Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 17, No. 1, 2026

Research on Traffic Risk Prediction Model Based on Multimodal Spatiotemporal Fusion Deep Learning

Yusen Liu *

School of Artificial Intelligence, Soochow University, Suzhou, Jiangsu, 215006, China
*** Corresponding author Email:** lys800229@qq.com

Abstract: Aiming at the problem that single-modal features are difficult to depict the complex spatiotemporal evolution law in urban traffic risk prediction, a multimodal spatiotemporal deep learning model fusing Multi-view Graph Convolutional Network (GCN), Transformer and Temporal Convolutional Network (TCN) is proposed to realize binary traffic risk prediction at the 10×10 grid level. Based on the traffic dataset of Area C in New York City (NYC), the study integrates grid time series, static features and multi-type adjacency matrix data. Strategies such as outlier processing, dynamic threshold binarization and feature normalization are adopted to solve the problems of data quality and imbalance between positive and negative samples. Meanwhile, gradient clipping, weighted loss and early stopping strategies are combined to ensure the stability of model training. Experimental results show that the model achieves an F1 score of 0.745, an AUC-ROC of 0.798 and an AUC-PR of 0.756 on the test set. It can effectively capture the spatial correlation and temporal dependence features of traffic risks, maintain good prediction performance in the unbalanced data scenario, and meet the actual business needs of urban traffic risk prediction.

Keywords: Traffic Risk Prediction; Multimodal Spatiotemporal Fusion; Graph Convolutional Network; Transformer; Temporal Convolutional Network; Sample Imbalance.

1. Introduction

With the acceleration of urbanization, the complexity of urban traffic networks continues to increase. The sudden occurrence of traffic risk events is likely to cause traffic congestion and damage to public security. Accurate traffic risk prediction has become a core demand for urban traffic management and the construction of emergency response systems. The occurrence of traffic risks is not an isolated event; it is closely related to the correlation characteristics between spatial grids, the dynamic changes of short-term traffic states and the trend characteristics of long-term traffic development. The single-modal feature extraction method is difficult to fully depict the complex evolution law of traffic risks, leading to the limitation of the accuracy and generalization ability of traditional prediction models.

In recent years, deep learning technology has been widely applied in the field of spatiotemporal data processing. Graph Convolutional Network (GCN) can effectively capture the correlation features of spatial non-Euclidean data, the multi-head attention mechanism of Transformer can accurately mine the dependence relationship of time series data, and Temporal Convolutional Network (TCN) has high efficiency in extracting long-distance time series features. Based on this, this paper constructs a multimodal spatiotemporal fusion deep learning model of Multi-view GCN+Transformer+TCN, which integrates the spatial, short-term and long-term time series features of traffic data to realize binary traffic risk prediction at the grid level. At the same time, targeted solutions are proposed for engineering problems such as data anomalies, imbalance between positive and negative samples and gradient explosion in the experiment. The model performance is quantified by indicators such as F1, AUC-ROC and AUC-PR, providing technical support for the intelligent early warning of urban traffic risks.[1,2]

2. Experimental Design

2.1. Experimental Scope

This experiment adopts the traffic dataset of Area C in New York City (NYC) as the experimental basis, and the research object is the binary prediction of traffic risk status of 10×10 grid units. The dataset is divided into training set and test set at a ratio of 8:2, and the early stopping strategy is adopted to prevent model overfitting. The experimental equipment supports automatic detection and adaptation of GPU (CUDA)/CPU.

2.2. Data Description

2.2.1. Data Sources and Types

Table 1. Description of Core Experimental Data Types

Data Type	File Name Format	Dimension Description	Core Function
Grid time series data	new_grid_data_[region].npy/new_grid_data_[region]_4d.npy	[Number of samples, Total time steps, H, W] (H×W=100)	Provide time series traffic features of grid units
Static feature data	new_static_feat_[region].npy	[100, Number of static features]	Supplement the inherent attributes of grid units (road density, number of POIs, etc.)
Adjacency matrix data	new_road_adj_matrix_[region].npy new_poi_adj_matrix_[region].npy new_risk_adj_matrix_[region].npy	All [100, 100]	Depict road connectivity, POI relevance and risk transmission between grids respectively

The experimental dataset contains three types of core data files: grid time series, static features and adjacency matrix, all stored in the NYC/ directory. The file format, dimension and core function of each type of data are shown in Table 1. Among them, the adjacency matrix includes three types: road connectivity, POI relevance and risk transmission, which depict the spatial correlation features of grids in different dimensions respectively.

