Proceedings of the 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI-2025)
DVD Part Number: CFP25US4-DVD; ISBN: 979-8-3315-2265-0

Score-Level Fusion Approach for Traffic Flow Prediction in Intelligent Transportation System

Axcellin.T
Research Scholar
Department of Computer Science
Vel's Institute of Science, Technology and Advanced Studies
Chennai, Tamil Nadu, India
axcellinrajesh@gmail.com

S. Kamalakkannan
Professor,
Department of Computer Science
Vel's Institute of Science, Technology and Advanced Studies
Chennai, Tamil Nadu, India
Kannan.scs@vistac.ac.in

A***bstract*****:** Traffic flow prediction (TFP) for intelligent transportation system is essential for effective transportation planning and city development. Conventional models often have difficulty with the complex nature of traffic data, which includes nonlinear patterns, spatial dependencies, and external influences such as weather or road conditions. Prediction accuracy may be significantly increased by utilizing an ensembling strategy, which combines the capabilities of many models. The goal of the study is to improve TFP accuracy by integrating the results of different machine learning (ML) models. The study uses Random Forest(RF) Naïve Bayes(NB), K-Nearest Neighbor(KNN), Extreme Gradient Boosting(XGBoost) and Support vector machine(SVM) as prediction models and based on the prediction probabilities and the score level fusion is performed with RF and XGBoost ensembling. The new fusion procedure uses a weighted average technique in which weights are assigned dynamically depending on individual model performance criteria including accuracy, precision, recall, and F1-score. Higher-performing algorithms are given greater significance in the final forecast, resulting in a dynamic and data-driven assembly strategy. The findings show that the efficiency of the ensemble of RF-XGBoost is higher compared to the contrasted methods. The model also uses feature selection approaches like mutual information to find the most significant traffic-related characteristics, lowering computational cost and increasing forecast efficiency.

Keywords: Traffic flow Prediction(TFP), Intelligent Transportation Systems(ITSs), Mutual Information, Random Forest(RF) Naïve Bayes(NB), K-Nearest Neighbor(KNN), Extreme Gradient Boosting(XGBoost) and Support vector machine(SVM)
.

I.INTRODUCTION

With the fast growth of the world's population, there is a rise in traffic congestion, a greater need for transportation, inadequate accessibility, and decreasing productivity as a result of urbanization. Despite ongoing scientific and technological advancements, many major cities throughout the world continue to lack sustainable modes of passenger and freight transit. Traffic congestion costs billions of dollars each year due to reduced efficiency, air pollution, and fuel loss, among numerous other factors[1].

Effective transportation systems are vital city infrastructure, especially in resource-constrained smart cities. Rapid breakthroughs in technology for communication and information are paving the way for ITSs, which are especially intended to work effectively and safely with current transportation infrastructure. One important aspect of ITS is its capacity to combine vast amounts of data from multiple sources for the detection of events[2].

Intelligent transportation systems (ITSs) are a network of interconnected infrastructures utilizing advanced technologies to improve traffic management and safety. Traffic flow forecasting is an essential aspect of systems for intelligent transportation. Intelligent transport systems (ITSs) attempts to improve the levels of service for different traffic infrastructure, alleviate traffic congestion, and avoid traffic accidents. TFP, a critical component of ITSs, refers to identifying fundamental trends from recorded traffic flow data and applying these patterns to predict potential traffic conditions. Accurate, effective, and reliable TFP may help passengers make accurate travel decisions as well as build dependable proactive traffic management strategies[3]. The major contributions are described below:

- The study discusses the approaches, focusing on those that are relevant and useful to the ITS literature.
- The study employs feature extraction using Mutual information based measures.
- The study designs an ensembling of RF and XGBoost employing score level fusion

Section 2 provides a key research findings on TFP for ITS. Section 3 describes the approach and the system model with preprocessing and feature extraction and prediction using the score level fusion ML models. Section 4 presented the findings and discussions of the outcomes. Finally, Section 5 discusses the conclusions and future scope of this research

