Research on Traffic Forecasting Using Time Convolutional Networks Optimised by RIME Algorithm

Haorui Li*

Guangdong Country Garden School, Foshan, Guangdong 528000, China

*13552931174@163.com

Abstract. This paper addresses the challenges in traffic flow forecasting—namely data non-linearity, strong spatio-temporal coupling, and hyperparameter dependence on manual tuning—by proposing a hybrid forecasting model based on the Frost-Ice Optimisation Algorithm (RIME). This model integrates a Time Convolutional Network (TCN), a Bidirectional Gated Recurrent Unit (BiGRU), and a Multi-Head Attention mechanism. The model leverages TCN's convolutional layers to capture local spatial features within traffic sequences, employs BiGRU bidirectional encoding to encode long-term temporal dependencies, and incorporates a multi-head attention mechanism to dynamically weight critical temporal step information. To further enhance performance, the RIME algorithm is applied for automatic hyperparameter tuning of learning rates, convolution kernel sizes, and neuron counts, overcoming limitations of traditional manual parameter adjustment. Experiments on real-world traffic datasets demonstrate that the proposed model achieves outstanding results in metrics such as Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), validating its effectiveness and robustness in high-volatility traffic forecasting tasks.

Keywords: Deep learning, Traffic forecasting, Intelligent algorithm.

1 Introduction

With accelerating urbanisation and the advancement of intelligent systems, traffic forecasting in sectors such as transport and power has become central to enhancing infrastructure efficiency[1]. Issues such as traffic congestion and fluctuations in grid load have become increasingly prominent. Accurate forecasting provides a basis for signal control and resource allocation[2]. However, traffic data exhibits strong volatility, spatiotemporal coupling, and nonlinear characteristics. Traditional models struggle to account for both spatial correlation and temporal dependency, while manually setting hyperparameters limits accuracy[3].

Reference [4] proposes an SVM-ANN fuzzy logic hybrid model for traffic forecasting in small and medium-sized cities. Its advantages include light weight and strong interpretability, while its disadvantages are limited dataset diversity and insufficient generalisation and anomaly adaptation. Reference [5] constructs a CPO-CNN-LSTM-Attention model to enhance Paris traffic prediction accuracy. Its strengths lie in noise

Y. Xu et al. (eds.), *Proceedings of the 2026 5th International Conference on Engineering Management and Information Science (EMIS 2026)*, Advances in Computer Science Research 130,
https://doi.org/10.2991/978-94-6239-652-4_16

resistance and stability, though its drawbacks encompass structural complexity, time-consuming training, and unvalidated adaptation to special scenarios. Reference [6] proposes the SSA-VMD-LSTM method to optimise transmission line current capacity prediction. Its strengths include reduced volatility and efficient optimisation, while its weaknesses are poor transferability and weak multi-source data fusion. Reference [7] constructs a CNN-LSTM-Attention model for predicting seepage flow. Its advantages encompass comprehensive spatio-temporal capture and robust performance, but it suffers from numerous parameters and reliance on manual parameter tuning.

This paper constructs a TCN-BiGRU-Multi-Head Attention-RIME fusion framework to enhance prediction performance. First, TCN extracts local spatial features from traffic data, while BiGRU models bidirectional long-term temporal dependencies. Multi-head attention is then embedded to dynamically assign step weights, thereby reinforcing critical information. Finally, the RIME algorithm optimises hyperparameters such as learning rate and convolution kernel size, circumventing the shortcomings of manual tuning. Validation on real-world datasets demonstrates superiority over traditional models in both accuracy and efficiency, providing robust support for practical traffic forecasting.

2 Method

2.1 Time Convolutional Network

The Temporal Convolutional Network (TCN) represents a class of convolutional architectures specifically engineered for sequence modeling[8].

For a one-dimensional input sequence $\mathbf{x} = [x_1, x_2, \ldots, x_T]$ and a convolutional kernel $\mathbf{w} = [w_1, w_2, \ldots, w_K]$ of size K , a standard convolution might rely on inputs from both the past and future of t . Causal convolution prevents this by applying a one-sided padding of K-1 zeros to the beginning of the input sequence. This ensures the output sequence length matches the input length, and that the output y_t is computed solely from inputs x_{t-K+1} to x_t. The mathematical operation is expressed as:

$$y_t = \sum_{k=1}^{K} w_k \cdot x_{t-K+k} \tag{1}$$

where the input values for indices less than 1 are taken as zero.

