Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.39142

Short-Term Traffic Flow Prediction for Urban Roads Based on Machine Learning Algorithms

Qili Jia

School of Information Engineering, Chang'an University, Xi'an, China
2135825496@qq.com

Abstract. As cities grow and vehicle fleets expand, congestion on urban arterials has become harder to manage. Short-term traffic flow prediction sits at the center of many intelligent transportation applications, from adaptive signal control to route guidance. Observed flow is driven by clock time, day type, weather, road class, and speed profiles, and the resulting series shows strong nonlinearity together with daily and weekly cycles. Conventional statistical predictors often miss these patterns, while deep models can fit them but remain sensitive to learning rate, hidden width, and regularization. Manual grid search is slow and frequently stops at suboptimal settings. We therefore combine BiLSTM, Transformer, and differential evolution (DE) into one pipeline called DE-Transformer-BiLSTM: BiLSTM reads the sequence in both directions, Transformer models longer interactions, and DE searches hyperparameters without hand tuning. On a roadside dataset of 1,728 samples, the proposed model beats eleven reference algorithms on MSE, RMSE, MAE, MAPE, and R^2, which supports its use in operational traffic management.

Keywords: Short-term traffic flow, Transformer, BiLSTM, differential evolution

1. Introduction

Traffic management personnel require not only historical averages, but also predictive data for several future time intervals. With the increasing number of detectors deployed within urban traffic networks, Intelligent Transportation Systems (ITS) platforms have now incorporated 15-minute-ahead forecasting as a standard feature [1]. This task presents considerable difficulty, as traffic flow fluctuates due to peak hours, holidays, rainfall or high temperatures, as well as differences in the geometric configuration between expressways and distribution roads.

The application of roadside sensors has now made it possible to directly extract flow patterns from data [2]. Nevertheless, when prediction results are highly dependent on the temporal variation in transaction volume, models such as SVM and random forests often still struggle to perform adequately [3].

2. Data sources

Over several consecutive days, we collected recorded data from fixed counters across multiple urban road sections. Each line contains traffic data and the available covariates at that timestamp [4]. Prior

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.39142

to modeling, we removed obvious outliers, imputed minor missing values, standardized the features, and aligned all time series to a common time reference. Ultimately, 1,728 samples remained. The predictor variables include: the hour of the day, workday indicator, holiday indicator, weather code, temperature, peak hour indicator, road type, and average speed; the target variable is the count value recorded within the subsequent 15 minutes [5].

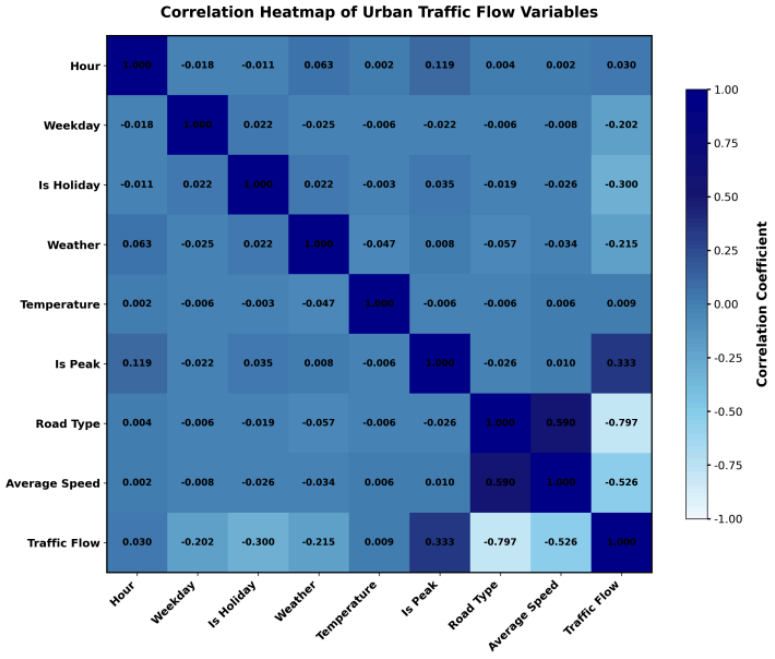


Figure 1. The correlation heat map

Figure 1 shows the paired Pearson correlation coefficients.

