Dean&Francis
ISSN 2959-6157

Traffic Congestion Level Prediction: A Comparative Study Based on Random Forest and XGBoost Models

Jincheng Yaochen

Dimensions International College, 238318, Singapore, Singapore
yaojincheng0213@gmail.com

Abstract:

With the acceleration of global urbanization, the problem of traffic congestion has become increasingly severe, exerting a significant impact on economic development, environmental governance, and the management of residents. To alleviate the problem of traffic congestion, this study focuses on the performance of the random forest and XGBoost models in predicting traffic congestion levels. In the data preprocessing stage, this study cleaned and encoded key features such as traffic speed and energy consumption, and constructed interaction variables through feature engineering to enhance the model's ability to capture traffic patterns. During the training phase of the model, this study combined multiple hyperparameter optimization methods, including GridSearchCV, RandomizedSearchCV, Optuna, and BayesSearchCV, for tuning and conducted a comparative analysis. The experimental results show that the optimized models have significant improvements compared to the Baseline. The combined accuracy of random forest + BayesSearchCV has increased from 79.92% to 97.52%, F1-macro has reached 0.9664, and F1-weighted has reached 0.9753. The combined accuracy rate of XGBoost + Optuna increased from 78.56% to 97.6%. The F1-macro score reached 0.9719, and the F1-weighted score reached 0.9761. This study further demonstrates the application potential of machine learning methods in the field of traffic prediction and clarifies a feasible direction for the construction of smart city transportation systems.

Keywords: Random Forest; XGBoost; Traffic Congestion; Hyperparameter Tuning; Machine Learning.

ISSN 2959-6157

1.Introduction

The process of global urbanization is constantly advancing. By 2025, the global urban population is expected to have exceeded 55% of the total population, and it is expected to further rise to approximately 68% in the coming decades [1]. With the continuous increase in the urban population, the impact of traffic congestion cannot be underestimated. With the continuous increase in urban population, the impact of traffic congestion cannot be underestimated. Take New York City, for example. In 2023, on average, each driver lost 101 hours due to traffic congestion, resulting in an annual economic loss of over $ 9.1 billion to the region. This highlights the severe constraints that traffic congestion imposes on social and operational efficiency, as well as economic development [2]. Meanwhile, idling and frequent acceleration and deceleration of vehicles during congestion will also increase fuel consumption and exhaust emissions, thereby deteriorate the city's air quality and posing a threat to the health of its residents [3]. Therefore, how to predict the traffic congestion level efficiently and accurately with the help of advanced algorithms has become an important issue that needs to be solved urgently.

In terms of research methods, machine learning is widely applied in traffic state prediction due to its ability to capture complex nonlinear relationships and handle high-dimensional data. Existing studies have shown that congestion level classification based on feature engineering and tree models has good applicability. For instance, the congestion prediction framework proposed by Zafar et al., which combines the estimated arrival time with features such as holidays and weather, demonstrated stable prediction performance in the classification task of five-level congestion labels [4]. In the field of traffic safety and operation management, gradient boosting methods, by introducing regularization and second-order optimization, not only accelerate the convergence speed of the model and enhance its generalization ability, but also improve the interpretability of the results with the help of interpretive tools such as SHAP, thus facilitating practical deployment [5]. In signal control scenarios, combining XGBoost with recurrent neural networks can capture more fine-grained periodic traffic fluctuations and effectively improve the accuracy of short-term traffic prediction at intersections [6]. In addition, using BayesSearchCV to perform hyperparameter tuning on XGBoost can achieve higher prediction accuracy in complex road networks and high-dimensional feature Spaces while ensuring high efficiency [7].

