Proceedings of the 3rd International Conference on Mechatronics and Smart Systems
DOI: 10.54254/2755-2721/135/2025.21218

Research on Traffic Prediction Based on Machine Learning

Sujue Deng[1,a,*]

[1]*School of Computer Science and Technology, Tiangong University, Tianjin, China*
a. 2211640211@tiangong.edu.cn
**corresponding author*

Abstract: With the rapid acceleration of urbanization, traffic congestion has become an increasingly serious challenge for cities worldwide, impacting both economic productivity and the quality of life for residents. Efficient traffic management is essential to alleviate congestion and optimize the use of infrastructure. Traffic forecasting, which involves predicting traffic flow and congestion patterns, plays a pivotal role in enhancing traffic management systems and facilitating better decision-making. This study explores the application of machine learning techniques alongside traditional statistical methods to predict traffic flow. The experimental results show that machine learning methods offer significant improvements in forecasting accuracy, providing more reliable predictions of traffic conditions compared to conventional approaches. The findings underscore the potential of these advanced methods to improve traffic management strategies, optimize resource allocation, and reduce congestion, ultimately leading to more sustainable urban transportation systems. These insights are valuable for urban planners and policymakers looking to implement data-driven solutions to address the growing challenges of urban mobility.

Keywords: Machine Learning, Probability, Prediction.

1. Introduction

As the economy develops, technology advances and urbanization accelerates, traffic congestion is increasingly affecting the lives of city residents, and this problem is becoming more prominent [1]. The importance of traffic forecasting is growing day by day. Traffic forecasting refers to predicting changes in traffic flow at a specific time or area by analyzing historical traffic data, real-time monitoring information, and various factors in the traffic network [2]. It helps traffic management departments optimize scheduling, plan infrastructure, reduce congestion, improve traffic efficiency, optimize resource allocation, and provide drivers with timely traffic information and navigation advice.

In recent years, with the development of artificial intelligence technology, especially the continuous maturation of machine learning techniques, Traditional statistical methods (such as ARMA, ARIMA, and SARIMA) can no longer meet the requirements [3].

An increasing number of models have been developed and applied to traffic forecasting [4]. Graph convolutional networks (GCN) have become a very popular approach, as they are capable of capturing the complex relationships and structural information between nodes in a graph and handling non-Euclidean data [2]. Ensemble learning methods [5], such as eXtreme Gradient Boosting (XGBoost), enhance predictive ability and generalization performance by combining multiple

DOI: 10.54254/2755-2721/135/2025.21218

decision tree (DT) models [5]. Probabilistic models, such as Bayesian networks, abstract the dependency relationships of traffic flow between multiple lanes and intersections into graph models and infer the flow distribution at each moment from historical data [6]. In addition to machine learning methods, traditional traffic flow prediction methods mainly rely on statistical models and empirical formulas, such as Logistic Regression (LR), Gaussian Naive Bayes (GNB), and Ridge Classifier (Ridge). Among them, time series models, like Seasonal Autoregressive Integrated Moving Average (SARIMA), are based on sequences formed over time [7]. This model depends on the variation of data over time, but it may not perform well with complex non-stationary data. Models like Autoregressive Integrated Moving Average (ARIMA) [8], which combine autoregression, differencing, and moving averages, capture the trend, seasonality, and random fluctuations in time series data. Regression methods, such as vector autoregression, can also be used to capture the linear dependencies between multiple time series variables. These methods model historical data to predict future traffic flow [9].

However, the above methods are time-consuming and prone to overfitting, often performing poorly in long-term forecasting tasks [10]. This study employs machine learning techniques to explore the performance of different models in the field of traffic forecasting. For data that follows a linear distribution pattern and has a small sample size, the Synthetic Minority Over-sampling Technique (SMOTE) method is introduced to increase the sample data and shuffle the dataset. Meanwhile, robustness and generalization tests are incorporated to enhance the model's learning ability when faced with data variations. Additionally, regularization techniques are used to determine whether the added data causes overfitting in the model.

The first section of this paper introduces relevant information, the second section presents the data used in the experiments, the third section discusses the experimental results, the fourth section outlines the limitations of the methods, and the fifth section presents the conclusions drawn from the experimental findings.

