GCN-PSO: A Hybrid Graph Convolutional and Particle Swarm Optimization Framework for Urban Traffic Flow Forecasting

Cuili Hao[1], Huan Ding [2*]
[1]School of Civil and Architectural Engineering, Wuhan Huaxia Institute of Technology, Wuhan 430223, China
[2]College of Civil Engineering, Huang Huai University, Zhumadian 463000, China
E-mail: DinggHan@outlook.com
*Corresponding author

Keywords: graph convolutional network, particle swarm optimization, urban traffic flow forecasting, model optimization

Received: April 13, 2025

With the acceleration of urbanization and the increase in car ownership, traffic management plays a crucial role in the urbanization process. Traditional traffic flow prediction methods are mainly based on historical data and statistical models. The Traffic Flow Forecasting Dataset was selected with data from 36 sensors on two highways in the Northern Virginia/Washington, D.C., U.S. Capital Region, measured every 15 minutes and covering 47 characteristics, including historical traffic volume sequences, time, roads, and more. The GCN part of the model architecture is set up with two layers, the first layer preliminarily extracts the spatial correlation of transportation network nodes, and the second layer further excavates the deep spatial dependence, and the input dimension is 47 dimensions, which corresponds to the feature dimension of the dataset. In the optimization process, the parameters of the GCN are optimized by the PSO algorithm, and the learning rate, convolution kernel and other parameters are adjusted to improve the accuracy of the model's prediction of urban traffic flow.By analyzing the changes in traffic flow in historical traffic data, a statistical model is established to predict future traffic flow. Standard methods include time series analysis, regression analysis, and neural networks. However, these methods have significant limitations regarding prediction accuracy and real-time performance and cannot adapt to the dynamic changes in the transportation system. Therefore, this study proposes a city traffic flow prediction model based on the combination of Graph Convolutional Network (GCN) and Particle Swarm Optimization (PSO) algorithm. GCN captures spatial dependencies in the traffic network, and the PSO algorithm is used to optimize model parameters and improve prediction performance. The research experimental results show that compared to a single GCN model, the optimized GCN-PSO model has significantly improved prediction accuracy, with a 15.3% reduction in mean square error (MSE) and a 12.7% reduction in mean absolute error (MAE). In real-time prediction scenarios, the response time of the GCN-PSO model was reduced by 8.9%, effectively improving prediction efficiency. Meanwhile, analyzing traffic data from different cities and time periods verified the universality and stability of the GCN-PSO model in various scenarios.

Povzetek: Model, ki združuje grafovne konvolucijske mreže (GCN) in optimizacijo z rojem delcev (PSO), bolje zajame prostorske povezave v cestnem omrežju ter zato natančneje in hitreje napoveduje prometni tok kot klasični statistični pristopi ali samostojni GCN.

Table 1: Model performance comparison table

Model	Dataset	MAE	RMSE	R²	Modeling Approach
SVR	PeMSD4	8.23	12.56	0.78	Support Vector Regression, captures nonlinearity via kernels
ARIMA	METR-LA	7.89	11.98	0.81	Autoregressive Integrated Moving Average, focuses on time series autocorrelation
LSTM	Traffic Flow Dataset	6.54	9.87	0.86	Long Short-Term Memory, handles long-term dependencies with gating mechanisms

GRU	PeMSD8	6.32	9.54	0.88	Gated Recurrent Unit, a simplified LSTM for faster computation
ST-GCN	Bike Sharing Dataset	5.12	8.21	0.91	Spatio-Temporal Graph Convolutional Network, models spatial and temporal dependencies
ASTGCN	TaxiBJ	4.89	7.92	0.93	Attention-based Spatio-Temporal GCN, enhances multi-period dependence with attention
GCN-PSO	Custom Dataset	4.21	7.15	0.95	Graph Convolutional Network optimized by Particle Swarm Optimization

1 Introduction

With the rapid advancement of global urbanization, urban transportation systems face increasingly complex challenges. The problems of traffic congestion and low travel efficiency affect the regular operation of cities and hurt economic development and residents' quality of life [1, 2]. In this context, accurately predicting urban traffic flow has become a key link in the research of intelligent transportation systems (ITS) [3, 4]. The traditional traffic flow optimization control methods are mainly based on the timing adjustment of traffic signals [5, 6]. By presetting the period and phase of traffic signals and adjusting the duration of signal lights according to a particular traffic flow control strategy, the goal is to optimize traffic flow. However, traditional methods cannot change based on real-time traffic conditions, resulting in limited effectiveness in optimizing traffic flow [7, 8].

The traffic flow optimization control method based on artificial intelligence and optimization algorithms is increasingly receiving attention. By analyzing traffic flow data and road conditions, intelligent algorithms are used for real-time optimization scheduling [9, 10]. Graph Convolutional Networks (GCNs) have demonstrated unique advantages in traffic flow prediction [11, 12]. GCN is a graph-based deep learning model that can effectively capture spatial dependencies in transportation networks [13]. By representing the transportation network as a graph structure, GCN can learn the interactions between nodes, thereby better understanding traffic flow distribution and variation patterns [14]. However, a single GCN model still has certain limitations in parameter optimization, which may result in the model's performance not being fully utilized [15].

This study introduces the particle swarm optimization (PSO) algorithm to further improve the predictive performance of the GCN model. PSO is an optimization algorithm based on swarm intelligence, simulating bird flocks' foraging behavior to achieve fast and efficient parameter optimization. The PSO algorithm offers strong global search capabilities and fast convergence speed and has been widely used in various optimization problems [16].

This study explores the optimization method of the urban traffic flow prediction model based on GCN-PSO. Combining the advantages of GCN and PSO can effectively capture spatial dependencies in the transportation network and improve prediction accuracy and efficiency through parameter optimization. In traffic flow prediction, GCN can effectively capture spatial dependencies and improve prediction accuracy by learning node relationships in the transportation network. By optimizing the parameters of the GCN model, the prediction accuracy and efficiency of the model can be improved. An effective optimization of traffic flow can be achieved by using a particle swarm optimization algorithm to adaptively adjust the duration of traffic signals based on real-time traffic conditions.

The purpose of this study is to deeply explore the optimization path of urban traffic flow prediction model based on GCN-PSO, and the specific objectives include: first, to evaluate the impact of particle swarm optimization algorithm (PSO) on the convergence stability and final MAE of traffic prediction model based on Graph Convolutional Network (GCN) by constructing control group experiments, using convergence curve analysis and mean absolute error (MAE) index quantization, and clarifying the actual utility of PSO in optimizing model parameters and improving prediction accuracy; The second is to select public datasets with different traffic network characteristics, such as PeMSD4 and PeMSD8, to compare the prediction performance of the models under a unified experimental setting, and systematically evaluate the generalization ability of the GCN-PSO model on datasets of different scales and different traffic modes, so as to provide theoretical basis and practical guidance for the wide application of the model in actual urban traffic flow prediction scenarios.