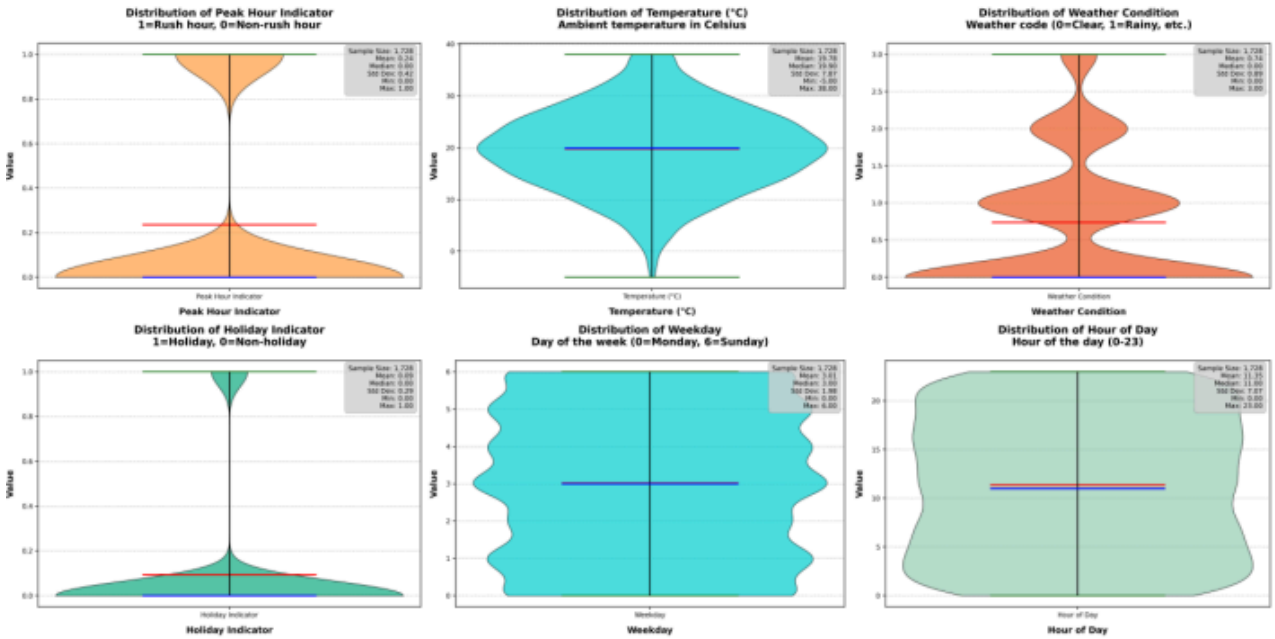


Figure 2. The violin plots

DOI: 10.54254/2755-2721/2026.39142

3. Method

3.1. Differential evolution algorithm

Starting from a small set of candidate parameters, DE continuously optimizes these parameter sets across successive generations through mutation, crossover, and greedy selection [6]. In each iteration, the mutation operation constructs a trial vector based on the weighted differences between randomly selected individuals; the crossover operation blends this trial vector with the current target vector; subsequently, the selection operation retains the vectors with higher scores and eliminates the remaining ones. Only three parameters—population size, crossover rate, and scaling factor—need to be pre-specified; nevertheless, the search process still explores the entire parameter space rather than advancing slowly along a single local gradient [7]. Figure 3 illustrates the iterative workflow adopted in this study.

