% (50.44), and MAE by 5.69% (32.81), while the value of R^2 increased by 19.51% from 0.41 to 0.49. This clearly shows that the proposed model is able to capture more variations in traffic volume and results in more accurate predictions. As compared to other individual baseline models, GB-OOBO reduced RMSE by 9.3% relative to the RF model, 9.6% relative to the DT model, 31.9% relative to the SVR model, and 27.2% relative to the LR model.

Table 4: Evaluation of Individual ML Algorithms with Respect to the GB-OOBO Algorithm

ML Algorithm	**RMSE**	**MAE**	**R^2**
LR	1036.55	808.34	0.01
DT	834.76	588.96	0.36
RF	832.02	634.42	0.37
SVR	1108.61	751.34	-0.13
GB	804.87	576.59	0.41
GB-OOBO	**754.43**	**543.78**	**0.49**

Table 5 shows the comparison of the proposed GB-OOBO model with existing results reported by Gupta et al. [8]. It is evident that the proposed model performed better than the LR and SVR models. As compared to the LR and SVR models [8], the proposed model achieved a reduction in RMSE of 66.0% and 5.1%, respectively. However, the RMSE values of GB, RF, and DT are lower than those of the proposed method. There might be several reasons for this difference, such as variations in the dataset, feature selection, pre-processing, strategy of data splitting, and evaluation methods in both studies.

Table 5: Comparative Analysis with the Existing Method

Methods	**ML Algorithms**	**RMSE**
Gupta et al. [8]	LR	2219.92
	GB	620.84
	RF	632.07
	DT	640.80
	SVR	795.05
Proposed Method	**GB-OOBO**	**754.43**

5. Conclusion

In this study, we have developed a traffic congestion recognition method based on ensemble learning and metaheuristic algorithms for overcome the sensitivity of individual ML algorithm on the dataset and hyper-parameter setting. In the proposed method, the standard dataset was collected from the open-source Kaggle repository. The dataset was pre-processed using the k-mean clustering and splitted into training and testing sets. After that, the training set was given to the five ML algorithms utilized in the ensemble learning approach and validated using the testing set. Based on their individual performance, voting classifier was utilized to determined which ML algorithm outperforms over other. Next, the selected ML algorithm parameters, hyper-tuned using the metaheuristic OOBO algorithm based on the RMSE objective function. The OOBO algorithm searches the solution space of the parameters in the lower and upper bound based on the population and iteration size. Finally, the parameter setting was done using the best values and traffic congestion recognition was done. From the simulation evaluation, we found that GB-OOBO algorithm achieves the lowest error metrics and high R-squared value over the individual ML algorithms.

References

[1] Y. Modi, R. Teli, A. Mehta, K. Shah, and M. Shah, "A comprehensive review on intelligent traffic management using machine learning algorithms," *Innovative Infrastructure Solutions*, vol. 7, no. 1, Dec. 2021, doi: 10.1007/s41062-021-00718-3. Available: https://doi.org/10.1007/s41062-021-00718-3

[2] A. Agrahari, M. M. Dhabu, P. S. Deshpande, A. Tiwari, M. A. Baig, and A. D. Sawarkar, "Artificial Intelligence-Based Adaptive Traffic Signal Control System: A Comprehensive Review," *Electronics*, vol. 13, no. 19, p. 3875, Sep. 2024, doi: 10.3390/electronics13193875. Available: https://doi.org/10.3390/electronics13193875

[3] Bakir, D., Moussaid, K., Chiba, Z. and Abghour, N., 2024, December. A comprehensive review of traffic congestion prediction models: Machine learning and statistical approaches. In *2024 IEEE International Conference on Computing (ICOCO)* (pp. 539-545). IEEE.

[4] R. Vatambeti *et al.*, "Enhancing urban traffic congestion prediction through efficientnet and optimized ensemble learning models," *Scientific Reports*, vol. 15, no. 1, p. 40224, Nov. 2025, doi: 10.1038/s41598-025-24012-9. Available: https://doi.org/10.1038/s41598-025-24012-9

[5] T. Zhang and S. Mao, "Machine Learning for End-to-End Congestion Control," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 52–57, Jun. 2020, doi: 10.1109/mcom.001.1900509. Available: https://doi.org/10.1109/mcom.001.1900509

[6] L. Hammoumi, S. Farah, M. Benayad, M. Maanan, and H. Rhinane, "Leveraging machine learning to predict traffic jams: Case study of Casablanca, Morocco," *Journal of Urban Management*, vol. 14, no. 3, pp. 813–826, Mar. 2025, doi: 10.1016/j.jum.2025.02.004. Available: https://doi.org/10.1016/j.jum.2025.02.004