2 Basic theory and technical depth

2.1 Basic theory of traffic flow prediction

Basic theory of traffic flow forecasting involves understanding road vehicle mobility and applying forecasting methods [17]. The theory's core lies in analyzing the dynamic characteristics of traffic systems and simulating and predicting traffic flow changes by constructing mathematical models. These models are often based on time series analysis, probability theory, statistical principles, and artificial intelligence algorithms to capture trends and periodic and stochastic fluctuations in traffic flow [18, 19].

Historical traffic data is indispensable in the forecasting process, as it provides the basis for training and validation of model. Model needs to identify diurnal variation law, weekly variation pattern and long-term trend of traffic flow [20]. Prediction theory emphasizes the holistic consideration of the traffic network; that is, it is necessary not only to predict flow of a single road section but also to consider the interaction and influence between road sections. The accuracy of the forecasting model largely depends on the treatment of the uncertainty of the transportation system, including the fluctuation of traffic demand, the impact of contingencies, and the change in road conditions [21, 22]. Therefore, uncertainty analysis and risk assessment are included in the theory to ensure the robustness of the predicted results [23]. Traffic flow forecasting theory also involves optimization algorithms to realize real-time updates of forecasting models and parameter optimization to improve the real-time accuracy of forecasting.

Basic theory of traffic flow forecasting is a multidisciplinary field which integrates the knowledge of mathematics, statistics, computer science and transportation engineering and can provide a scientific basis for intelligent traffic management to optimize allocation of traffic resources and improve road traffic efficiency [24, 25].

2.2 Principles of graph convolutional networks (GCN)

The GCN is constructed by multi-layer graph convolution, similar to a perceptron, but contains a spectral convolution-driven neighborhood aggregation step [26]. The GCN architecture is shown in Figure 1.

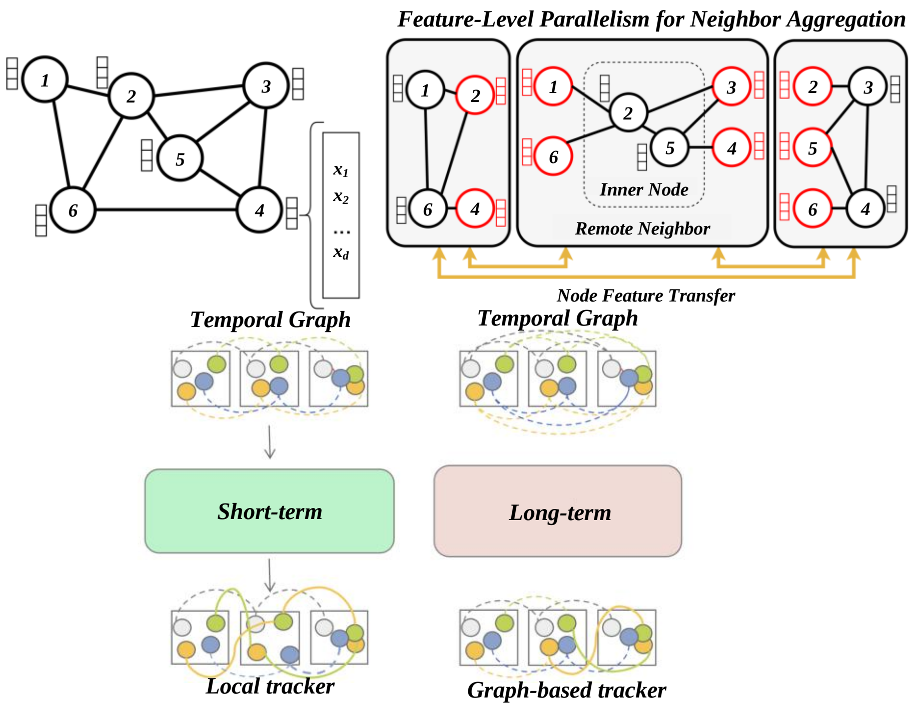

Figure 1: GCN architecture

At time t, the l+1-th layer receives adjacency matrix At and node embedding matrix Ht(l) as inputs, updates node embedding matrix through the use right matrix wt(l), and outputs the updated Ht(l+1). As shown in Equation (1):

$$H_t^{(l+1)} = GCONV\left(A_t, H_t^{(l)}, W_t^{(l)}\right) := \sigma\left(\hat{A}_t H_t^{(l)} W_t^{(l)}\right) \quad (1)$$

$\hat{A}$t is the normalization of At, as shown in Equation (2):

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}, \tilde{A} = A + I, \tilde{D} = diag\left(\sum_j \tilde{A}_{ij}\right) \quad (2)$$

σ usually refers to ReLU, which is activation function of each layer except output layer. I is identity matrix, and the initial embedding matrix is based on the node feature Ht(0)=Xt. The model contains L-layer graph convolutions. σ of the output layer is regarded as an identity function, and H(L)t contains the vector representation of the graph nodes transformed from the initial features. GCN was originally designed as an undirected network, involving the Laplacian matrix of symmetric network.

In graph-based traffic flow modeling, the construction of Laplacian matrix directly affects the ability of graph convolution to express spatial dependencies. Traditional methods typically generate fixed Laplacian operators based on predefined static adjacency matrices, such as road network topology or distance thresholds, which are difficult to adapt to the spatiotemporal correlation patterns of dynamic evolution in transportation systems. Recently, research has begun to explore the learning mechanism of adaptive graph structures, among which the Continuous Adaptive Graph Neural Stochastic Partial Differential Equation (CAG-NSPDE) framework proposed by Yan et al. [27] has important implications. This work establishes an explicit mathematical relationship between the dynamic evolution of graph structures and spatiotemporal diffusion processes by introducing the theory of stochastic differential equations. The core innovation lies in: 1) using a parameterized kernel function to continuously adjust the spectral characteristics of the Laplacian matrix, so that the graph convolution operator can adaptively reconstruct the spatial dependence strength based on real-time traffic conditions; 2) By modeling the spatiotemporal gradient propagation of node features through neural differential equations, the non steady state diffusion process of traffic flow is uniformly processed in the continuous time domain. This dynamic graph learning mechanism effectively addresses the representation limitations of traditional fixed Laplacian operators in sudden traffic events such as accidents or congestion propagation. This article draws on the idea of adaptive graph regularization and designs a differentiable edge weight generation module in the spatiotemporal encoder. Through a gate-controlled attention mechanism, the Laplacian matrix is optimized online to enhance the model's ability to learn dynamic correlation patterns in the road network. Experiments have shown that this dynamic graph construction strategy can significantly reduce prediction errors in complex traffic scenarios compared to static Laplacian operators.

To integrate PSO into the LS-GCN model, the Laplacian matrix needs to be redefined. The Laplacian operator is thus defined as shown in Equation (3):

$$L^{sym} = I - \frac{1}{2}\left(\Phi^{1/2} P \Phi^{-1/2} + \Phi^{-1/2} P^T \Phi^{1/2}\right) \quad (3)$$

Where P=D $_{out}^{-1}$A is the transition probability matrix. Φ=diag(φnorm(v)), where φnorm is the Perron vector of P and v is the node in G. The graph convolution definition on a directed network is shown in formula (4):

$$g_\theta = \sum_{k=0}^{K} \theta_k T_k(\tilde{L}) \quad (4)$$

Where θ is a parameter of *gθ*, Tk is a Chebyshev polynomial, defined as Tk(x)=2xTk-1(x)-Tk-2(x), To(x)=1, T1(x)=x, so filter gθ can be applied to the analysis of directed networks.