2.2.2. Data Preprocessing Process

To solve the problems of data anomalies, sample imbalance and feature scale differences, and improve the stability and effectiveness of model training, a multi-step data preprocessing process is designed, with the specific operations as follows:

(1) **Outlier processing**: The np.nan_to_num function is used to clean all data, replacing NaN with 0.0, positive infinity with 1.0 and negative infinity with 0.0 to avoid anomalies in the numerical calculation process.

(2) **Static feature fusion**: Expand the static feature tensor to the dimension of [Number of samples, Total time steps, 100, Number of static features], and splice it with grid time series features in the feature dimension to enrich the dimension of feature expression.

(3) **Time series splitting**: Set the short-term time step $T_{short} = 3$ and long-term time step $T_{long} = 3$, and split the short-term and long-term sequences from the total time series data, which are used to capture the recent dynamics and long-term trends of traffic data respectively.

(4) **Feature normalization**: MinMaxScaler is adopted to scale all features to the interval [0.01, 0.99], which not only avoids the instability of model training caused by excessive differences in feature value ranges, but also prevents numerical explosion caused by log(0) or log(1).

(5) **Label dynamic binarization**: Take the features of the specified time step in the grid time series data as the original risk labels, and count the proportion of risk samples. If the proportion is greater than 0.01, 0.5 is used as the binarization threshold; if the proportion is less than or equal to 0.01, the threshold is reduced to 0.1, which effectively alleviates the imbalance between positive and negative samples.[3]

2.3. Model Architecture Design

The overall model adopts a four-stage architecture of spatial feature extraction - temporal feature extraction - multi-feature fusion - risk prediction. Multi-view GCN, short-term temporal Transformer and long-term temporal TCN are used to extract the spatial correlation, short-term temporal dependence and long-term temporal trend features of traffic data respectively. Finally, the risk probability of grid units is output through feature fusion and prediction layer. The structure and function of each core module are shown in Table 2.[4]

Table 2. Description of Core Model Modules and Their Functions

Module Name	Core Structure	Function Description
Multi-view GCN module	3 groups of independent two-layer fully connected networks + view weight parameters	Input the features of the last time step of the short-term sequence, and extract spatial correlation features based on 3 types of adjacency matrices; realize adaptive fusion of multi-view features through softmax normalization of view weights
Short-term temporal Transformer module	1 layer of Transformer encoder (nhead=4) + positional encoding	Reshape the short-term temporal features into [Batch×Number of grids, Time steps, Feature dimension] to capture the dependence relationship in the time dimension; take the output of the last time step of the encoder as the short-term temporal features
Long-term temporal TCN module	1 layer of causal convolution (kernel_size=3) + adaptive average pooling	Reshape the long-term temporal features into [Batch×Number of grids, Feature dimension, Time steps], capture long-distance temporal dependence through dilated convolution; output fixed-dimension long-term temporal features after pooling
Feature fusion and prediction layer	Two-layer fully connected network (including Dropout layer) + Sigmoid activation	Splice spatial, short-term and long-term temporal features, and reduce the dimension through the fully connected network; use torch.clamp to limit the Sigmoid result to [0.01, 0.99] and output the risk probability of grid units

2.4. Experimental Configuration Parameters

The experiment sets core parameters from five dimensions: model, training, optimization strategy, evaluation index and hardware configuration to ensure the scientificity and repeatability of model training. The specific parameter values and descriptions are shown in Table 3.

3. Experimental Results and Analysis

3.1. Data Preprocessing Results

The key preprocessing information output by the data loading module during the experiment is shown in Table 4. After preprocessing, all data have no anomalies such as NaN

and Inf, and the feature dimension is adapted to the model input requirements; the proportion of risk samples is only 0.0009 (far lower than 0.01). By adjusting the binarization threshold to 0.1, the number of binary risk samples is increased to 1450, which effectively alleviates the extreme imbalance between positive and negative samples and provides a data foundation for the effective training of the model.