For a dilation factor d , the convolution operation at timestep t is formulated as:

$$y_t = \sum_{k=1}^{K} w_k \cdot x_{t-(K-k)d} \tag{2}$$

When d=1 , this simplifies to a standard causal convolution. By stacking multiple layers with dilation factors that increase exponentially (e.g., d = 2^l for layer l), the receptive field of the network grows exponentially. The receptive field RF_l at layer l can be calculated as:

$$RF_l = 1 + \sum_{i=1}^{l} (K_i - 1) \times d_i \tag{3}$$

2.2 Bidirectional Gated Recurrent Unit

The core of the GRU unit relies on two gating mechanisms, the reset gate and the update gate, to regulate the flow of information[9]. For a given timestep t with input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$, the operations are defined as follows:

The update gate $\mathbf{z}_t$ determines the degree to which the previous hidden state is carried forward. It is computed by:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z) \tag{4}$$

where σ is the sigmoid function, $\mathbf{W}_z$ is the weight matrix, $\mathbf{b}_z$ is the bias vector, and $[\mathbf{h}_{t-1}, \mathbf{x}_t]$ denotes the concatenation of the two vectors.

The reset gate $\mathbf{r}_t$ controls how much of the previous hidden state is used to compute a new candidate state. Its calculation is:

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r) \tag{5}$$

The candidate hidden state $\tilde{\mathbf{h}}_t$ is then generated using the modulated previous state:

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \cdot [\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_h) \tag{6}$$

Here, $\odot$ represents the Hadamard (element-wise) product. The reset gate allows the unit to potentially "forget" parts of the past state when computing the new candidate.

2.3 Multi-Head Attention Mechanism

The foundation is the scaled dot-product attention[10]. Given a query matrix $\mathbf{Q}$, a key matrix $\mathbf{K}$, and a value matrix $\mathbf{V}$, the attention weights are computed by taking the softmax of the dot products of the query with all keys, scaled by the square root of the key dimension d_k to counteract gradient vanishing issues that arise when the dot products become large in magnitude. The output is a weighted sum of the values:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{7}$$

The multi-head attention mechanism projects the queries, keys, and values h times using separate, learned linear projections. For each head i (where i = 1, …, h), the projected matrices are:

$$\mathbf{Q}_i = \mathbf{Q}\mathbf{W}_i^Q, \quad \mathbf{K}_i = \mathbf{K}\mathbf{W}_i^K, \quad \mathbf{V}_i = \mathbf{V}\mathbf{W}_i^V \tag{8}$$

Here, $\mathbf{W}_i^Q, \mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are the learnable projection matrices for head i . The scaled dot-product attention is then applied to each set of projected matrices in parallel:

$$\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) \tag{9}$$

2.4 RIME Optimization Algorithm

The algorithm maintains a population of candidate solutions, conceptualized as ice particles. Each solution $\mathbf{X}_i (for i = 1,2,\ldots,N,$ where N is the population size) is evaluated by a fitness function $f(\mathbf{X}_i)$, which typically represents the inverse of a loss function like prediction error; thus, a higher fitness indicates a better solution.

A critical control parameter is the temperature T , which decreases as the iteration count t increases, simulating an annealing process. The temperature update can be modeled as:

$$T(t) = T_0 \cdot \exp(-\lambda t) \tag{10}$$

where T_0 is the initial temperature and λ is a decay constant. This decreasing temperature governs the balance between exploration and exploitation.

3 Experiments

3.1 Dataset and Simulation Environment

The experimental data was derived from real-world traffic flow monitoring data of a typical urban road section with a time span of 6 months, including 43800 data samples. The data was collected at a 5-minute sampling interval, covering key features such as historical traffic flow, forming a multi-dimensional input feature matrix. chronological order to simulate the actual prediction scenario.

The experiments were conducted on a computer with an Intel Core i7-12700H processor, 32GB DDR5 memory, and an NVIDIA RTX 3060 GPU. The software environment included Python 3.9, PyTorch 2.0 deep learning framework, and Scikit-learn for data processing and evaluation.

3.2 Simulation Results

Figure 1 illustrates the adaptation value evolution during the RIME algorithm's model hyperparameter optimisation process. The horizontal axis represents the iteration count, while the vertical axis denotes the adaptation value. The graph reveals a rapid decline in adaptation value during the first two iterations, followed by a stabilisation phase.

Figure 2 illustrates the dynamic evolution of loss during model training, with “training error” as the radial axis and “number of iterations” as the circumferential axis. The curve trajectory indicates that as iterations increase, training error progressively diminishes.