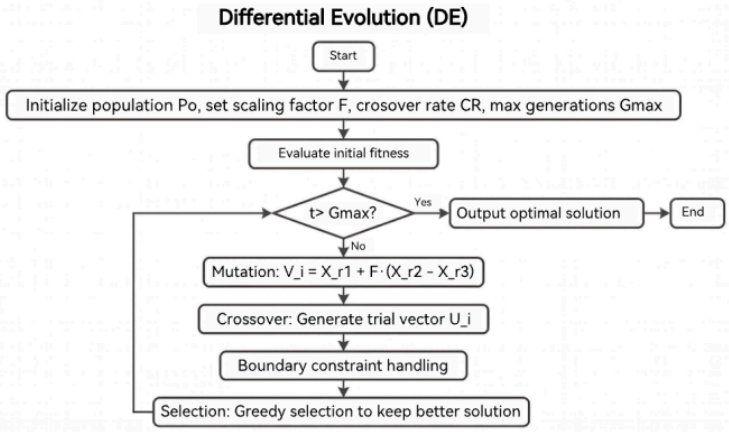


Figure 3. The network structure of the differential evolution

3.2. Transformer

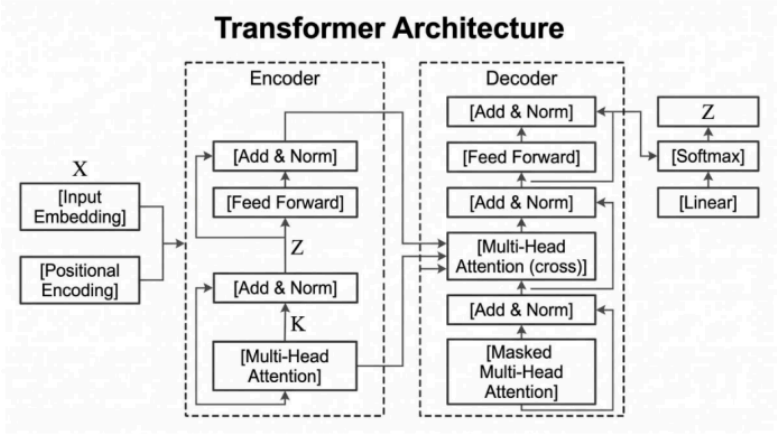


Figure 4. The network structure of Transformer

In our stack, the Transformer block handles sequence modeling without recurrence. Multi-head self-attention lets each time step attend to every other step in the window, so a morning peak, a weather shift, or a speed drop several intervals away can still affect the current estimate. We use several attention heads in parallel; each head learns a slightly different pattern, and their outputs are

DOI: 10.54254/2755-2721/2026.39142

merged before the next layer. Residual links and layer normalization keep training stable when the input series is noisy, and a small feedforward network at each position adds the nonlinearity needed for sharp flow changes. Figure 4 shows where this module sits inside the hybrid model.

3.3. BiLSTM

BiLSTM scans the same input window twice: one LSTM moves forward in time, the other moves backward, and their hidden states are joined at each step [9]. The forget, input, and output gates control what to drop, what to keep in memory, and what to pass on, which makes it easier to train on longer segments without gradients dying out. For traffic data this matters because a single window often contains both the build-up before a peak and the drop afterward; reading in both directions lets the model see that full shape rather than only what came before the current point. Figure 5 presents the module layout.

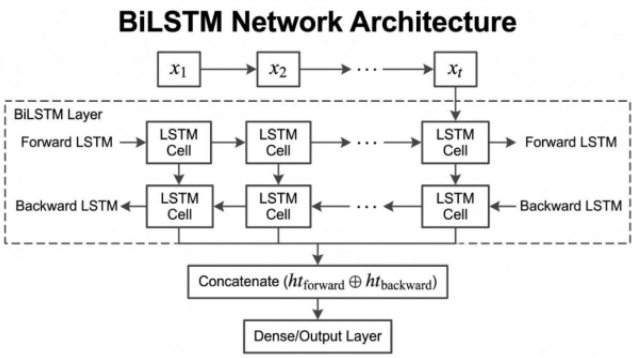


Figure 5. The network structure of BiLSTM

3.4. DE-Transformer-BiLSTM

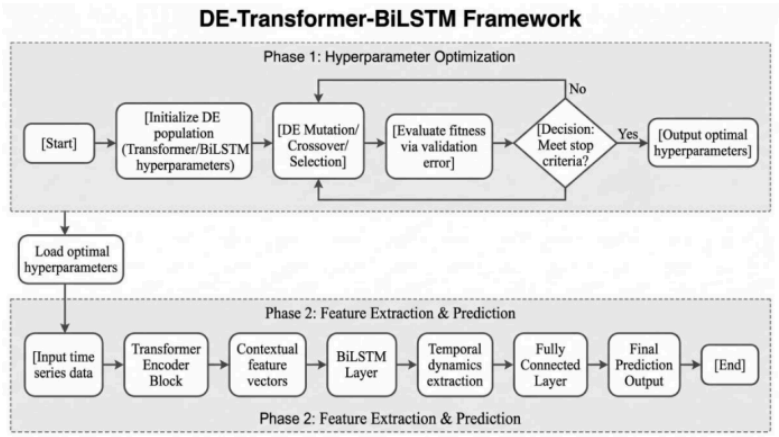


Figure 6. The network structure of DE-Transformer-BiLSTM

The full pipeline is drawn in Figure 6. Raw inputs enter BiLSTM first, which turns the local sequence into contextual embeddings. Those vectors then pass through a Transformer encoder that reweights each time step, so a distant surge in volume or a weather shift earlier in the window can still shape the current forecast. Around this core we wrap a DE loop that searches learning rate, hidden width, and the L2 penalty inside fixed bounds. The best setting found by DE is used to retrain the network on 70% of the samples; the remaining 30% is kept aside for testing [10].