II. RELATED WORKS

Several studies on traffic flow forecasting have been undertaken over the last few decades. However, most previous research has focused on building new algorithms or models to achieve innovative predicting accuracy. Ou et al., (2024) proposes an interpretable movement of traffic forecasting system based on widely used tree-ensemble techniques. The framework consists of many important components that are merged into an extremely flexible and customized multi-stage pipeline, allowing for the smooth inclusion of multiple algorithms and tools. To test the framework's performance, the generated tree-ensemble models and another three usual kinds of baseline models, encompassing statistical time series,

979-8-3315-2266-7/25/$31.00 ©2025 IEEE

shallow learning, and deep learning, were evaluated on three datasets gathered from various types of roads[3]

Nagarajan et al.,(2024) presents a new Dampster-Shafer data fusion-based Adversarial Deep Learning (DS-ADL) Model for ITS in fog cloud situations. The suggested approach considers three types of adversarial attacks: image level, feature level, and decisions level. Adversarial cases are developed at every stage to fully assess the system's susceptibility. To improve the system's capabilities, researchers harness the power of various critical components[4].

GNNs provide a potential helpful framework for capturing complicated patterns and relationships among varied components, such as segments of road and crossings, by taking into account both temporal and spatial dependencies. GNN-based traffic forecasting has lately been examined in numerous research; however, complete evaluations of information fusion methodologies for GNN-based traffic forecasts, including an examination of their benefits and limitations, are required. Ahmed et al., (2024) study fills the knowledge gap and provides future insights into prospective breakthroughs and emerging areas of research in GNN-based fusion approaches, as well as their potential uses in urban development and smart cities[1].

Nantoi et al., examines models in ITSs for real-time traffic flow management, with a focus on decision-making procedures. It encompasses predicting, planning, executing, and regulating techniques for managing traffic flow and reducing congestion. Traffic flow prediction techniques, such as dynamic route guiding and traffic flow prediction, use historical data and real-time inputs to make proactive decisions. Traffic flow planning methods, such as the dynamic route guidance index and the route efficiency factor, help with route selection and signal timing optimization. To simplify the limitless complexity, the authors consider that it is helpful to define the management capacity paradigm of ITSs into two independent scenarios of "stable and known situation" and "unstable and with large uncertainty situation[5]

Chong et al., 2024 investigate the integration of FL in ITSs, with an emphasis on FL's use in TFP, trajectory prediction, space utilization estimate, and target identification. Despite its potential, FL adoption confronts hurdles, including data diversity, communication and bandwidth limitations, and constraints on resources on edge devices [6]. Khalil et al., 2024 provides a comprehensive assessment of DL use in ITS, concentrating especially on practitioners' techniques for addressing these diverse difficulties. The emphasis is on architectural and problem-specific elements that influence the development of innovative solutions. In addition to shedding light on cutting-edge DL algorithms, the study also looks into the potential applications of DL and large language models (LLMs) in ITS, such as TFP, vehicle identification and classification, road condition surveillance, traffic sign recognition, and autonomous automobiles[7]. Many current research may not completely utilize additional data from many models since they depend on conventional fusion procedures like weighted average or voting processes.

III.METHODOLOGY

The next subsections show how to create traffic flow forecasting models based on a ensemble forecasting models using score level fusion from multiple ML models and the overall workflow is shown in figure 1.