[7] Z. Miao and Q. Liao, "IoT-Based traffic prediction for smart cities," *IEEE Access*, vol. 13, pp. 52369–52384, Jan. 2025, doi: 10.1109/access.2025.3552276. Available: https://doi.org/10.1109/access.2025.3552276

[8] S. Gupta, A. Kumar, and A. Kumar, "Analysis of Traffic Flow Congestion by Integrating Supervised Machine Learning with K-mean Clustering," *SN Computer Science*, vol. 6, no. 3, Mar. 2025, doi: 10.1007/s42979-025-03761-4. Available: https://doi.org/10.1007/s42979-025-03761-4

[9] S. Alsubai, A. K. Dutta, and A. R. W. Sait, "Hybrid deep learning-based traffic congestion control in IoT environment using enhanced arithmetic optimization technique," *Alexandria Engineering Journal*, vol. 105, pp. 331–340, Jul. 2024, doi: 10.1016/j.aej.2024.06.072. Available: https://doi.org/10.1016/j.aej.2024.06.072

[10] N. Deepika and G. Pandove, "A comparison of ML models for predicting congestion in urban cities," *International Journal of Intelligent Transportation Systems Research*, vol. 22, no. 1, pp. 171–188, Feb. 2024, doi: 10.1007/s13177-024-00387-3. Available: https://doi.org/10.1007/s13177-024-00387-3

[11] T. Zhang, J. Xu, S. Cong, C. Qu, and W. Zhao, "A hybrid method of traffic congestion prediction and control," *IEEE Access*, vol. 11, pp. 36471–36491, Jan. 2023, doi: 10.1109/access.2023.3266291. Available: https://doi.org/10.1109/access.2023.3266291

[12] M. Anjaneyulu and M. Kubendiran, "Short-Term traffic congestion prediction using hybrid deep learning technique," *Sustainability*, vol. 15, no. 1, p. 74, Dec. 2022, doi: 10.3390/su15010074. Available: https://doi.org/10.3390/su15010074

[13] M. Z. Mehdi, H. M. Kammoun, N. G. Benayed, D. Sellami, and A. D. Masmoudi, "Entropy-Based traffic flow labeling for CNN-Based Traffic congestion prediction from Meta-Parameters," *IEEE Access*, vol. 10, pp. 16123–16133, Jan. 2022, doi: 10.1109/access.2022.3149059. Available: https://doi.org/10.1109/access.2022.3149059

[14] M. Gollapalli *et al.*, "A Neuro-Fuzzy approach to road traffic congestion prediction," *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, vol. 73, no. 1, pp. 295–310, Jan. 2022, doi: 10.32604/cmc.2022.027925. Available: https://doi.org/10.32604/cmc.2022.027925

[15] T. Bokaba, W. Doorsamy, and B. S. Paul, "A Comparative study of ensemble models for predicting road traffic congestion," *Applied Sciences*, vol. 12, no. 3, p. 1337, Jan. 2022, doi: 10.3390/app12031337. Available: https://doi.org/10.3390/app12031337

[16] Möbius, "Road Trafic Dataset," *Kaggle.com*, 2018. https://www.kaggle.com/datasets/arashnic/road-trafic-dataset

[17] A. Navarro-Espinoza *et al.*, "Traffic flow prediction for smart traffic lights using machine learning algorithms," *Technologies*, vol. 10, no. 1, p. 5, Jan. 2022, doi: 10.3390/technologies10010005. Available: https://doi.org/10.3390/technologies10010005

[18] Y. Zhai, Y. Wan, and X. Wang, "Optimization of Traffic Congestion Management in Smart Cities under Bidirectional Long and Short-Term Memory Model," *Journal of Advanced Transportation*, vol. 2022, pp. 1–8, Apr. 2022, doi: 10.1155/2022/3305400. Available: https://doi.org/10.1155/2022/3305400

[19] N. Zarbakhsh and G. McArdle, "Predicting Traffic Congestion During Covid19 Using Human Mobility and Street-Waste Features," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-4/W3-2022, pp. 301–308, Oct. 2022, doi: 10.5194/isprs-annals-x-4-w3-2022-301-2022. Available: https://doi.org/10.5194/isprs-annals-x-4-w3-2022-301-2022

[20] D.-H. Shin, K. Chung, and R. C. Park, "Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data," *IEEE Access*, vol. 8, pp. 150784–150796, Jan. 2020, doi: 10.1109/access.2020.3016469. Available: https://doi.org/10.1109/access.2020.3016469