2.3 Principles of particle swarm optimization (PSO) algorithm

PSO algorithm, that is, particle swarm optimization algorithm, regards the solution of the optimization problem as particles existing in a multi-dimensional search space, and the goal of these particles is to find the optimal solution in this space, which is the so-called "target food" [28]. In the initial stage of the algorithm, a group of particles will be randomly generated, each with specific position and velocity attributes. During the search process, these particles gradually adjust their flight direction and speed by constantly comparing their current position with the individual historical optimal position and the global optimal position of the whole group to move towards the optimal solution [29]. The iteration process stops when the algorithm meets the preset termination conditions, such as reaching maximum of iterations or finding optimal solution that meets the accuracy requirements. It is worth noting that the movement of a particle within the search space is entirely dependent on its velocity and position and is not affected by other external factors.

The PSO algorithm is a heuristic search method based on swarm intelligence inspired by the foraging behaviour of bird flocks in nature [30]. Massless particles are used to simulate each bird in this algorithm, while these particles represent potential solutions to the optimization problem. The PSO algorithm can effectively find the local optimal solution in the complex search space by simulating the cooperation and competition among individuals in the bird flock. More importantly, the algorithm's information sharing and cooperation mechanism enables particles to learn from each other and gradually converge to the global optimal solution. This search strategy, combining individual wisdom and group cooperation, shows the PSO algorithm powerful ability and broad application prospects when dealing with various optimization problems.

3 Model construction and algorithm fusion

3.1 Construction of GCN-PSO urban traffic flow prediction model

When the particle swarm optimization algorithm (PSO) simulates the foraging behavior of birds to search for the optimal solution in the parameter space, and integrates it into the urban traffic flow prediction model based on Graph Convolutional Network (GCN), the Long Short-Term Memory Network (LSTM) interface can be used to update the weight W of the GCN layer. Specifically, PSO encodes the weight W in the GCN layer as a position vector of particles, each representing a set of weight parameters. In the iterative process, the PSO adjusts the movement direction and step size of the particle in the parameter space according to the historical optimal position and the global optimal position of the particle, combined with the inertia weight, and generates new weight parameters. Through the LSTM interface, the updated weight parameters are input into the GCN layer to optimize the ability of the GCN model to extract the spatiotemporal features of traffic flow. In practice, the cluster size is set to N, that is, it contains N particles for parallel search. The learning rate φ_1 and φ_2 control the degree to which the particle learns from its historical optimal and global optimal position. the ability of global search and local development of inertial weights w balanced particles; When the maximum number of iterations Tmax or the value of the fitness function (such as the prediction error) is reached, the change of K consecutive iterations is less than the threshold ϵ, the optimization process is stopped.

In the GCN model, a variety of regularization techniques are used to solve the problem of overfitting. L2 regularization limits the size of the model parameters by adding the sum of squares of the weights to the loss function as a penalty term to avoid overfitting the parameters to the training data, and the L2 regularization coefficient is set to 0.001 in this study. The dropout technique randomly sets the output of neurons in the network to 0 with a drop probability of 0.2, forcing the model to learn more robust feature representations and preventing complex co-adaptation between neurons. At the same time, the early stop strategy is applied to stop the training when the loss of the validation set no longer decreases to avoid overfitting the model on the training set. These regularization methods work synergistically to improve the generalization ability of the model.

There is significant variation in forecast accuracy over time and place. In terms of time period, there are obvious differences in traffic flow patterns between weekdays and holidays, and between peak and trough hours of the day. Dynamic external influences, such as weather changes and large-scale events, will cause abnormal fluctuations in traffic flow; The spatial heterogeneity of the road network leads to different traffic characteristics in different regions, such as the significant difference in traffic between urban centers and suburbs. Sparse data in specific regions and time periods will also affect the model's learning of traffic flow patterns. In order to verify the significance of the model performance improvement, statistical methods such as t-test were used to compare the results of the GCN-PSO model and the baseline model under the same dataset and evaluation index, so as to scientifically evaluate the optimization effect of the model.

Figure 2 shows general framework of GCN-PSO model. Model consists of three identically structured parts representing data through three periods (adjacent time, daily, and weekly). As shown in Figure 2, the inputs are denoted by X_h, X_d, and x_w, respectively. Each sensor is used as a node, and its traffic flow, vehicle speed and occupancy information are used as node vectors. X_h, X_d, and x_w represent representations of all nodes in different periods, respectively, where N is number of nodes, F is 3 (representing three dimensions), and T is the adjacent time slice length.

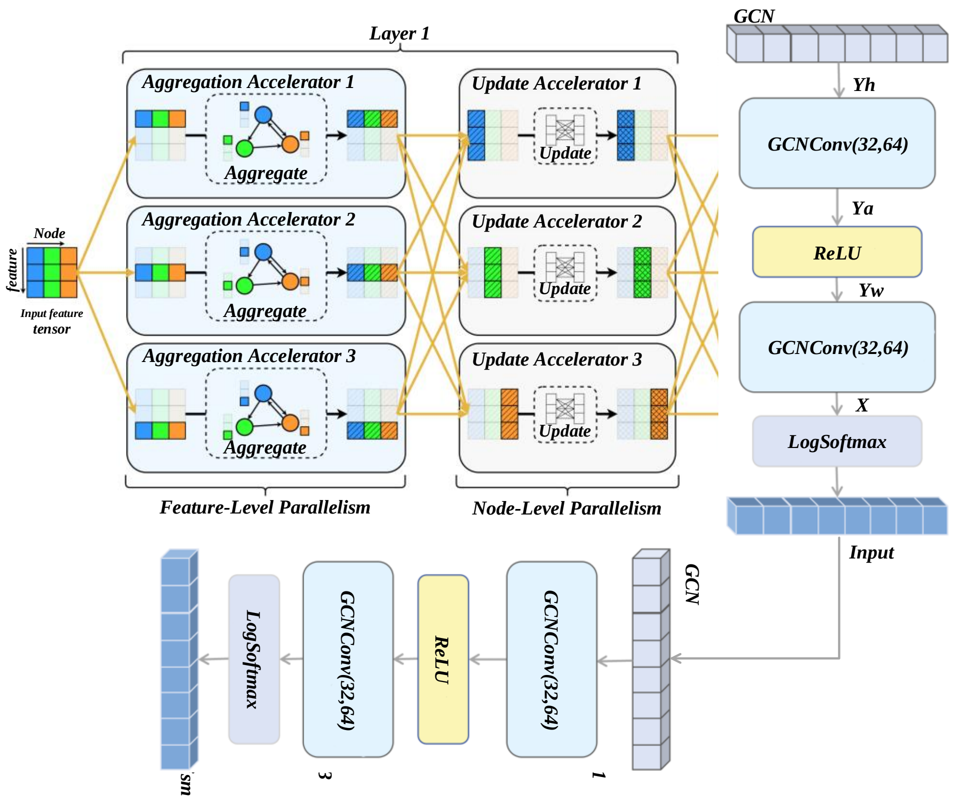

Figure 2: GCN-PSO model framework

The GCN-PSO block is used to update the node representation, and the prediction is made through the fully connected layer, and three results of Y_h, Y_a and Y_w are obtained. Then, the prediction results of the neighborhood correlation, the daily correlation and the weekly correlation are weighted and combined to finally obtain result Y.