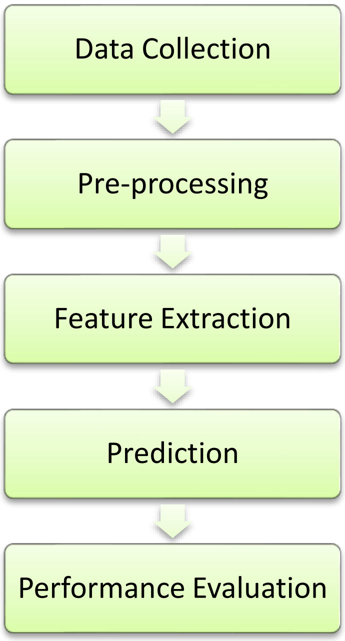

Figure 1 Work flow of Traffic flow prediction

(i) Data Pre-processing

First, collect traffic-related data (for example, historical volume of traffic, the weather, and roadway characteristics). Preprocessing the data includes handling missing values, outlier removal and standardizing the data to scale it uniformly. Partition the collection of data into training and testing sets (e.g., 80/20 split).

(ii) Feature Extraction

It involves collecting representative features from unstructured information and it is an essential step in the development of features for ML. Road traffic flows are impacted by a variety of causes and can have different characteristics and patterns. In order to incorporate these traits and trends into traffic flow models for forecasting, significant elements must be extracted. Mutual Information (MI) is an effective strategy for feature selection and extraction, particularly when anticipating traffic flow. It assesses the quantity of information exchanged between the two factors and assists in determining which attributes are most significant to a target variable (for example, traffic flow). Some of the significant features related TFP are listed in the table 1.

Table 1 Significant Features related to TFP

S.No	Name of the feature	Description
1	Traffic volume	Number of vehicles
2	Weather conditions	Temperature, rain and humidity
3	Time of a day	Hour, day of the week
4	Special occasions	Holidays, accidents

5	Traffic signals states	Red, green. Yellow
6	Speed	Average speed of vehicles
7	Road Conditions	Wet, dry, congested
8	Historical data	Traffic flow of the past periods

The target variable is the TFP and the mutual information is calculated between each feature and the target variable to measure the relationship between them. The formula for MI between two variables X and Y is

$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \qquad (3)$$

In equation (3), p(x,y) is the joint probability of x and y, p(x) and p(y) are the marginal probabilities of x and y. Once the MI values have been determined, rank the features according to their MI with the goal variable (traffic flow). Higher MI scores suggest more important characteristics. Features with low MI values can be removed, lowering the dataset's dimensionality. Choose the top features according to their MI score. One can either select a predetermined number of top characteristics or choose a threshold for the MI score. For example, features having a MI score larger than a given threshold (e.g., 0.1) may be chosen. Then train the ML model such as Random Forest, Naive Bayes, KNN , XGBoost and SVM for traffic flow prediction using the chosen features. Use the reduced feature set to create a more effective framework with potential for improved generalization performance.

(iii) Prediction Models

Prediction models are technologies that estimate future events based on past data. These models evaluate patterns and correlations in the data to create predictions about previously unknown data. Score-level fusion models are assembled by merging the output of various models to increase prediction performance. The objective is to maximize the strengths of several models, usually by combining their forecasts in a way that minimizes their particular shortcomings. The overall design of score level fusion process is presented in figure 2.

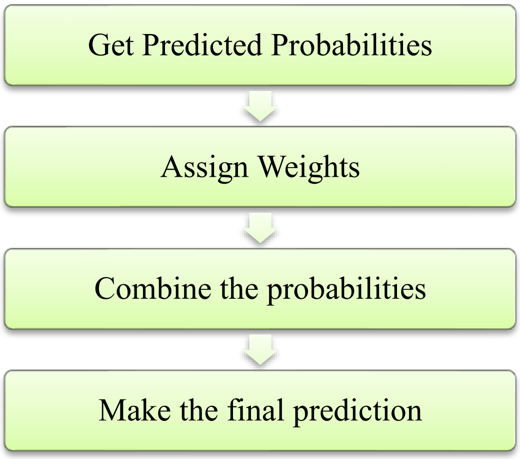
Get Predicted Probabilities
Assign Weights
Combine the probabilities
Make the final prediction