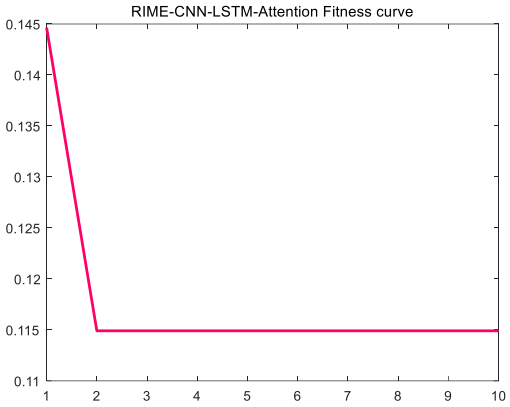

Fig. 1. RIME Algorithm Fitness Curve

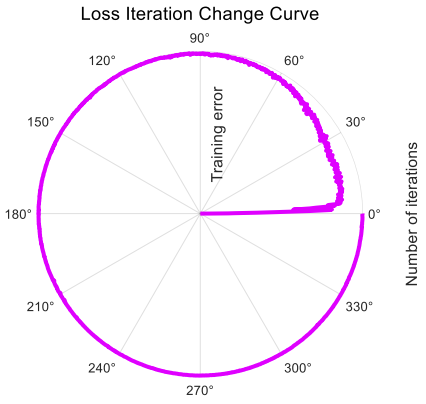

Fig. 2. Loss Iteration Change Polar Coordinate Plot

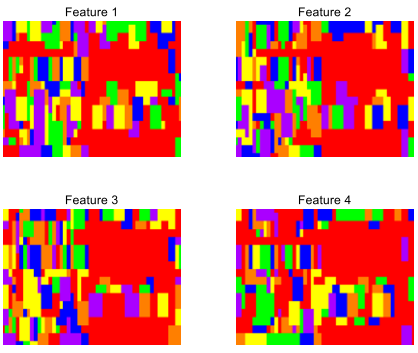

Fig. 3. Characteristic Thermal Distribution Map

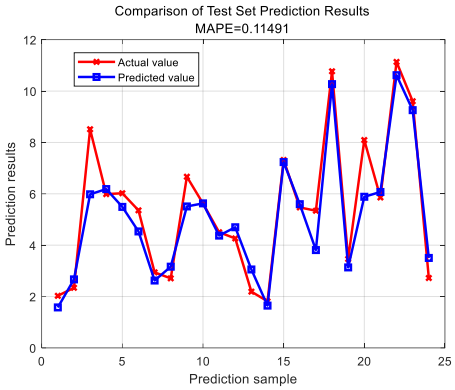

Fig. 4. Comparison of Test Set Prediction Results

In Figure 3, the four subplots respectively display the distribution heatmaps of the four input features, with different colours representing variations in feature values. As evident from the plots, each feature exhibits distinct regionalised distribution patterns.

Figure 4 compares the trends of actual values (red) and predicted values (blue) on the test set, with a MAPE of 0.11491. Overall, the predicted values closely align with the actual trends, demonstrating good fit at peaks and troughs.

4 Conclusion

This study demonstrates the significant advantages of integrating multimodal deep learning with intelligent optimisation algorithms in complex traffic forecasting through the construction of a hybrid TCN-BiGRU-Multi-Head Attention-RIME framework. The model simultaneously captures spatio-temporal features and bidirectional long-term dependencies, while the RIME optimisation mechanism effectively enhances the rationality of hyperparameter configuration and convergence efficiency. The final test set prediction curve exhibits high alignment with actual values (MAPE of 0.11491), indicating the model possesses robust generalisation capabilities and engineering applicability.

References

1. Fang Y, Liang Y, Hui B, et al. Efficient large-scale traffic forecasting with transformers: A spatial data management perspective[C]//Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1. 2025: 307-317.
2. Cai D, Chen K, Lin Z, et al. JointSTNet: Joint pre-training for spatial-temporal traffic forecasting[J]. IEEE Transactions on Consumer Electronics, 2024, 71(2): 6239-6252.
3. Magar B ,Kulkarni R ,Khatavkar V , et al.Predictive network congestion management for enterprise systems: a graph neural network approach with operational insights[J].Journal of Electrical Systems and Information Technology,2025,12(1):92-92.DOI:10.1186/S43067-025-00282-1.

4. Abbas B ,Alyas T ,Abbas Q , et al.A hybrid support vector machine and neural network model with fuzzy logic fusion for smart city traffic prediction[J].Scientific Reports,2025,15(1):34758-34758.
5. Topilin I, Jiang J, Feofilova A, et al. Traffic Flow Prediction via a Hybrid CPO-CNN-LSTM-Attention Architecture[J]. Smart Cities, 2025, 8(5).
6. Wang Shuai, Shen Jiewen, Xu Bin, et al. Dynamic Ampacity Prediction Method for Overhead Transmission Lines Based on SSA-VMD-LSTM[J/OL].Electronic Measurement Technology,1-11[2025-11-20].https://link.cnki.net/urlid/11.2175.tn.20250616.1912.012.
7. Li Shiwan, Yuan Mingdao, Xu Yunqian, et al. CNN-LSTM Earth-rock Dam Infiltration Prediction Model Based on Attention Mechanism[J].Science Technology and Engineering,2025,25(21):9102-9108.
8. Xia Ying, Liu Min. Traffic Flow Prediction Based on Spatiotemporal Attention Convolutional Neural Network[J].Journal of Southwest Jiaotong University,2023,58(02):340-347.
9. Li Yuhang, Zhuang Jihui, Chen Zhenbin, et al. Mechanical Design and Manufacturing,2025,(09):379-384.
10. Yu Anjun, Li Yindi, Yang Zheyi, et al. Highway traffic flow prediction method based on multi-dimensional attention mechanism[J].Journal of Automotive Safety and Energy Conservation,2025,16(03):463-469.