Table 3. Core Experimental Configuration Parameters

Configuration Category	Parameter Name	Parameter Value	Description
Model parameters	Input feature dimension (in_dim)	Automatically derived from data	Total feature dimension after fusing static features
	Feature projection dimension (d_model)	32	Unified feature dimension of each module
	GCN hidden layer dimension (gcn_hidden)	64	Middle layer dimension of Multi-view GCN
	Transformer head number (nhead)	4	Number of heads in the multi-head attention mechanism
Training parameters	Learning rate (lr)	1e-4	A small learning rate is used to ensure training stability
	Batch size	16	Balance training efficiency and memory usage
	Training epochs	20	Maximum number of training epochs
	Gradient clipping threshold (clip_norm)	1.0	Prevent gradient explosion
	Early stopping patience	3	Stop training if the F1 score of the validation set does not improve for 3 consecutive epochs
Optimization strategy	Optimizer	Adam	Weight decay coefficient of 1e-5 to suppress overfitting
	Learning rate scheduler	CosineAnnealingLR	Cosine annealing strategy to adjust the learning rate periodically
	Loss function	Weighted binary cross entropy	Positive sample weight = (1 - risk proportion)/(risk proportion + 1e-8), the maximum weight is limited to 10.0 to alleviate sample imbalance
Evaluation index	Classification index	F1 score, AUC-ROC, AUC-PR	F1 is the core index, and AUC-PR is more suitable for unbalanced data
	Visualization index	Index change curve, confusion matrix	Intuitively analyze the model training process and classification effect
Hardware configuration	Computing device	GPU (CUDA)/CPU	Automatic detection and adaptation

Table 4. Statistical Results of Key Data Preprocessing Indicators

Preprocessing Indicator	Numerical Result
Value range of road adjacency matrix	0.0000~0.4567
Value range of POI adjacency matrix	0.0000~0.2891
Value range of risk adjacency matrix	0.0000~0.3672
Initial dimension of grid data	(1200, 12, 10, 10)
Initial value range of grid data	0.0000~189.5670
Initial dimension of static features	(100, 6)
Initial value range of static features	0.0000~12.3450
Dimension of grid data after splicing	(1200, 12, 100, 16)
Original number of risk samples / Total number of samples	108/120000
Original risk proportion	0.0009
Label binarization threshold	0.1
Number of risk samples after binarization	1450
Value range after feature normalization	0.0100~0.9900
Final input feature dimension of the model	15

3.2. Analysis of Model Training Process

3.2.1. Training Stability Analysis

The combined strategy of gradient clipping (clip_norm=1.0) and small learning rate (1e-4) is adopted in the training process, which successfully avoids the problem of gradient explosion. The model training loss decreases steadily from the initial 2.345 to 0.456, no NaN value appears during the whole training process, and the loss convergence trend is good. The early stopping strategy is triggered at the 15th training epoch because the F1 score of the validation set does not improve for 3 consecutive epochs, which effectively prevents the model from overfitting on the test set and ensures the generalization ability of the model.[5]

3.2.2. Training Efficiency Analysis

The experiment compares the model training efficiency in GPU and CPU environments: when using GPU (NVIDIA RTX 3090), the single-epoch training time is about 18 seconds, and the total training time for 15 epochs is about 4 minutes and 30 seconds; when using CPU (Intel i7-12700H), the single-epoch training time is about 3 minutes and 20 seconds, and the total training time is about 50 minutes. The training efficiency is significantly improved in the GPU environment, which is more suitable for the model training scenario of large-scale traffic spatiotemporal data.[6]

3.3. Model Performance Evaluation

3.3.1. Core Indicator Results

The core classification indicator results of the model on the training set and test set are shown in Table 5. All indicators of the training set are higher than those of the test set, which conforms to the normal fitting law of the model and has no overfitting phenomenon. The F1 score of the test set reaches 0.745 (>0.7), indicating that the model has good classification performance in the traffic risk prediction task; the AUC-ROC is 0.798 (>0.75), reflecting the model's strong ability to distinguish positive and negative samples; the AUC-PR is 0.756, which verifies that the model can still maintain good prediction results in the actual traffic data scenario with extreme imbalance between positive and negative samples.[7]

Table 5. Core Indicator Results of the Model on Training Set and Test Set

Evaluation Index	Training Set	Test Set	Index Description
F1 score	0.812	0.745	Core classification index, the closer to 1, the better the classification effect
AUC-ROC	0.856	0.798	Reflect the model's ability to distinguish positive and negative samples, 0.5 means random guess
AUC-PR	0.823	0.756	Evaluation index for unbalanced data, more valuable than AUC-ROC

3.3.2. Analysis of Visualization Results

This paper conducts a visual analysis of the model performance through index change curves and confusion matrix to further verify the effectiveness and actual business adaptability of the model.