Figure 2 Design of score level fusion

Some of the prediction models used in this study for TFP are discussed in the next section

a) Random Forest

Random Forest is an adaptable method that may be used to anticipate traffic patterns. It creates an ensemble of decision trees and aggregates their outputs, which helps avoid overfitting and boosts the model's resilience[8]. The steps for the Random Forest Algorithm is as follows

- Data: Input Features (X): The variables that are independent utilized in prediction. The target variable (Y) is the dependent variable, which can be continuous or categorical
- Bootstrap Sampling: Create many samples for bootstrapping from the training data. Each sample is formed by picking data points at random and replacing them. Some data points can appear more than once in a sample, whereas other ones may not exist at all.
- Construction of Decision tree
  - For each bootstrap samples, create a decision tree:
  - At each split in the tree, randomly choose a selected group of features (e.g., √N for classification or N/3 for regression, where N is the overall number of features).
    a. Split the data at the node by selecting the best feature from the subset.
    b. Repeat till the tree reaches its deepest level or other stopping conditions (for example, the minimum number of samples per leaf).
  - Ensemble Creation
    a) Repeat steps 2 and 3 to generate a large number of trees (n_estimators). Prediction.
    b) Aggregate forecasts from all trees by majority vote.
    c) Regression involves calculating the average of projections from all trees.

b) Naive Bayes(NB) Model

The NB model is a probabilistic ML approach based on Bayes' theorem, which assumes that characteristics are dependent upon being given a classification label. While Naive Bayes is normally used for classification tasks, it may be modified for TFP, if the problem requires categorical outputs, such as forecasting traffic congestion levels. NB calculates the likelihood of each class based on the input characteristics, then chooses the class with the highest chance.

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \qquad (1)$$

In equation (1) , P(C|X) indicates the posterior probability of class C given input X, P(X|C), represents the likelihood of feature X given class C, P(C) is the prior probability of class C and P(X) is the evidence or marginal probability.

NB classifiers require a set of linear variables that are extremely adaptable to a learning issue. Maximum-likelihood training relies on evaluating a closed-form expression, and requires longer than linear, in contrast to iterative estimation, which is costly and used in many other types of classifiers.

It's used to create classifier models that assign class labels to define problem cases, which can be represented as vectors of features values, with the class labels drawn from a finite set. NB classifiers presume that the value of a specific feature is unaffected by the value of any other characteristic in the given class. In numerous real-world scenarios, the estimation of parameters for NB models uses maximum likelihood; however, the NB model can be performed without the use of Bayesian probability or any other Bayesian process[9]

c) K-Nearest Neighbour(KNN)

The KNN algorithm is a classifier developed using supervised learning that uses proximity to classify or predict the grouping of a data item. It is an instance-based learner that fails to develop a classification model without samples. The main idea of KNN during classification is that individual testing samples are compared locally to k surrounding training data points in variable space, and their category is determined based on the classification of the nearest k neighbors. Neighbors are frequently determined using a Euclidian distance measure between the researched data item and its k neighbors. Predictions are based on a majority vote from surrounding samples. The KNN algorithm operates on the premise of comparable closeness through distance estimates[10]. When constructing a KNN model, the steps are as follows:

- Estimate the number of nearest neighbors, often known as K. For example, if K=2, the two closest spots based on the distance calculation will be picked to figure out where an instance would be allotted. Selecting K can be difficult in a KNN algorithm. Selecting a small K indicates a greater effect on the outcome. On the other side, adopting a larger K may result in a smoother decision border with reduced variance but increased bias. One method for determining K is to train a model with different K neighbors, such as 1, 2, etc., to determine which K will results in the highest testing accuracy
- To determine the distance between every single instance and all samples used for training, use a distance function like Euclidean. Calculate the distance between a new data point and all existing points in the original dataset utilizing a distance measure, like Euclidean distance:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