2. Data and Method

2.1. Data Collection and Description

The dataset specifically includes the Time column, which represents the exact time of the records, the Day of the week column, which indicates the day of the week, the columns for the number of different types of vehicles, the Total column for the total vehicle count, as well as the corresponding traffic conditions, as shown in Table 1.

Table 1: Datasets.

Time	Date	Day of the week	Car Count	Bike Count	BusCount	TruckCount	Total	Traffic Situation
12:00:00 AM	10	Tuesday	31	0	4	4	39	Low
12:15:00 AM	10	Tuesday	49	0	3	3	55	Low
12:30:00 AM	10	Tuesday	46	0	3	6	55	Low
12:45:00AM	10	Tuesday	51	0	2	5	58	Low
1:00:00 AM	10	Tuesday	57	6	15	16	94	normal
10:45:00 PM	9	Thursday	16	3	1	36	56	normal

DOI: 10.54254/2755-2721/135/2025.21218

Table 1: (continued).

11:00:00 PM	9	Thursday	11	0	1	30	42	normal
11:15:00 PM	9	Thursday	15	4	1	25	45	normal
11:30:00 PM	9	Thursday	16	5	0	27	48	normal
11:45:00 PM	9	Thursday	14	3	1	15	33	normal

2.2. Model

eXtreme Gradient Boosting (XGBoost) is an efficient implementation of Gradient Boosting Decision Trees (GBDT) that can handle missing values, nonlinear relationships, and large-scale data. It is widely used in various machine learning competitions.

Random Forest (RF) is an ensemble model based on decision trees. It builds multiple trees through random sampling of data and features, then aggregates their predictions by voting or averaging, providing strong robustness and resistance to overfitting.

Extreme Randomized Trees (ERT) are similar to RF but add more randomness by selecting split points completely randomly, which speeds up training and improves generalization.

K-Nearest Neighbors (KNN) is an instance-based model that predicts the class of new data by calculating the distance (e.g., Euclidean distance) between samples. It works well for small datasets but performs poorly with high-dimensional data.

LR is a linear model commonly used for binary classification problems. It outputs a probability value and is suitable for linearly separable data.

Ridge is a linear model with regularization. It uses L2 regularization to constrain model complexity, thus improving robustness and preventing overfitting.

Histogram-based Gradient Boosting Classifier (HBC) is a powerful model suitable for large-scale data. It improves training efficiency using histogram binning techniques and is applicable for both classification and regression tasks.

Bootstrap Aggregating (Bagging) is a parallel ensemble learning method that performs multiple bootstrap resampling of the dataset to train several base classifiers, then averages or votes on their predictions to enhance stability and accuracy.

Gradient Boosting Trees (GBDT) is an iterative decision tree ensemble model that optimizes model performance by progressively minimizing a loss function, making it suitable for complex datasets.

GNB is a probability-based classification method that assumes features are independent and follow a Gaussian distribution. It works well with simple, small datasets.

Support Vector Classification (SVC) is a powerful classification model, particularly suited for high-dimensional data. It can handle non-linearly separable data using kernel functions (such as linear or RBF kernels).

3. Results and Discussion

3.1. Experimental configuration

The training and testing sets are split from the original dataset at a ratio of 9:1. Regularization is applied only to XGBoost, HBC, Bagging, and GBC, with both XGBoost and HBC having a learning

DOI: 10.54254/2755-2721/135/2025.21218

rate of 0.001, while GBC has a learning rate of 0.005. The models all use the default loss functions; XGBoost uses the logistic loss function (also called log loss), and HBC uses categorical cross-entropy, which is a loss function used for classification tasks. Its form is similar to log loss and measures the difference between the model's predicted probability distribution and the true distribution. Bagging does not directly use a loss function. Instead, it generates the final prediction by voting or averaging the predictions of multiple base learners (by default, decision trees). GBC, like XGBoost, uses the log loss function.