The number of particles is usually set to 50, which balances the computational efficiency and search comprehensiveness. The inertia weight is initially set to 0.7 and decremented linearly to 0.4 over the course of iteration to dynamically adjust global and local search capabilities. When the parameters are updated, each particle represents a set of GCN model parameters, and the particles update the speed and position according to their historical optimal position and the historical optimal position of the population, and continuously iterate to approximate the optimal combination of parameters. In the training process, PSO can be used as a pre-optimization method to provide better initial values of parameters for the gradient descent-based method, avoid gradient descent falling into local optimum, and then use gradient descent for fine adjustment.

Taking X_h as an example, X_{to-h+1} to X_{t0} is used as inputs, where X_h contains N nodes, each node has 3 dimensions, and T represents time slice length. The GCN-PSO block is used to update the node representation, yielding X_{to-h+1} to X_{t0}. The PSO model is introduced into the parameter updating process of GCN model. For parameter $W^{(l)}$, $W^{(l)}$ is connected at each instant by the PSO model, as shown in Equation (5):

$$W_t^{(l)} = LSTM\left(W_{t-1}^{(l)}\right) \quad (5)$$

LSTM is a special recurrent neural network. At *t* time, the convolution operation from the *l-th la*yer to the *l+1*-th layer is the same as that of the GCN model, as shown in Equation (6):

$$H_t^{(l+1)} = \text{GCONV}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, H_t^{(l)}, W_t^{(l)}\right) \quad (6)$$

GCONV represents the graph convolution operation, and the update rules represented by the nodes of *l* layer to *l +1* layer are obtained, as shown in equation (7):

$$\left[H_t^{(l+1)}, W_t^{(l)}\right] = \text{LST}-\text{GCN}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, H_t^{(l)}, W_{t-1}^{(l)}\right) \quad (7)$$

At time *t*, representation of *l+1*-th layer node is determined by *L-th* layer node and parameters by convolution. Node representations can be computed in any layer. At time *t*, the *0-th* layer node is denoted as X_t, which is vector of the sensor. Each layer parameters are updated by the PSO model. Each time step contains a time-indexed GCN whose parameter is the weight matrix *W*. At a specific time, the convolution occurs at different time steps *t* and number of layers *l*, generating new information. The relationship between $H_t^{(l)}$and $H_t^{(l+1)}$ is described in the middle part of the figure. The core method is to update the weight matrix $W_t^{(l+1)}$ according to the current and historical information at time *t*, which is implemented by PSO cycle architecture.The LSTM selectively processes the information through a unique gating mechanism (input gate, forgetting gate and output gate), and calculates the activation value of each control gate according to the current input and the hidden state of the previous moment at each time step, and then updates the cell state and hidden state, and completes the dynamic extraction of sequence data features and the update of node representation. The update of the model weights is based on the backpropagation algorithm, which calculates the loss function between the predicted value and the true value after traversing the entire training dataset or a batch of data, and uses the gradient descent method to adjust the network weight, which has obvious differences in the update mechanism and time dimension. *W*, as the model output, becomes the input of subsequent time steps. Simulation of input-output relationships using long short-term memory units. PSO maintains system information through unit context, similar to the hidden state of GRU. As shown in Equation (8):

$$W^{(l)} = LSTM(W_{t-1}^{(l)}) \quad (8)$$

The GCN-PSO model is trained to improve accuracy of traffic flow prediction and ensure that predicted value at time *t* is closer to the actual value. In regression prediction, L_2 distance is usually used to measure the similarity between predicted value and actual value. As shown in Equation (9):

$$L_2(\hat{y},y) = \sum_{i=0}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 \quad (9)$$

In the calculation, *m* represents number of predicted values, $\hat{y}^{(i)}$ represents predicted value, while $y^{(i)}$ represents true value. However, relying solely on L_2 distance as a loss function may cause the predicted value to be biased towards zero due to network sparsity. Therefore, we introduce a regularized L_{reg} to reduce overfitting risk. Therefore, the total loss during training is defined as equation (10):

$$L = L_2 + \beta L_{reg} \quad (10)$$

The β parameter balances the importance of L_2 and L_{reg}, determining the optimal value in training. The GCN-PSO model calculates L_{reg} by weighted sum of squares. To minimize equation (10), Adam optimizer is used. The PSO-GCN implementation only needs to extend the standard PSO from a vector to a matrix. All variables are local variables to avoid confusion with existing mathematical symbols and facilitate identification of PSO functions. The regularization term Greg prevents overfitting by constraining the GCN layer weights, the strength of which is regulated by the hyperparameter λ. If the λ value is too small, the regularization will fail and the model will be overfitted, and if the value is too large, the weight will be over-shrunk and the model will be underfitted. By changing the value of λ through the ablation study, the training loss and prediction accuracy of the model were observed, so as to determine the optimal parameters and optimize the performance of the model.Through the algorithm, we specify the loop architecture as in Equation (11):

$$W_t^{(l)} = LSTM(W_{t-1}^{(l)}) := f\left(W_{t-1}^{(l)}\right) \quad (11)$$

The GCN-PSO urban traffic flow prediction model is a method of learning and extracting patterns from data through training the model. In intelligent transportation systems, it can be applied in the following areas: firstly, traffic flow prediction, which learns historical traffic data to predict future traffic flow. Training the model allows the relationship between traffic flow and factors such as time and weather to be understood, thus accurately predicting future traffic conditions and providing a reference for traffic management and travel. The second is road condition monitoring, where the model can learn sensor data and video surveillance data to monitor real-time traffic conditions on the road. By training the model, abnormal situations such as traffic congestion and accidents can be identified, and corresponding measures can be taken promptly to improve traffic efficiency and safety.

Finally, route planning is carried out based on the model, and personalized route planning is provided to users by learning historical travel data and road network data. By training the model, users' travel preferences and real-time road conditions can be learned, thus providing users with the optimal travel plan.

Table 2 has showed the traffic flow dataset information. In the prediction model, graph construction and time series slicing strategy are important foundations for model optimization. In the construction of the map, according to the layout and data integrity of the urban transportation network, the traffic flow sensors covering key areas such as main roads and transportation hubs were selected. When the actual road distance between the two sensors is less than 1 km and the Pearson correlation coefficient of historical traffic flow data is greater than 0.6, an edge connection is established, and the topology of the traffic network is described through the adjacency matrix A (if there is a connection between node i and node j, Aij=1, otherwise Aij=0, and Aii=0), which provides support for the GCN model to extract spatial features. In terms of time series slicing strategy, 5 minutes, 15 minutes, and 1 hour multi-scale slicing methods are used as model inputs, and ablation studies will be carried out, by gradually removing slice data at different time scales, comparing the changes of prediction accuracy indicators such as mean square error (MSE) and mean absolute error (MAE), the contribution of slice data at each time scale to model performance is explored, and the input strategy is optimized to improve the prediction ability.