Some review studies have also provided a systematic understanding of this field. Boukerche and Coutinho conducted a comprehensive review of intelligent traffic prediction models based on machine learning, sorting out the development trajectory from traditional regression and tree models to graph neural networks and spatio-temporal deep learning methods [8]. From the perspective of environmental externalities, Pang et al. found a significant coupling relationship among fuel prices, traffic congestion, and carbon emissions, indicating that the governance of congestion is not only related to travel efficiency but also an important way to achieve urban emission reduction targets [9]. Furthermore, Vlahogianni et al.'s review of short-term traffic prediction methods emphasizes that establishing a unified evaluation index system and comparable experimental framework is the key to promoting further research development [10]. At the data level, new data sources such as floating vehicles (FCD/GPS) have become an important support for congestion level discrimination. Classification research based on GPS shows that by combining supervised learning methods such as random forests, the congestion status of roads under different conditions can be effectively identified, providing technical support for dynamic traffic management and travel information services [11].

Based on the above background, this paper will focus on the multi-classification problem of predicting the congestion level of urban roads. In this study, Random Forest (RF) and XGBoost will be selected as the core baseline models, and the impact of different hyperparameter optimization strategies on model performance will be compared under a unified data and evaluation framework. The aim is to explore efficient and scalable model combination schemes, providing solid data support and practical references for urban intelligent transportation management and policy-making.

2.Methods

2.1 Data pre-processing

This paper conducts experiments using the public dataset "Smart Mobility new" on GitHub, which contains a total of 5,000 samples and fourteen feature words. To ensure the learning effect of the model, the experiment adopted several steps for data preprocessing.

Firstly, in this experiment, features that were not directly related to the prediction task were eliminated, such as Latitude, Longitude, Timestamp, and Sentiment Score. Although these variables can reflect the characteristics of geographical location and social emotions, they have a weak relationship with the traffic congestion level in this experiment and may introduce noise. Secondly, the values were normalized in this experiment. Indicators such as Traffic Speed, Road Occupancy Rate, Emission Level, and Energy Consumption were wrongly stored as large integers separated by dot numbers or a percentage string.

In this regard, this experiment converts it into reasonable floating-point data through string replacement and type conversion. For categorical variables, such as Weather Condition and Traffic Light Status, this experiment uses the method of Label Encoding to convert the text labels into integer encodings, enabling them to be directly used by the model. For missing values, in this experiment, all abnormal entries that cannot be converted into numerical values are uniformly marked as NaN, and all missing samples are eliminated in subsequent steps to ensure the integrity of the input matrix during the model training process. In addition, this experiment introduced interactive features, such as the combination of the number of vehicles and accident reports, to construct the index of "the number of accident-related vehicles", as well as interactive terms like "speed-energy consumption" and "speed-accident", in order to enhance the expressive ability of the model. Finally, in this experiment, the numerical features were standardized, with the features scaled to zero mean and unit variance, avoiding the interference of different dimensions on the model and ensuring that the tree model and gradient method are more stable at the numerical scale.

2.2 Model Design

2.2.1 Random Forest

Random forest is an ensemble classification method based on Bagging (self-sampling Ensemble). It builds several different training subsets by performing multiple self-sampling (Bootstrap) with pullback on the original training set, and independently trains a classification decision tree on each subset. During the node division process of the tree, only a portion is randomly selected from all the features each time as candidate features for splitting, thereby enhancing the diversity of the model and reducing the risk of overfitting. In the prediction stage, the random forest integrates the classification results of all trees and ultimately outputs the category that appears most frequently in the voting results as the predicted value.

2.2.2 XGBoost

XGBoost is a classification algorithm based on the idea of gradient boosting, which continuously optimizes the model through an iterative approach. During the training process, in each round, a new classification tree is added based on the prediction results of the existing model to fit the current prediction error, thereby gradually improving the overall classification accuracy. Unlike traditional gradient boosting methods, XGBoost introduces an additional regularization term in the objective function to control model complexity and prevent overfitting. Meanwhile, it utilizes the first and second derivatives of the loss function for approximate optimization, enabling parameter updates and node divisions to obtain more efficient analytical solutions, thereby accelerating the convergence speed and enhancing stability.

2.3 Evaluation Indexes

2.3.1 Accuracy

Accuracy is the most intuitive classification evaluation metric, used to measure the overall proportion of correct predictions made by the model. Its calculation formula is shown in (1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Among them, True Positive (TP) is the True positive class, True Negative (TN) is the true Negative class, False Positive (FP) is the False positive class, and False Negative (FN) is the false negative class.