3.2. Experimental result

As shown in Table 2, XGBoost, RF, HBC, Bagging, and GBC achieved an accuracy close to or reaching 100% under normal conditions. Even with noise interference, the accuracy only slightly decreased, maintaining between 87% and 91%, demonstrating strong robustness. These models also showed good generalization ability in the generalization tests. SVC, ERT, and KNN had accuracy close to 90% under normal conditions and performed stably in both noise and generalization tests, especially SVC and ERT, which showed strong stability. GNB performed the worst, with a significant drop in accuracy under noise interference and generalization tests, indicating weak resistance to disturbance and poor generalization ability, making it unsuitable for complex tasks in real-world scenarios. Overall, XGBoost, RF, HBC, Bagging, and GBC performed the best in terms of robustness and generalization, while other models like Ridge, GNB, and DC have more limited applicability. The details are shown in Table 3, Table 4 and Table 5.

Table 2: Default Accuracy.

XGBoost	1.00(+/-0.00)
RF	1.00(+/-0.00)
ERT	0.97(+/-0.01)
KN	0.93(+/-0.01)
LG	0.90(+/-0.02)
Ridge	0.77(+/-0.01)
HBC	1.00(+/-0.00)
Bagging	1.00(+/-0.00)
GBC	1.00(+/-0.00)
GNB	0.81(+/-0.02)
SVC	0.94((+/-0.01)

Table 3: Accuracy under Noise.

XGBoost	0.95(+/-0.01)
RF	0.94(+/-0.01)
ERT	0.94(+/-0.00)
KN	0.93(+/-0.01)
LG	0.83(+/-0.01)
Ridge	0.70(+/-0.01)
HBC	0.94(+/-0.01)
Bagging	0.94(+/-0.01)
GBC	0.94(+/-0.01)
GNB	0.78(+/-0.01)
SVC	0.92(+/-0.01)

DOI: 10.54254/2755-2721/135/2025.21218

Table 4: Generalization Accuracy.

XGBoost	1.00(+/-0.00)
RF	0.99(+/-0.00)
ERT	0.98(+/-0.00)
KN	0.96(+/-0.01)
LG	0.84(+/-0.02)
Ridge	0.73(+/-0.02)
HBC	1.00(+/-0.00)
Bagging	1.00(+/-0.00)
GBC	1.00(+/-0.00)
GNB	0.78(+/-0.01)
SVC	0.94(+/-0.01)

Table 5: Regularization Accuracy.

XGBoost	0.98(+/-0.00)
HBC	0.99(+/-0.00)
Bagging	1.00(+/-0.00)
GBC	0.95(+/-0.01)

By controlling the sub-sample ratio, among other factors, the generalization ability of the model is improved. Cross-validation (cross_val_score) was then used to evaluate the performance of each model on the training data, outputting the average accuracy and standard deviation for each model (as shown in Figure 1), to compare their performance and stability.

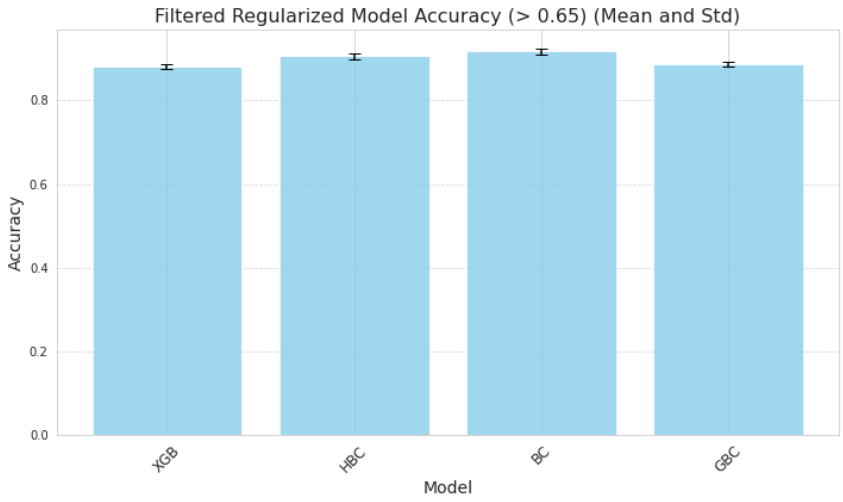

Figure 1: Accuracy with Regularization (Picture credit: Original).