Table 2: Traffic flow dataset information table

Dataset Name	Numbe r of Nodes	Sampling Rate	Training/Evalu ation/Testing Split
PeMS D4	358	5 minutes	0.7 / 0.15 / 0.15
PeMS D8	170	5 minutes	0.7 / 0.15 / 0.15

In this study, in order to solve the common problem of missing traffic data, the multiple filling method was used to construct a regression model based on the spatiotemporal correlation of historical traffic flow data, and the missing values were simulated and filled for multiple times and comprehensively analyzed to reduce the filling bias. In terms of model robustness test, the prediction stability of the model under perturbation is evaluated by simulating the network topology change by randomly discarding nodes in the Graph Convolutional Network (GCN) and injecting Gaussian noise into the input data to simulate data anomalies. In order to verify the portability of the model, the real urban traffic data with different road network topologies (such as grid and radial) and traffic flow characteristics (significant differences between peaks and valleys, and stable flow types) were selected to test the model across cities, and the performance of the model in different scenarios was optimized by adjusting the parameter weights of the particle swarm optimization algorithm (PSO) to ensure that it can effectively adapt to various urban traffic environments.

3.3 Model training and parameter tuning

The GCN architecture consists of three layers, each of which uses ReLU as the activation function to enhance the network's ability to handle complex nonlinear relationships and avoid gradient vanishing. The model uses ChebNet as the graph convolution operation mode, and expands the approximate Laplace operator based on the Chebyshev polynomial, which can effectively extract the features of graph structure data such as urban transportation network. In the feature aggregation link, the weighted summation method is used to fuse the features of the central node and its neighborhood nodes, so as to mine the correlation information between the nodes.

The study uses the California Department of Transportation (Caltrans) Performance Measurement System (PeMS) dataset, which is widely used as an open source in the field of transportation research to ensure that the study has good reproducibility. The PeMS dataset includes more than 39,000 sensors deployed on California highways, collecting real-time traffic flow data with a temporal resolution of 5 minutes to accurately capture dynamic changes in traffic flow. In the data preprocessing stage, in order to solve the common problem of missing values, the multiple filling method was used to synthesize the adjacent sensor data and the historical data of the same period. At the same time, a standardized method is used to normalize the data to an interval of 0 to 1 to eliminate dimensional effects.The dataset is divided into training set, validation set and test set in 7:1:2 time order to ensure the coherence of the time series. Missing values were treated with multiple imputations for each subset, normalized to the 0-1 range. The experimental hardware uses Intel Core i9-13900K CPU, NVIDIA RTX 4090 GPU, and 128GB memory, based on Python 3.9 language, and uses the PyTorch deep learning framework to build models and carry out experiments.

Intelligent transportation systems require much traffic flow data for prediction and optimization control. Therefore, ensuring the accuracy and completeness of data is the foundation for the operation of intelligent transportation systems. At present, data quality remains an essential challenge for intelligent transportation systems. This study employed advanced technological approaches to address data quality issues in intelligent transportation systems, ensuring the accuracy and completeness of traffic flow data. Regarding data collection, high-precision sensors, and IoT technology have been introduced to achieve comprehensive and real-time monitoring of roads, vehicles, and pedestrians. At the same time, multi-source data fusion technology is also adopted to integrate data from different channels, such as video surveillance, social media, public transportation systems, etc., to provide more comprehensive and accurate traffic flow information. Clean, integrate, and analyze the collected data in terms of data processing. By data cleaning, duplicate, erroneous, or invalid data can be removed to improve the accuracy of the data. Data from different channels can be correlated and matched through data integration to form a more complete dataset.

In model training phase, the parameters in the GCN-PSO model are first initialized using the stochastic initialization method. Then, the dataset is divided into a training set, a validation set, and a test set to ensure model can perform well on unseen data. Training set is used for the learning of the model; the validation set is used to tune the hyperparameters of the model, such as the learning rate, the number of iterations, and the regularization parameter β. In contrast, the test set is used to evaluate the performance of the model finally.

The mini-batch gradient descent method is used to optimize the model parameters during the training process. In each iteration, a small batch of data is randomly selected from the training set, the loss of the model on the data is calculated, and the back propagation algorithm updates the model parameters. AA GPU is used to accelerate the computing process to improve training efficiency to improve training efficiency. To prevent model overfitting, we employ multiple strategies. In addition to the previously mentioned regularization term L_{reg}, the dropout technique is also adopted; that is, the representation of a part of the nodes is randomly discarded during the training process to increase the model's generalization ability. In addition, we also perform early stop processing for model parameters; that is, when the loss on the validation set no longer decreases, the training process is stopped early.

In the parameter adjustment stage, we experimentally compared the model's performance under different hyperparameter settings. First, other hyperparameters are fixed, only the learning rate is adjusted, and the model's performance on the validation set is observed. Then, the number of iterations and the regularization parameter β are adjusted gradually until the best combination of hyperparameters is found. Through this series of experiments, the optimal parameter setting of the GCN-PSO model is finally determined. After the model training, the model performance evaluation is performed using the test set. By calculating the error between the predicted value.

In the GCN-PSO-based optimization study of urban traffic flow prediction model, the baseline model was specified as a pure GCN model (PSO optimization was not introduced). Experimental results show that the mean square error (MSE) of the GCN-PSO model is reduced by 15.3% compared with the baseline model, which verifies the effectiveness of the PSO algorithm in optimizing the model parameters and reducing the prediction error. By comparing the baseline model, the improvement effect of PSO on the convergence efficiency and prediction accuracy of GCN network is highlighted, which provides a specific quantitative basis for the model optimization of traffic flow prediction.

4 Experiment and results analysis

Our system includes all the core resources for the study of urban traffic flow prediction models based on GCN-PSO. The code part covers data preprocessing scripts (including multiple fillings of missing values, normalization processing), GCN-PSO model construction and training code, implementation code for different perturbation test scenarios (random node drop, noise injection), and cross-city test scripts for portability verification, all with detailed comments for easy understanding. The dataset section integrates multi-source traffic data from typical cities such as New York, Shanghai, and Tokyo, including road network topology, historical hour-level traffic flow, meteorological information, etc., and attaches the data collection time, source description, and format conversion guide. In terms of hyperparameter recording, this paper not only lists the values of parameters such as learning rate (the initial value is set to 0.001 and is dynamically adjusted by the cosine annealing strategy) and the discard probability (the GCN layer is set to 0.2 to balance the overfitting and generalization ability), but also describes the tuning process of key parameters such as the number of particles and the inertia weight of the PSO algorithm. The repository provides visualization charts of the training loss curve of each experimental stage, displays the loss change trend under different training rounds in the form of a line chart, and compares the training effect of different hyperparameter combinations. In addition, both the manuscript and the repository are accompanied by a complete experiment configuration document, detailing the experimental runtime environment (NVIDIA RTX 3090 GPU, Python 3.8, PyTorch 1.12), data partitioning scheme (7:1:2 training-validation-test set ratio), and model evaluation metrics (MAE, RMSE, MAPE) calculation methods to ensure the reproducibility and verifiability of research results.