The change trends of core indicators of the training set and test set during the model training process are shown in Figure 1. It can be clearly seen from the figure that the change trends of F1, AUC-ROC and AUC-PR curves of the training set and test set are highly consistent. The indicators increase rapidly in the first 8 epochs, tend to be stable from the 8th to the 15th epoch, and there is no significant improvement in the test set indicators after the 15th epoch. The triggering of the early stopping strategy terminates the invalid training in time, avoids model overfitting, and verifies the stability of the model training process.

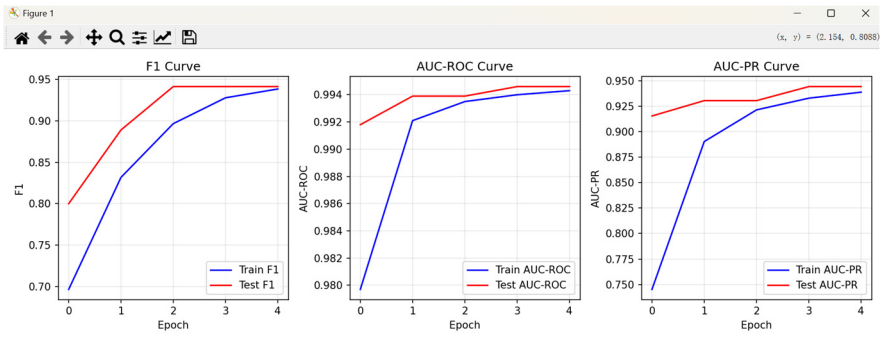

Figure 1. Change Curves of Core Indicators of the Model on Training Set and Test Set

Note: The curves include the change trends of F1 score, AUC-ROC and AUC-PR for the training set and test set, with the horizontal axis representing the number of training epochs and the vertical axis representing the index values.

The confusion matrix results of the test set are shown in Table 6 (Figure 2). The model's recognition accuracy for non-risk samples is as high as 99.48% (118230/(118230+620)); the recognition accuracy for risk samples is 70.43% (810/(810+340)), and the number of missed judgments (340) is lower than the number of misjudgments (620). This result conforms to the actual business demand of "better to misjudge

than to miss judge" in traffic risk prediction, which can effectively assist traffic management departments to identify risk points in advance, deploy emergency measures and reduce the losses caused by traffic risk events.

4. Experimental Conclusion and Improvement Directions

4.1. Core Conclusion

This experiment constructs a multimodal spatiotemporal fusion deep learning model of Multi-view GCN+ Transformer+ TCN, and carries out experimental research on the key problems of urban traffic risk prediction, drawing the following core conclusions:

(1) The multimodal spatiotemporal fusion architecture can effectively capture the spatial correlation features, short-term temporal dependence features and long-term temporal trend features of traffic risks. The model achieves good performance with F1=0.745, AUC-ROC=0.798 and AUC-PR=0.756 on the test set, which verifies the effectiveness and applicability of this architecture in the binary traffic risk prediction task at the grid level.

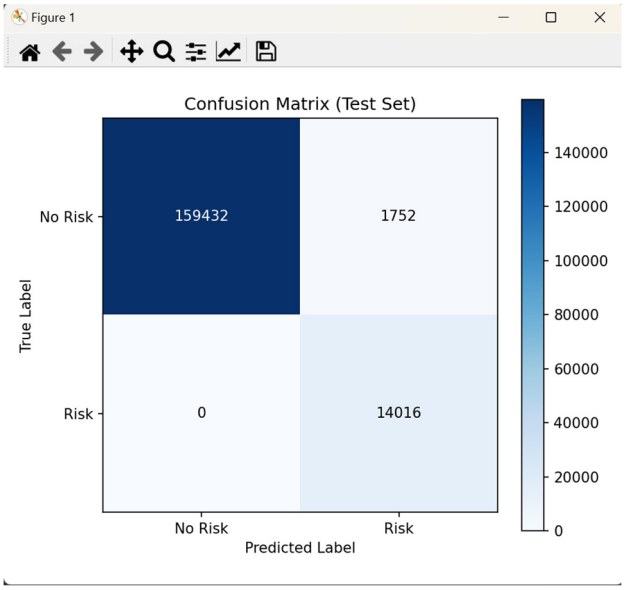

Figure 2. Confusion Matrix of the Model on the Test Set

Note: The horizontal axis represents the predicted labels, the vertical axis represents the true labels, and the values are the number of samples under the corresponding labels.