From the learning curves of these four models (as shown in Figure 2, Figure 3, Figure 4, and Figure 5), it can be seen that, overall, overfitting is not significant. XGBoost may exhibit slight overfitting, as the training accuracy is close to 1, while the testing accuracy is slightly lower, with a noticeable gap between the two. However, as the training data increases, the performance on the test set gradually improves. Bagging shows a small difference between the training and testing accuracy, and the curve tends to stabilize, demonstrating good generalization ability. The training accuracy of GBC is slightly higher than the testing accuracy, but the gap is not large, and the test performance improves

DOI: 10.54254/2755-2721/135/2025.21218

as the data increases, with no obvious overfitting. The training and testing accuracies of HBC almost overlap, indicating strong generalization ability and no overfitting issues.

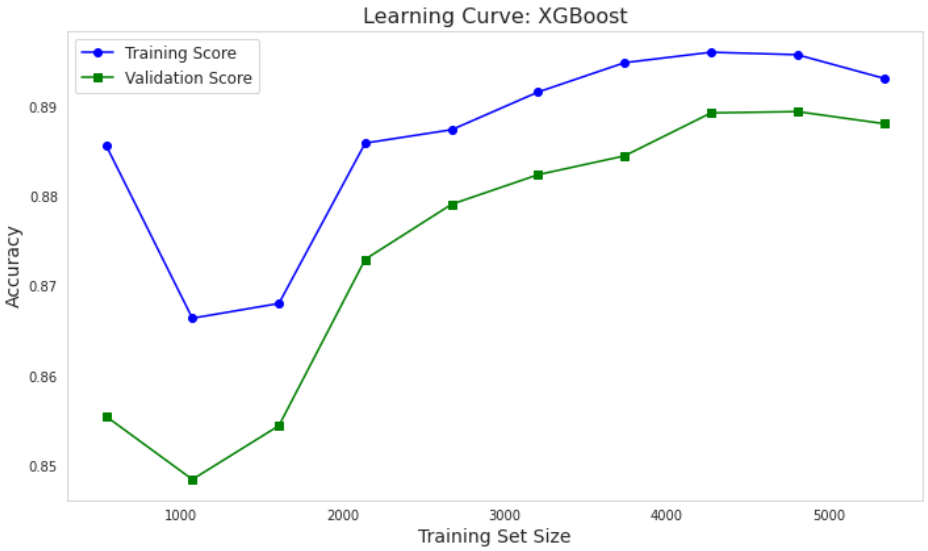

Figure 2: XGBoost Learning Curve (Picture credit: Original).

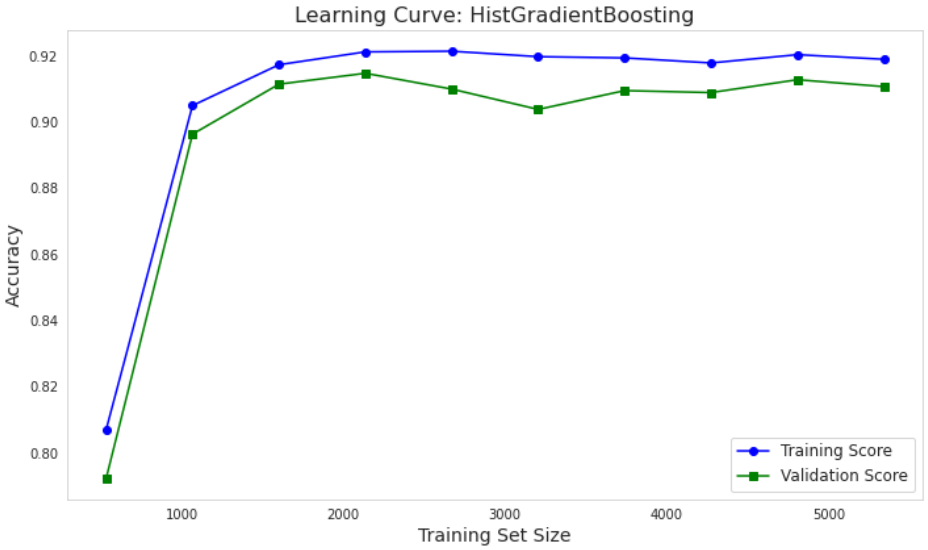

Figure 3: HGB Learning Curve (Picture credit: Original).