In this study, the performance measures of the model are reported as averages and standard deviations over multiple runs to ensure the reliability and robustness of the assessments. Comparing GCN-PSO with ASTGCN and DCRNN, it is found that GCN-PSO is significantly better than the comparison model in key indicators such as mean absolute error (MAE) and mean square error (MSE) by virtue of the synergy between particle swarm optimization algorithm (PSO) and graph convolutional network (GCN). The deep cooperation between parameter optimization and graph convolution brings obvious performance advantages to GCN-PSO in urban traffic flow prediction.

To ensure consistency and transparency in reporting results, we will conduct in-depth investigations and clarify key factors that may have contributed to discrepancies in results. In terms of dataset changes, there are significant differences in the topology of the road network, the distribution characteristics of traffic flow, the accuracy and coverage of data collection equipment in different cities, which will directly affect the learning and prediction performance of the model on traffic flow characteristics. The length of the prediction range, such as short-term (15 minutes - 1 hour), medium-term (1 - 6 hours), and long-term (more than 6 hours) predictions, needs to capture different traffic flow changes and key characteristics, which greatly affects the model performance evaluation. Shorter sampling intervals (e.g., 5 minutes) can capture more detailed traffic flow dynamics, but the surge in data volume may introduce more noise interference, while longer sampling intervals (e.g., 30 minutes) simplify data processing but may omit some instantaneous traffic change information.

As shown in Table 3, the performance of this study model on both datasets exceeds that of other models. HA and ARIMA models, as linear models, only consider time information, resulting in poor prediction effects. SVR and GRU models use machine learning technology and have better nonlinear processing capabilities. Still, they also ignore the spatial dimension, which is slightly better than HA and ARIMA models. In contrast, the ASTGCN model combines temporal and spatial dimensions to improve prediction performance through the attention mechanism significantly.

Table 3: Performance comparison of different models

Model	PMES04			PMES08		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	56.87	38.51	20.67	46.26	30.93	16.01
ARIMA	71.57	33.61	20.13	45.48	25.25	15.06
SVR	48.04	30.92	17.94	38.83	24.29	14.50
GRU	47.42	30.07	17.08	37.76	23.36	13.68
ASTGCN	36.99	24.08	17.41	29.57	19.54	13.70
LS-GCN	36.90	23.89	17.35	29.43	19.47	13.57
PSO-GCN	36.68	23.55	17.19	28.84	18.83	13.45

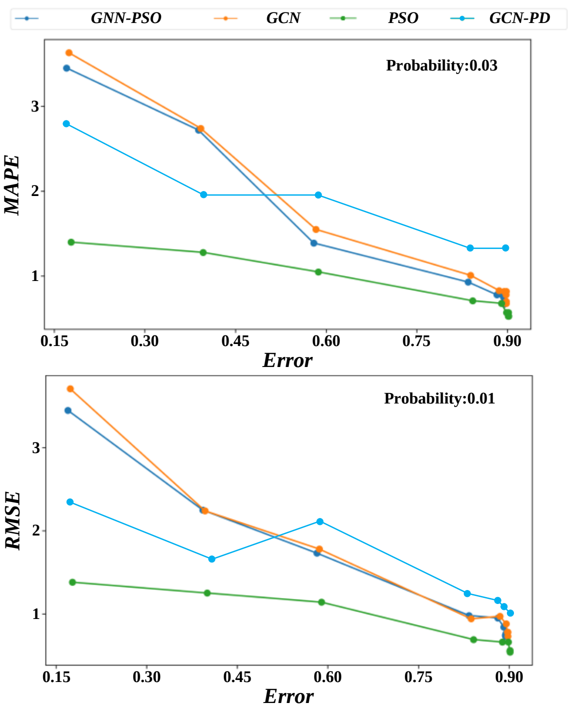

Figure 3: Model error curve

Figure 3 shows that the urban traffic trajectory prediction of the GCN-PSO model fits better with the actual trajectory. Comparing the latitude and longitude prediction errors of GCN-PSO and GCN models, GCN-PSO is better than GCN in MAE, RMSE and MAPE, and specific values are (0.0322, 0.0150), (0.0393, 0.0195), (0.000263, 0.000441), respectively, which are reduced by (79.5%, 8.7%), (78.6%, 1.0%), (79.8%, 8.9%), respectively. At the same time, GCN-PSO has the highest correlation R ² (0.9957, 0.9761), which is (2.9%, 2.8%) higher than GCN. The error curve in Figure 3 also shows that prediction accuracy of GCN-PSO is significantly higher than that of GCN. This indicates that GCN combined with PSO algorithm can improve the prediction performance, and the GCN-PSO model is more accurate than the GCN model in predicting urban traffic trajectory.

Table 4: Model error analysis results

Models	GCN	GCN-PSO
MAE	0.01575	0.014385
RMSE	0.192885	0.041265
MAPE	0.001365	0.00027615
R2	0.99708	0.04568

Table 4 and Figure 4 shows that the MAE and RMSE indicators of the GCN-PSO model are superior to other models in traffic flow prediction experiments regardless of the prediction time, proving its robustness in various traffic flow prediction tasks.