- Next, order the distances from lowest to tallest. Next, choose the nearest neighbor(s) depending on the number of nearest neighbors (K) chosen in step 1. In other words, pick the K neighbors with the shortest distance.
- Determine the category (classification) or numerical value of the nearest neighbors acquired in step 3.
- To forecast the value or class of an instance, use the majority of its nearest neighbors.

d) XGBoost

XGBoost, or Extreme Gradient Boosting, is a flexible, networked gradient-boosted decision tree (GBDT) ML framework. It offers parallel tree boosting and is the most used ML library for regression, categorization, and ranking tasks. It is a decision-tree(DT)-based ensemble ML technique that employs a gradient-boosting framework. In prediction issues involving unstructured data (pictures, text, etc.), artificial neural networks(ANNs) surpass all existing algorithms or frameworks. However, for small-to-medium structured/tabular data, DT algorithms are now rated best-in-class.

XGBoost creates a predictive model by iteratively merging the predictions of numerous independent models, most often decision trees. The method works by progressively adding weak learners to the ensemble, with each new learner focused on fixing the mistakes caused by the previous ones. It minimizes a given loss function during training using a gradient descent optimization approach. The ensemble's models, also known as base learners, might come from the same or distinct learning methods. Bagging and boosting are two often utilized ensemble learners.

While DTs are one of the easiest models to understand, their behavior is very varied. The single training dataset is randomly divided into two halves. When both of these models are fitted, it provides different outcomes. This tendency causes decision trees to display large variance. Bagging or boosting aggregation helps to decrease variation in all learners. The bagging technique's base learners are composed of many decision trees created in parallel. These learners are trained using data collected through replacement sampling. The final projection is based on the average production of all learners[11,12]

e) Support Vector Machines

Support vector machines, or kernel-based techniques, are used for information classification and data set classification. SVM's robust theoretical statistical basis allows it to operate with thousands of distinct characteristics with ease. An SVM model is primarily determined by the selection of its kernel; hence, it is important to select the proper kernel for every application situation in order to obtain good results. The concept of SVM is designed to deal with complex data classification by addressing the optimization issue and determining the best classifying hyperplane in the multidimensional feature space.

It divides the classes using a decision surface or hyperplane to optimize the margin between them. The data points nearest to the hyperplane are known as support vectors. The support vectors are the key aspects of the training set, therefore the samples used for training do not need to be enormous, but they must include support vectors[13].

(f) Ensemble Method

In this ensemble method RF is utilized as a base model and XGboost is used to enhance its prediction. The main patterns in the data are captured using RF. After then, the training dataset is used to train the RF model, which produces predictions for both the training and validation/test sets. After then, the XGboost receives the RF model's missing residuals for additional improvement. The XGBoost model focuses on the areas where the RF model had trouble and learns to predict the residual errors.

IV. RESULTS AND FINDINGS

The prediction probabilities of the RF, NB, KNN, XGBoost and SVM are listed in table 1. Each row shows the expected probability for a specific sample from the test set. Each algorithm's probability for classes 0 and 1 is displayed in separate columns. The table 2 shows how each approach estimates the likelihood of the sample belonging to a single class.

Table 2 (a) Prediction probabilities of ML models

Sample	RF Class 0	RF Class 1	NB Class 0	NB Class 1	KNN Class 0	KNN class 1
1	0.24	0.76	0.32	0.68	0.45	0.55

Table 2 (b) Prediction probabilities of ML models

Sample	SVM class 0	SVM class1	XGBoost Class 0	XGBoost class 1
1	0.32	0.68	0.25	0.75

From the table 2(a) and (b), it is found that the RF and the XGBoost models scores were higher compared to KNN, SVM and NB. The ensembling of RF and XGBoost is suggested to improve the prediction accuracy[14]. The performance metrics like accuracy, precision, recall and F1-score are as follows

- Accuracy is defined as the fraction of accurately predicted samples.

$$Accuracy = \frac{Number\ of\ correct\ Preditions}{Total\ predcitions} \quad (1)$$

- Precision refers to accurate positive forecasts.