2.3.2 Average macro F1 score (F1-macro)

F1-macro is a commonly used comprehensive indicator in multi-classification problems. Its calculation process is to first calculate the F1 score of each category separately, and then take the arithmetic mean of the F1 values of all categories. Its calculation formula is shown in (2):

$$F1_{macro} = \frac{1}{N}\sum_{i=1}^{N} F1_i \tag{2}$$

Here, N represents the number of categories and the F1 score of the i-th category. The F1 value of each category is calculated based on Precision and Recall. Their calculation formulas are as shown in (3):

$$F1_i = \frac{2 \bullet P_i \bullet R_i}{P_i + R_i},\ P_i = \frac{TP_i}{TP_i + FP_i},\ R_i = \frac{TP_i}{TP_i + FN_i} \tag{3}$$

F1-macro treats all categories equally and is suitable for evaluating the overall performance of the model on various samples.

2.3.3 Weighted F1 score (F1-weighted)

The main difference between F1-weighted and F1-macro lies in the weighting method. It also first calculates the F1 score of each category, but when taking the average, it will be weighted according to the number of samples of each category. Its calculation formula is shown in (4):

$$F1_{weighted} = \frac{1}{M}\sum_{i=1}^{N} n_i \bullet F1_i,\ M = \sum_{i=1}^{N} n_i \tag{4}$$

Among them is the number of samples in the i-th category, and M is the total number of samples. Through weighting, F1-weighted can more truly reflect the overall performance of the model in the case of category imbalance.

ISSN 2959-6157

3. Experimental results and comparisons

This experiment is based on the random forest and XGBoost models. Four different hyperparameter tuning strategies, namely GridSearchCV, RandomizedSearchCV, Optuna, and BayesSearchCV, are introduced for parameter tuning, generating a total of eight complete models. This experiment not only employed the evaluation metric of Accuracy, but also introduced the comparison between the macro average F1 score (F1-macro) and the weighted F1 score (F1-weighted), comprehensively measuring the prediction performance of each model under the imbalanced situation of samples in different categories. Table 1 shows the comparison chart of the experimental results of each model.

Table 1. Performance Comparison between Random Forest and XGBoost

Parameter adjustment method	RF_Accuracy	RF_F1_macro	RF_F1_weighted	XGB_Accuracy	XGB_F1_macro	XGB_F1_weighted
Baseline	79.92%			78.56%		
GridSearchCV	97.52%	0.9664	0.9753	97.28%	0.9663	0.973
RandomizedSearchCV	97.2%	0.9653	0.9721	97.28%	0.9693	0.973
Optuna	97.36%	0.9666	0.9737	97.6%	0.9719	0.9761
BayesSearchCV	97.84%	0.9705	0.9785	97.6%	0.9716	0.9761

As shown in Table 1, on the unoptimized Baseline, both Random Forest and XGBoost performed poorly, with accuracy rates of 79.92% and 78.56% respectively. This indicates that the model with default parameters failed to fully capture the complex patterns of traffic data. However, after introducing four parameter adjustment methods, the accuracy of predictions generally increased to over 97%, and the F1 index also reached around 0.97. This indicates that the balance of predictions in various categories of the model has been guaranteed, and it can accurately identify different traffic congestion levels.

The tuning strategies of GridSearchCV and RandomizedSearchCV, the former relies on exhaustive search, which has a relatively low computational efficiency; The latter improves efficiency through random sampling, but there is a risk of leaving behind the optimal solution. In this experiment, the performances of the two tuning strategies were relatively close, with the accuracy rate maintained at 97.2% to 97.5%, and the F1 index was approximately 0.965 to 0.969.

Optuna is a framework based on Bayesian optimization and dynamic network search, which can quickly find high-performance solutions in a large parameter space. In this experiment, Optuna performed the most outstandingly on XGBoost, with an accuracy rate of 97.6% and an F1 index of 0.9719. The optimization strategy of BayesSearchCV, compared with Optuna, is more stable in search strategy and less sensitive to randomness. In this experiment, the combination with the random forest model performed the best, with an accuracy rate of 97.84% and an F1 index of 0.9705.