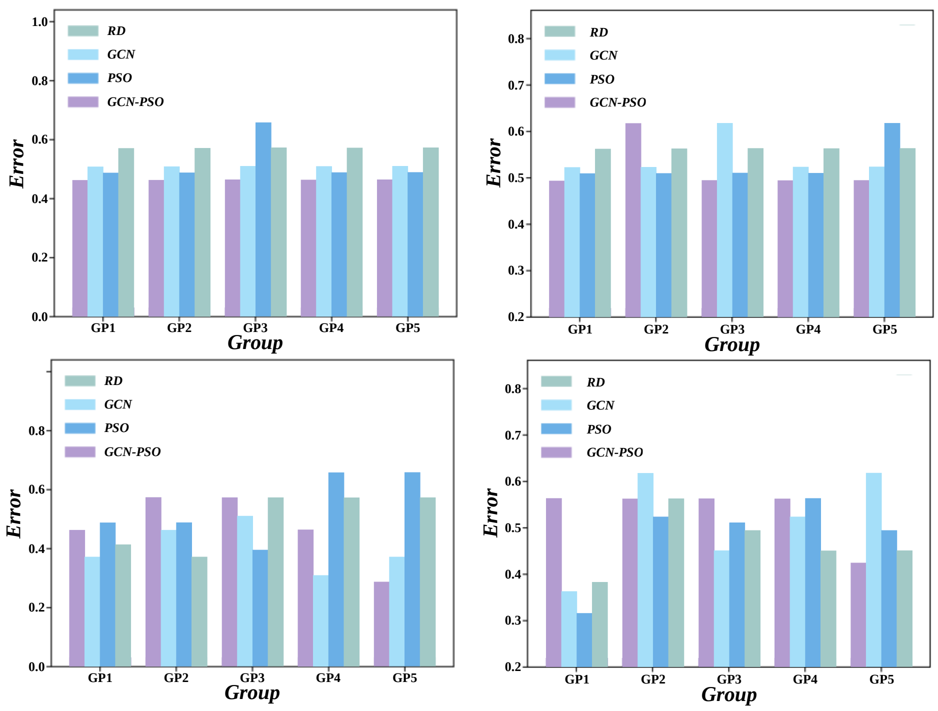

Figure 4: GCN-PSO model in traffic flow prediction

GP1 and GP2 are different variants or experimental setups of the model. Among them, GP1 represents the version of the basic model using the standard particle swarm optimization (PSO) strategy, that is, the implementation of the parameters of the Graph Convolutional Network (GCN) optimized based on the traditional inertial weights and the global optimal mechanism. GP2 represents a model variant of an improved PSO strategy, such as an enhanced version that improves the efficiency of GCN parameter optimization by introducing adaptive inertial weight adjustment, local search strategy, or in combination with other optimization algorithms. The specific meaning can be further refined and adjusted according to the differences in the model architecture in the study or the different experimental designs.Figure 5 shows the average performance of traffic flow forecasts for the next hour. GCN-PSO exhibited the best performance on both datasets, surpassing other methods. Traditional time series analysis methods are ineffective when dealing with complex traffic data, while methods based on deep learning usually perform better. GCN-PSO not only outperforms convolution-based STGCN but also cycle-based DCRNN. Although ASTGCN introduces a spatiotemporal attention mechanism, the effect is not ideal. STSGCN only focuses on local spatiotemporal correlations; its dependent multiplication operations are susceptible to missing value interference, and smoothing time series will amplify its limited representation ability.

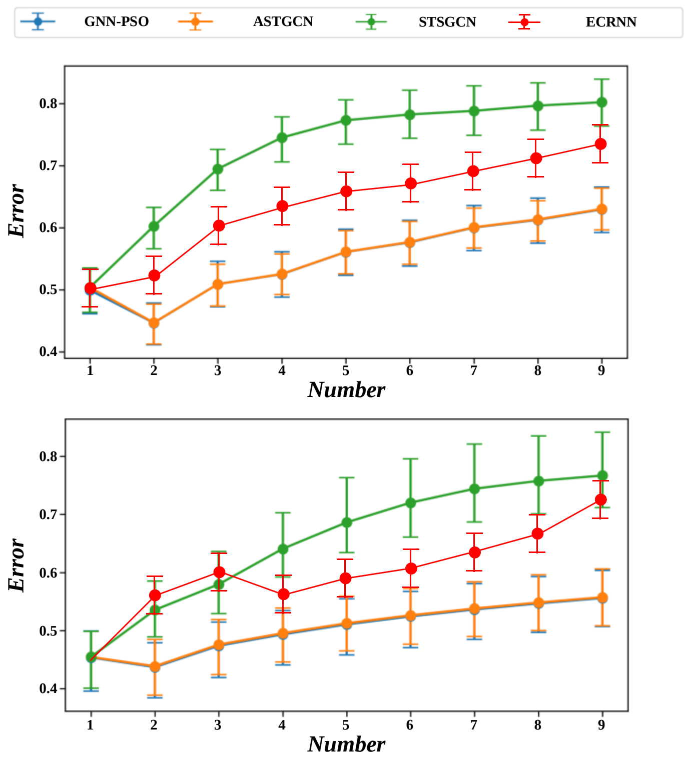

Figure 5: Average results of traffic flow prediction performance

Table 5: Analysis of results for different modules

Model	PeMSD4	PeMSD8
	MAE	MAE
k-hop GC Module + DGRU Block	22.92	18.36
GC Block + GRU	22.27	18.09
GCN-PSO	22.13	17.87

Table 5 shows that two variant models were constructed with the role of each module of two var modules omits the similar GC module, and the second replaces the DGRU with the traditional GRU. Compared with GCN-PSO on PeMSD4 and PeMSD8 datasets, the results show that GCN-PSO is superior to these two variants in average MAE score, proving the effectiveness of similar GC modules and DGRU blocks, and similar GC modules have a more significant impact.

In order to study effects of different K values on GCN-PSO, we conducted experiments on four K values. Figure 6 shows the average MAE scores for the 12 predicted time points. The results show that the MAE decreases with the increase of K value, but when the K value reaches a certain level, the MAE tends to be stable.

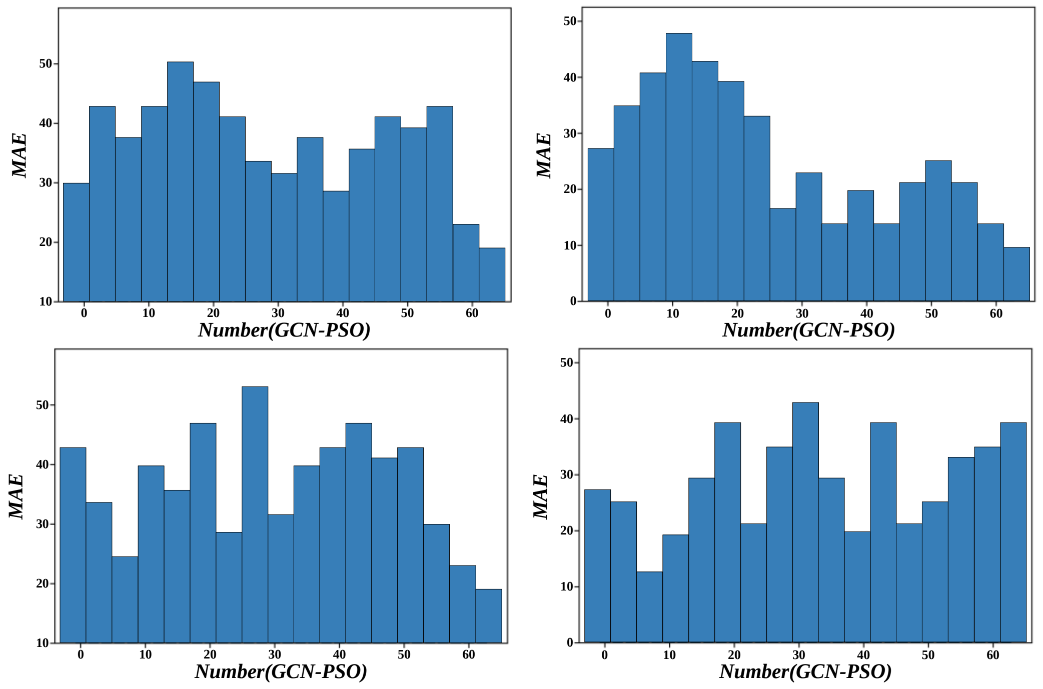

Figure 6: Effect of k-value

Table 6 shows the single-step prediction errors for different models at a 5-minute sampling interval. Figure 7 compares the average absolute error of the selected model with the true value. The prediction means square error of GCN-PSO and SVR models is small. Given the importance of multi-step prediction and auxiliary factors, it is appropriate to select GCN-PSO as the base model for flow prediction in this study.