DOI: 10.54254/2755-2721/2026.39142

4. Result

DE employs a population size of 8 over six generations. The search ranges are as follows: learning rate [0.001,0.01], number of hidden nodes [10, 30], and regularization coefficient [0.0001,0.1]. This deep module employs four heads, each containing 32 channels, with a maximum position encoding dimension of 512, a batch size set to 256, up to 1,000 training epochs, a gradient clipping value of 10, a dropout rate of 0.01, and a learning rate decay to 0.1× after epoch 850.

Table 1 lists the error values of the twelve models under the same data split. The lag effect is particularly pronounced in both linear regression and BP networks. Tree-based ensemble algorithms—GBDT, XGBoost, CatBoost, and AdaBoost—demonstrated excellent performance and outperformed hybrid models across all evaluation metrics. The performance metrics of DE-Transformer-BiLSTM were MSE = 1204.65, RMSE = 34.708, MAE = 27.93, MAPE = 4.612%, and R^2 = 0.981, representing the best results in the table.

Table 1. The results of the comparative experiment

Model	MSE	RMSE	MAE	MAPE	R^2
GBDT	1410.658	37.559	29.483	5.556	0.978
BP neural network	2982.536	54.613	45.061	9.469	0.954
Decision tree	1777.623	42.162	32.203	5.809	0.973
XGBoost	1474.486	38.399	29.869	5.349	0.977
ExtraTrees	1816.74	42.623	33.019	5.642	0.973
Random Forest	1930.705	43.94	30.383	5.242	0.971
Linear Regression	3351.41	57.891	47.037	12.172	0.947
CatBoost	1570.358	39.628	30.394	5.633	0.975
AdaBoost	1559.632	39.492	30.495	5.307	0.976
DE-Transformer-BiLSTM	1204.65	34.708	27.93	4.612	0.981

Bar charts in Figure 7 compare metric ranks across models, as shown in Figure 7.

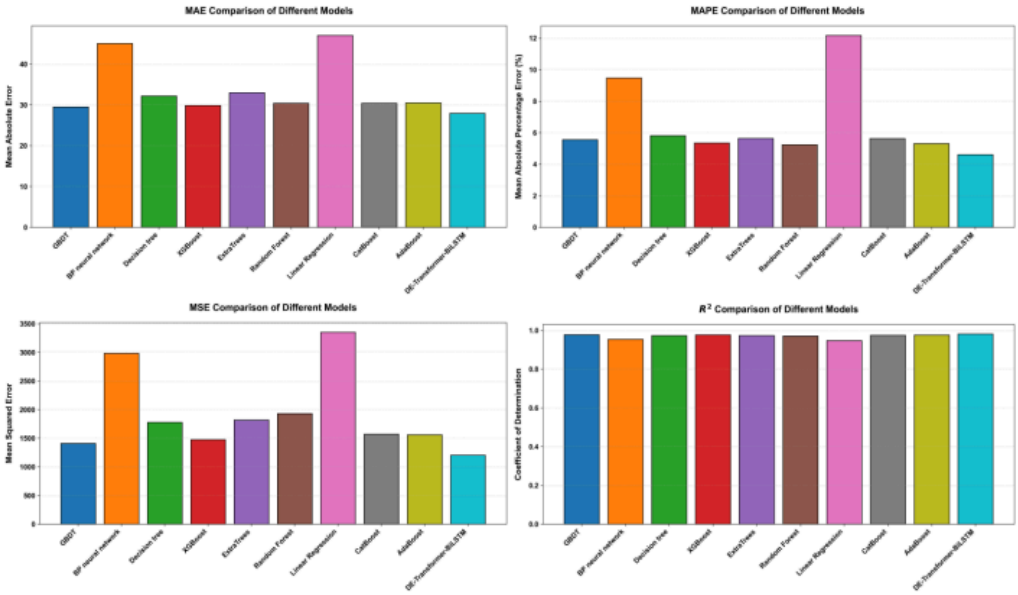


Figure 7. The comparison results of each index of each DE-Transformer-BiLSTM test set

DOI: 10.54254/2755-2721/2026.39142

Figure 8 plots predicted against measured flows on the test subset; most points hug the diagonal, with larger scatter only at the highest volumes.