DOI: 10.54254/2755-2721/135/2025.21218

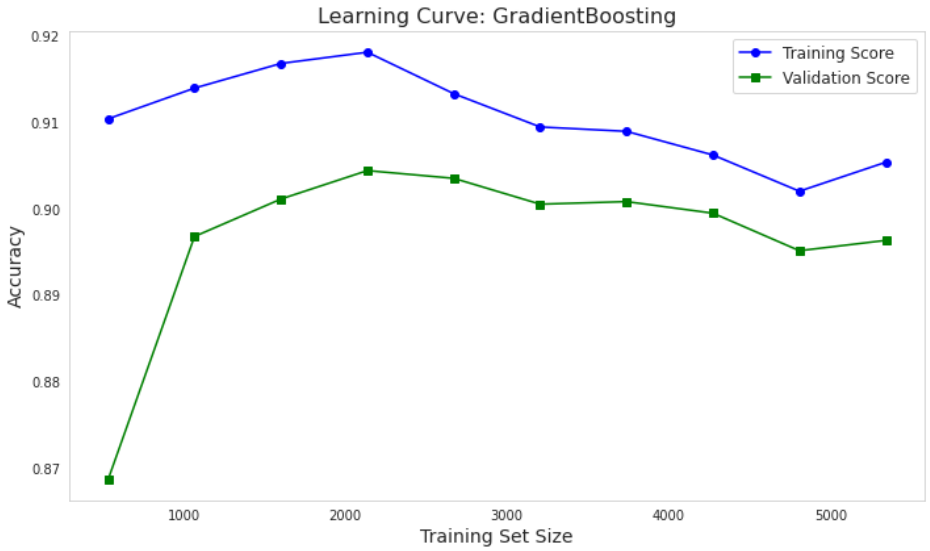

Figure 4: GBC Learning Curve. (Picture credit: Original).

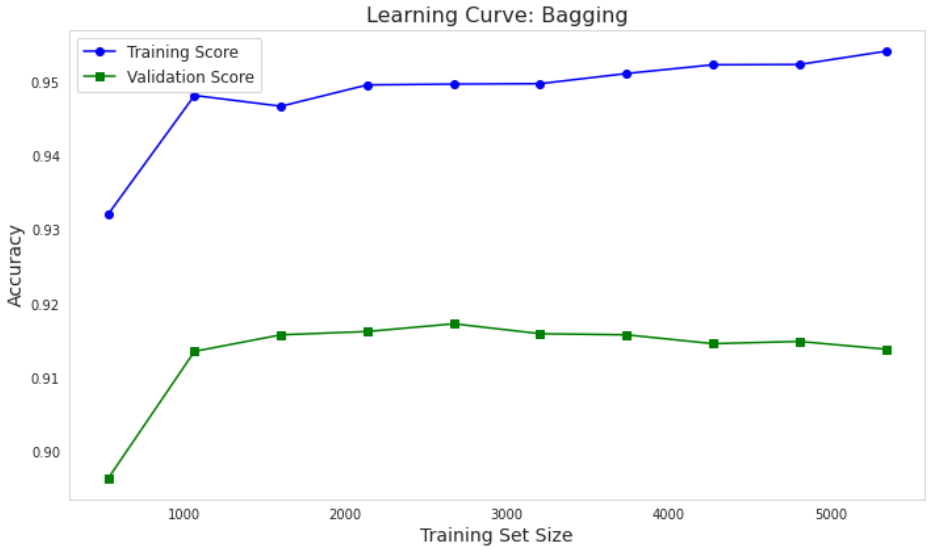

Figure 5: Bagging Learning Curve (Picture credit: Original).

Regularization significantly improved the robustness and generalization ability of the models. Under both default and noisy conditions, the accuracy of the XGBoost, HBC, Bagging, and GBC models was relatively high, but some models, such as GNB, exhibited instability under noisy conditions. However, after regularization, the accuracy of the models further improved in the generalization test, with a significant reduction in variability (standard deviation). Notably, the Bagging model achieved an accuracy of 1.00, showing the best performance. This indicates that regularization enhances the stability of the model when dealing with noise and distribution changes, and the performance of different models highlights the importance of selecting the appropriate model and regularization method. Among them, Bagging, XGBoost, and HBC demonstrated good performance and balance in this experiment.