$$Precsion = \frac{True\ Positives(TP)}{True\ Positives(TP)+False\ Postives(FP)} \quad (2)$$

- Recall: The proportion of true positives accurately detected

$$Recall = \frac{True\ Positives(TP)}{True\ Positives(TP)+False\ Negatives(FN)} (3)$$

- F1-Score: The harmonic mean of accuracy and recall.

$$F1 - score = 2 * \frac{Precision*Recall}{Precision\ +Recall} \quad (4)$$

The table 3 shows the performance analysis of RF , NB, KNN, XGBoost , SVM and the proposed score level fusion of RF and XGBoost.

Table 3 Performance Measures

Methods/ Measures	Accuracy	Precision	Recall	F1-Score
RF	0.93	0.92	0.91	0.91
NB	0.81	0.82	0.81	0.80
KNN	0.86	0.84	0.04	0.83
XGboost	0.90	0.91	0.88	0.87
SVM	0.88	0.89	0.87	0.87
RF-XGBoost	0.95	0.93	0.92	0.92

From table 3, it is seen that the accuracy is 0.95 and it is found that the RF-XGBoost leads individual models in all measures, highlighting the strength of ensembling. Both RF and XGBoost perform similarly well and are strong individual models. Precision for ensemble method is 0.93, which is best at reducing false positives and the recall is 0.92 which captures more true positives. . The F1-score of 0.92 provides the best balance between precision and recall. The RF-XGBoost combination achieves the best results on all measures by utilizing the advantages of both RF and XGBoost.

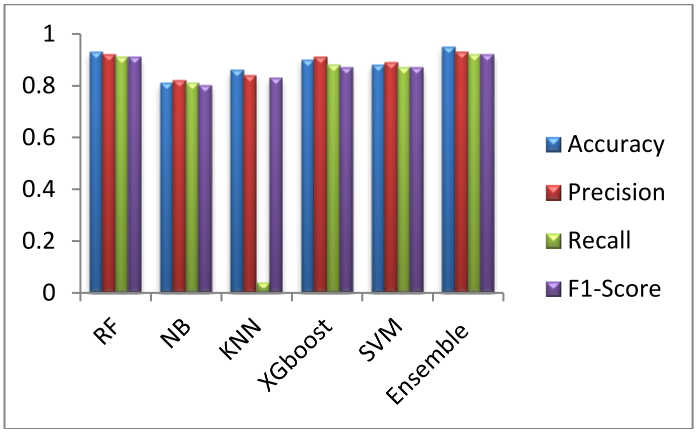

Figure 3 Metrics vs methods

From figure 3 , it is clear that the ensemble learning shows consistently greater performance in terms of accuracy, Precision, recall and F1-score RF and XGBoost beat other models across all measures. SVM outperforms KNN and Naive Bayes, but falls slightly behind RF and XGBoost. Naive Bayes had the lowest ratings because of simple and feature independence assumptions.

Moreover, RF is reliable and effective in reducing volatility and capturing general patterns. It is capable of effectively processing high-dimensional and noisy data. It can manage residual distribution imbalances and is very good at learning intricate patterns. Error is repeatedly reduced by its boosting structure. Although RF + XGBoost ensembles offer notable improvements in accuracy, their scalability is dependent on the use case and the resources at hand. Implementing such models at scale requires optimizing memory and computational efficiency.

IV. CONCLUSION

The study examines the various ML models for TFP and suggests a score level fusion approach using RF –XGBoost. The findings show that the RF-XGBoost outperforms with the accuracy of 95% than the RF, NB, SVM, KNN and XGBoost as single models. The ensemble model with score-level fusion improves TFP for ITS. The model delivers greater performance by combining the complementing characteristics of RF and XGBoost resulting in more efficient transportation systems. When compared to individual models and standard ensemble approaches, the suggested model outperformed the latter in terms of accuracy, precision, recall, and F1 score. Furthermore, the use of mutual information-based feature selection decreased computing complexity while keeping the most relevant traits for traffic flow prediction. Future research will concentrate on expanding the ensemble model to incorporate real-time data that is streaming, allowing for dynamic adjustments to traffic estimates as more information becomes available. This increases reactivity and flexibility in real-world ITS contexts. A common perception of ensemble models is that they are black-box systems. Using methods like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), future research may concentrate on enhancing interpretability.