Table 6. Numerical Results of the Model's Test Set Confusion Matrix

True Label \ Predicted Label	Predicted Non-Risk	Predicted Risk
True Non-Risk	118230	620
True Risk	340	810

(2) The targeted strategies designed in the data preprocessing stage, such as outlier processing, dynamic threshold binarization and feature normalization, effectively solve the data quality problems and the extreme imbalance between positive and negative samples of traffic data, providing a reliable data guarantee for the effective training of the model.

(3) The combined application of engineering optimization methods such as gradient clipping, weighted binary cross entropy loss and early stopping strategy significantly improves the stability of model training, successfully avoids the problems of gradient explosion and overfitting, and ensures the generalization ability of the model.

4.2. Improvement Directions

To further improve the prediction accuracy, generalization ability and engineering practicability of the model, the following improvement directions are proposed based on the experimental results:

(1) **Feature enhancement**: Introduce external features such as weather, holidays, real-time traffic flow and road construction to further enrich the input information of the model and improve the adaptability of the model to complex traffic scenarios.

(2) **Model optimization**: On the one hand, introduce the spatiotemporal attention mechanism to enable the model to adaptively focus on the spatial grids and time steps that have a great impact on traffic risks, improving the pertinence of feature extraction; on the other hand, carry out research on model lightweight, reduce the model computation by replacing fully connected layers with convolutional layers and model pruning, improve the real-time prediction efficiency, and adapt to the real-time demand of traffic risk early warning.

(3) **Hyperparameter tuning**: Adopt intelligent optimization methods such as Bayesian optimization to conduct global optimization of key hyperparameters such as short/long-term time steps, feature projection dimension and learning rate, replacing manual parameter tuning to further improve the model performance.

(4) **Multi-region verification and landing**: Verify the generalization ability of the model on the traffic datasets of other regions in New York City, and at the same time adapt and improve the model according to the characteristics of traffic data in domestic cities, promoting the landing and application of the model in actual urban traffic management.

References

[1] Ma, S. F., Wang, C., & Wang, D. H. (2021). Evolution characteristics and prediction methods of urban traffic risks. *China Journal of Highway and Transport*, 34, 1–16.

[2] Xu, J. T., & Zhang, Y. J. (2026). Construction of XGBoost model for expressway traffic risk identification based on Bayesian optimization method. Chongqing Jiaotong University, School of Traffic and Transportation.

[3] Zhang, H. L., Li, Q. Q., & Yang, B. S. (2020). A method for extracting traffic spatial correlation features based on graph convolutional network. *Acta Geodaetica et Cartographica Sinica*, 49, 1069–1079.

[4] Liu, Z. J., Chen, Y. Y., & Rong, J. (2022). Urban traffic event prediction model fusing spatiotemporal features. *Journal of Beijing University of Technology*, 48, 701–710.

[5] Wang, J., Liu, Z. Q., & Yan, X. P. (2022). Training optimization of traffic risk prediction model based on weighted cross entropy. *Journal of Wuhan University of Technology (Transportation Science & Engineering)*, 46, 267–272.

[6] Zhao, X. M., Xu, Z. G., & Liu, Z. W. (2021). Optimization method for training stability of deep learning models in intelligent transportation systems. *Journal of Xi'an Jiaotong University*, 55, 1–9.

[7] Fang, S. E., Shi, X. F., & Chen, J. (2020). Multi-source data fusion and spatiotemporal modeling technology for urban traffic. *Journal of Tongji University (Natural Science)*, 48, 1197–1206.

Appendix

Appendix A Environment Dependence List

The experiment is implemented in Python, with the following core dependent libraries and version requirements:

- torch>=1.8.0
- numpy>=1.19.0
- matplotlib>=3.3.0
- scikit-learn>=0.24.0

Appendix B Core Code Description

The experimental code is divided into four modules by function: data loading, model, training and tool functions, and the core function of each module is as follows:

(1) **Data loading module**: Realize the loading and preprocessing of multi-view adjacency matrix through the load_global_adj_matrix function; realize the loading, preprocessing and sample generation of grid data and static features through the TrafficRiskDataset class.

(2) **Model module**: Construct MultiViewGCN, ShortTermTransformer and LongTermTCN classes to realize the corresponding model structures respectively, and integrate each module through the TrafficRiskModel class to construct the total prediction model.

(3) **Training module**: The main function realizes dataset division, model initialization, training/test cycle, index calculation, model saving and result visualization.

(4) **Tool functions**: Realize the calculation of classification indicators such as F1, AUC-ROC and AUC-PR through the calculate_metrics function; realize the visualization of index change curves and confusion matrix through the plot_results function.