DOI: 10.54254/2755-2721/135/2025.21218

4. Limitations and Future outlooks

The results of this experiment are based solely on the dataset used, and further investigation into how the model's performance may change with different datasets has not been conducted. Since different datasets have distinct features and structures, changing the dataset could lead to variations in the model's performance during training. The models in the experiment were used with their default hyperparameter configurations, without any hyperparameter tuning. In fact, many machine learning models have a variety of hyperparameters, and the choice of these hyperparameters can significantly impact the model's performance. However, this experiment did not adjust or optimize these hyperparameters, and all models used the default settings. In practical applications, hyperparameter tuning could further improve the model's accuracy and stability.

Although this experiment has demonstrated the performance of different models on the current dataset, future research could optimize the models to enhance their performance and generalization ability. Firstly, the dataset used in the experiment has certain limitations, and more diverse datasets could be introduced in the future. Secondly, hyperparameter tuning is one of the key factors for improving model performance, and optimizing the model's hyperparameters could significantly increase accuracy. In terms of model selection, deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown excellent performance in handling complex tasks, and their application in this experimental task could be explored in the future. With these improvements, future research is likely to further enhance the model's performance and application effectiveness, making it more reliable and efficient in real-world scenarios.

5. Conclusion

This experiment analyzed the limitations of different models in traffic flow prediction and improved their performance through several measures. First, increasing the number of data samples and shuffling the dataset helped enhance the model's learning ability, especially in cases of data imbalance or scarce samples. This approach effectively boosted the model's generalization ability. Second, by incorporating robustness and generalization tests, the model's stability was further improved when facing data variations and noise, ensuring its reliability in different environments. Finally, introducing regularization methods and analyzing the learning curves helped identify and prevent overfitting, thereby improving the model's generalization capability and adaptability to unseen data. This study highlights the tremendous potential of machine learning and deep learning in the field of traffic prediction. In the future, optimizing the model's hyperparameter settings could further improve performance, providing more accurate and efficient predictive tools for practical applications.

References

[1] *Wang, Q., Lu, Q., & Shi, P. (2024). Multi-graph diffusion attention network for traffic flow prediction. Computer Applications, 1-10.*

[2] *Li, Q., Zhu, K., Li, B., & Zhao, C. (2024). Traffic flow prediction method based on spatiotemporal dependency adaptive fuzzy embedding. Control Engineering.*

[3] *Ma, F., Zou, F., Liao, L., Luo, Y., Hu, Z., & Chen, W. (2024). Toll station online traffic flow prediction model based on temporal features. Journal of Shaanxi University of Science and Technology, 1-11. https://doi.org/10.19481/j.cnki.issn2096-398x.20241204.001.*

[4] *Tang, J., Zhang, Y., Zou, X., & Qiu, L. (2024). Traffic flow prediction based on attention mechanism and spatiotemporal graph convolutional network. Journal of Kunming University of Science and Technology (Natural Science Edition), 1-9. https://doi.org/10.16112/j.cnki.53-1223/n.2025.01.482.*

[5] *Li, X., Cao, K., & Kuang, H. (2024). Study on travel mode choice based on hyperparameter optimization ensemble learning. Journal of Traffic and Transportation Engineering and Information, 1-13.*

DOI: 10.54254/2755-2721/135/2025.21218

[6] *Wang, T., Yang, G., & Ouyang, M. (2024). Short-term traffic prediction using Bayesian networks with multiple residual compensations. Journal of Harbin Engineering University, 45(09), 1810-1817.*
[7] *Cui, J. (2024). Research on intelligent traffic flow prediction model based on deep learning and big data analysis. Research on Informatization, 50(03), 16-22.*
[8] *Yu, L., Shao, Z., Xu, C., et al. (2024). Short-term traffic flow prediction based on seasonal ARIMA model. Traffic World, (25), 2-5.*
[9] *Zhang, W., Liu, R., Zhang, H., et al. (2023). Short-term OD prediction for urban rail transit based on vector autoregression and dynamic mode decomposition. Journal of Beijing Jiaotong University, 47(06), 41-49.*
[10] *Lü, Q. (2024). Research on multi-scale traffic flow prediction of urban road network based on multidimensional data mining. Microcomputer Applications, 40(11), 289-293.*