REFERENCES

[1]. Ahmed, S. F., Kuldeep, S. A., Rafa, S. J., Fazal, J., Hoque, M., Liu, G., & Gandomi, A. H. (2024). Enhancement of traffic forecasting through graph neural network-based information fusion techniques.

[2]. Reddy, K. H. K., Goswami, R. S., & Roy, D. S. (2024). A deep learning-based smart service model for context-aware intelligent transportation system. The Journal of Supercomputing, 80(4), 4477-4499.

[3]. Ou, J., Li, J., Wang, C., Wang, Y., & Nie, Q. (2024). Building trust for traffic flow forecasting components in intelligent transportation systems via interpretable ensemble learning. Digital Transportation and Safety, 3(3), 126-143.

[4]. Nagarajan, S. M., Devarajan, G. G., Ramana, T. V., Bashir, A. K., & Al-Otaibi, Y. D. (2024). Adversarial Deep Learning based Dampster–Shafer data fusion model for intelligent transportation system. Information Fusion, 102, 102050.

[5]. Nantoi, V., Nantoi, D., & Pădure, O. (2024). Models for real-time traffic flow manageability and decision-making in intelligent transportation systems. Journal of Social Sciences, (3), 35-70.

[6]. Chong, Y. W., Yau, K. L. A., Ibrahim, N. F., Rahim, S. K. A., Keoh, S. L., & Basuki, A. (2024). Federated Learning for Intelligent Transportation Systems: Use Cases, Open Challenges, and Opportunities. IEEE Intelligent Transportation Systems Magazine.

[7]. R. A. Khalil, Z. Safelnasr, N. Yemane, M. Kedir, A. Shafiqurrahman and N. SAEED, "Advanced Learning Technologies for Intelligent Transportation Systems: Prospects and Challenges," in *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 397-427, 2024, doi: 10.1109/OJVT.2024.3369691

[8]. Sun, S., Yan, H., & Lang, Z. (2024). A study on traffic congestion prediction based on random forest model. Highlights in Science, Engineering and Technology, 101, 738-749.

[9]. Hassan, M., & Arabiat, A. (2024). An evaluation of multiple classifiers for traffic congestion prediction in Jordan. Indonesian Journal of Electrical Engineering and Computer Science, 36(1), 461-468.

[10]. Mrudula, S. T., Ritonga, M., Sivakumar, S., Jawarneh, M., Sammy, F., Keerthika, T., ... & Roy, B. (2024). Internet of things and optimized knn based intelligent transportation system for traffic flow prediction in smart cities. Measurement: Sensors, 35, 101297.

[11]. Yu, W., & Xie, F. (2024). Research on Traffic Congestion Prediction Based on XGBoost. Frontiers in Traffic and Transportation Engineering, 4(1), 1-8.

[12]. Jin, T., Zhang, Z., & Liu, B. (2024). Machine learning advancements in traffic forecasting: hybrid optimization of LS-SVM for urban traffic management. Advances in Transportation Studies, 62.

[13]. Sun, Z. M., Ren, G., Hu, Y. X., & Lin, H. (2023, September). Improved traffic flow estimation based on integrated learning methods. In Eighth International Conference on Electromechanical Control Technology and Transportation (ICECTT 2023) (Vol. 12790, pp. 500-507). SPIE.

[14]. Rani, P., & Sharma, R. (2023). Intelligent transportation system for internet of vehicles based vehicular networks for smart cities. Computers and Electrical Engineering, 105, 108543.