4. Conclusion

This study utilized two tree-planting models, namely random forest and XGBoost, to classify and predict the urban traffic congestion level. The model performance was comprehensively compared through four hyperparameter optimization methods (GridSearchCV, RandomizedSearchCV, Optuna, and BayesSearchCV). The experimental results show that hyperparameter tuning can significantly improve the predictive performance of the model. Compared with the baseline model, the accuracy and F1 value of the tuned model have been greatly enhanced.

Among all the experimental combinations, the combination of random forest and BayesSearchCV performed the best in terms of overall prediction Accuracy (Accuracy = 97.84%, F1_weighted = 0.9785), demonstrating strong stability and generalization ability. The combination of XGBoost and Optuna performed exceptionally well on F1-macro (0.9719), indicating that it has more advantages in the balanced prediction of different congestion levels. From this, it can be seen that in traffic prediction tasks, different optimization strategies have different focuses on model performance. BayesSearchCV strikes a balance between search efficiency and stability, making it more suitable for urban traffic management departments with high stability requirements. Optuna performs even better in enhancing model balance and is suitable for real-time traffic prediction platforms that need to quickly adapt to data distribution.

However, this study still has certain limitations. The features used in the experiment focused on vehicle speed, traffic volume, accident reports, etc. Although these fea-

tures have a strong correlation with traffic congestion, some potential influencing factors have not been fully incorporated, such as holidays, road construction conditions and other variables. In the future, more abundant external data sources can be introduced for further optimization. At the methodological level, it is also possible to explore the deep integration of multimodal data, unifying and integrating multi-source features such as camera images, real-time speed band data, traffic events and weather information into a single prediction framework to capture more abundant and detailed spatiotemporal patterns.

References

[1] United Nations Department of Economic and Social Affairs Population Division. World Urbanization Prospects 2018: Highlights. New York: United Nations, 2019.

[2] INRIX Research. 2023 Global Traffic Scorecard. INRIX, 2023. [Online]. Available: https://inrix.com/scorecard

[3] De Vlieger I, De Keukeleere D, Kretzschmar J. Environmental effects of driving behaviour and congestion. Atmospheric Environment, vol. 35, no. 27, pp. 4573–4586, 2000.

[4] Zafar N and Ul Haq I. Traffic congestion prediction based on Estimated Time of Arrival. PLOS ONE, vol. 15, no. 12, e0238200, 2020.

[5] Parsa A B, Movahedi A, Taghipour H, Derrible S, Mohammadian A K. Toward safer highways: application of XGBoost and SHAP for real-time accident detection. Accident Analysis & Prevention, vol. 136, Article 105405, 2020.

[6] Mahmoud N, Abdel-Aty M, Cai Q, Yuan J. Predicting cycle-level traffic movements at signalized intersections using XGBoost, LSTM, and GRU models. Transportation Research Part C: Emerging Technologies, vol. 128, Article 103162, 2021.

[7] Lu X, Chen C, Gao R, Xing Z. Prediction of high-speed traffic flow around city based on BO-XGBoost. Symmetry, vol. 15, no. 7, Article 1453, 2023.

[8] Boukerche A and Coutinho R W L. Machine learning-based traffic prediction models for intelligent transportation systems. Computer Networks, vol. 181, Article 107530, 2020.

[9] Pang J, Wang J, Yu J, Zhang H. Gasoline prices, traffic congestion, and carbon emissions. Transportation Research Part D: Transport and Environment, vol. 116, Article 103634, 2023.

[10] Vlahogianni E I, Karlaftis M G, Golias J C. Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies, vol. 43, pp. 3–19, 2014.

[11] Ahmed U, Tu R, Xu J, Amirjamshidi G, Hatzopoulou M, Roorda M J. GPS-based traffic conditions classification using floating car data. Transportation Research Record, vol. 2677, no. 2, pp. 1445–1454, 2023.