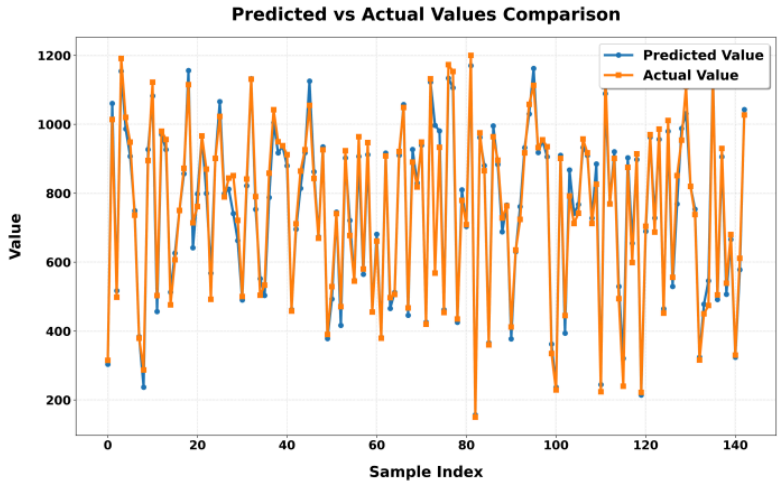


Figure 8. The predicted-actual value line graph of the DE-Transformer-BiLSTM test set

5. Conclusion

We evaluated whether integrating evolutionary optimization with the BiLSTM–Transformer core model could improve the accuracy of 15-minute urban traffic flow forecasting. Compared to commonly used machine learning benchmark methods, this combined approach indeed reduces the error rate on detector datasets, while the DE method significantly alleviates the workload of manual search. BiLSTM provides local temporal cues, whereas attention diffusion exerts its influence across the entire time window. Subsequent work includes conducting online updates for land-use change and carrying out pilot tests in corridor areas characterized by sparse data availability. Nevertheless, the current accuracy and stability of this model make it an ideal candidate solution for signal timing experiments and short-range network monitoring platforms.

6. Discussion

Compared to GBDT, XGBoost, and other strong baseline models, while the performance improvement was not substantial, it was consistently observed across all error metrics we reported; this indicates that the hybrid model captures temporal structures that were overlooked by tabular learners. BiLSTM effectively captures the sharp rises and falls around peak periods, while the attention mechanism detects slow variations associated with weather conditions or off-peak times. DE also reduces the workload associated with manual tuning: once the search scope is defined, the same workflow can be repeatedly executed without the need to adjust settings individually for each case. However, all experiments were conducted using the same urban detector dataset; therefore, their performance in scenarios involving emergencies, sensor outages, or the addition of new corridors remains unclear. Before applying this model to real-time signal control, it is necessary to perform rolling retraining and testing on other networks.

References

[1] Yang, Y., et al. (2026). New fusion methods using a weighted l1 conflict measure for reliability assessment with multi-credibility evidence. *Information Sciences*, 123173.

DOI: 10.54254/2755-2721/2026.39142

[2] Qureshi, K. A., & Malick, R. A. S. (2025). Tweet credibility ranker: A credibility features' fusion model. *Cognitive Computation, 17*(1), 56.
[3] Xiaoyu, Y., et al. (2025). Multi-source interactive resilient fusion algorithm based on RIEKF. *IEEE Sensors Journal*.
[4] Si, L., et al. (2023). Fusion positioning of mobile equipment in underground coal mine based on redundant IMUs and UWB. *IEEE Transactions on Industrial Informatics, 20*(4), 5946–5958.
[5] Srba, I., et al. (2026). A survey on automatic credibility assessment using textual credibility signals in the era of large language models. *ACM Transactions on Intelligent Systems and Technology, 17*(2), 1–80.
[6] Huang, R., et al. (2024). Risk assessment approach for tunnel collapse based on improved multi-source evidence information fusion. *Environmental Earth Sciences, 83*(1), 18.
[7] Zhang, Y., et al. (2024). Cross-modal credibility modelling for EEG-based multimodal emotion recognition. *Journal of Neural Engineering, 21*(2), 026040.
[8] Yang, T., et al. (2024). Enhancing quality evaluation in traditional Chinese medicine: Utilizing dual wavelength fusion fingerprint, electrochemical fingerprint, and DSC fingerprint. *Journal of Chromatography A, 1722*, 464907.
[9] Sharaf, I. M., et al. (2024). A novel dual-level multi-source information fusion approach for multicriteria decision making applications. *Applied Intelligence, 54*(22), 11577–11602.
[10] Zhuo, R., et al. (2025). The grinding wheel wear condition monitoring method based on multi-sensor information hierarchical fusion for high-speed cylindrical grinding. *Advanced Engineering Informatics, 67*, 103541.