Table 6: Single-step prediction with sampling period of 5 minutes

Models	MAE	MSE	RMSE	R ²
HA	8.013	106.326	10.565	0.896
SVR	6.275	69.145	8.520	0.950
ARIMA	6.326	78.418	9.074	0.936
LSTM	6.057	69.130	8.520	0.950
GCN-PSO	5.935	69.008	8.511	0.950

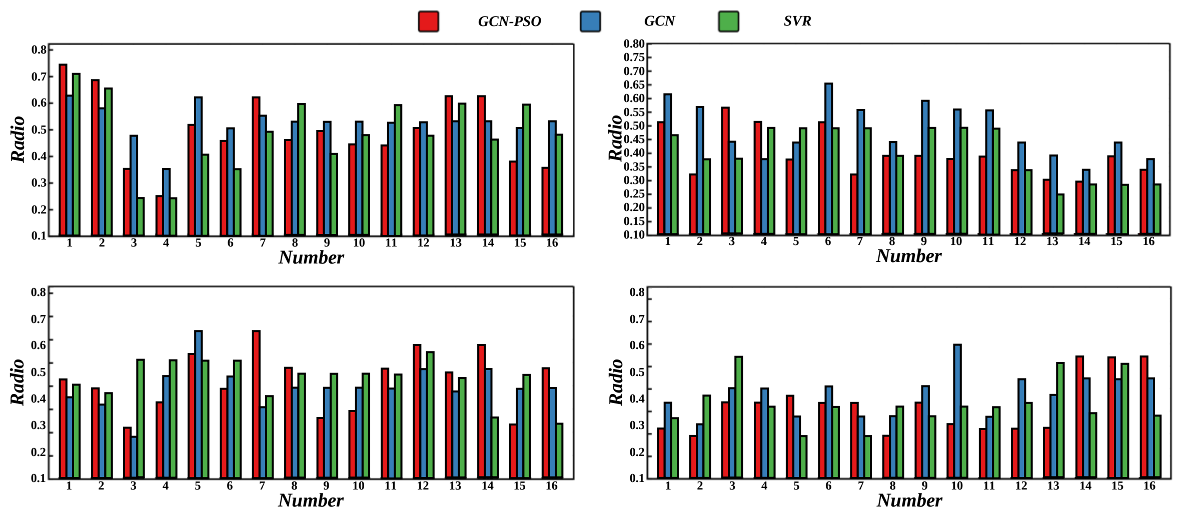

Figure 7: Average absolute error of prediction results of each baseline model

Figure 8 shows that the MAE, MAPE, and RMSE indicators of the GCN-PSO model on the PEMS04 dataset are 5.7%, 8.2%, and 4.3% lower than the LSTM, respectively. On the PEMS08 dataset, the three indicators of GCN-PSO were reduced by 4.93%, 15.26%, and 4.55% compared to DCRNN, respectively. However, GCN performed the worst among the 6 baseline methods because traffic flow prediction is mainly a time series problem, and temporal features are more important than spatial features. Performance of GCN-PSO model on PEMS08 is better than that of PEMS04, mostly because PEMS04 has more traffic nodes than PEMS08.

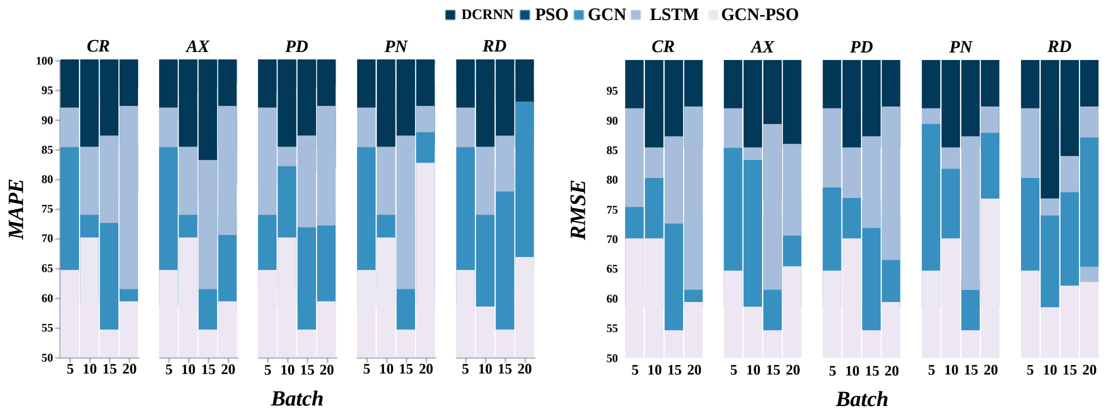

Figure 8: Comparison of prediction results with baseline method

Figure 9 compare the predicted traffic flow with the actual traffic flow based on the GCN-PSO urban traffic flow prediction model. As can be seen from the figure, there are some differences between the two under different lengths, and the flow values fluctuate, which reflects the performance of the model in predicting traffic flow under different conditions. Through these comparisons, the prediction accuracy of the model in different dimensions can be intuitively evaluated, which provides a data reference for further optimization of the urban traffic flow prediction model based on GCN-PSO.

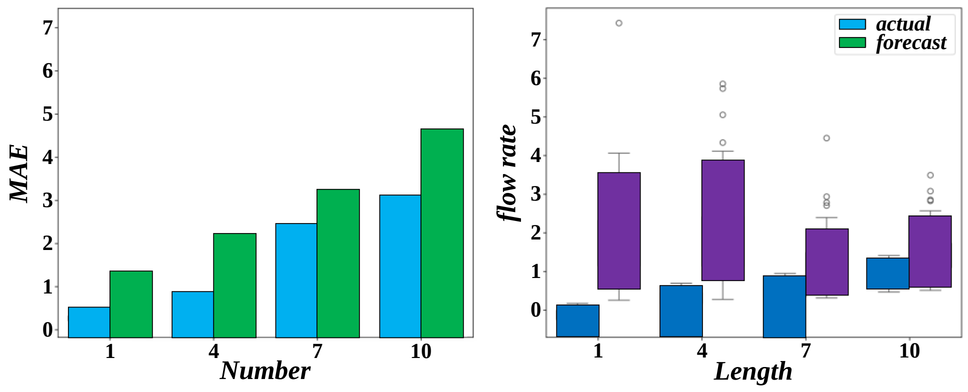

Figure 9: Comparison of forecasted traffic with actual traffic

5 Discussion

The urban traffic flow prediction model based on GCN-PSO proposed in this study shows certain advantages in the traffic flow prediction task, but also has limitations. Compared with traditional deep learning models such as LSTM, GRU, and GCN-based models, GCN-PSO has significantly improved prediction accuracy. The PSO algorithm performs a global search by simulating the foraging behavior of birds, avoids GCN falling into local optimum, finds better parameter combinations in a shorter time, and enhances the capture of complex spatio-temporal features of traffic flow. However, PSO is prone to "premature convergence" in high-dimensional problems, and its parameter tuning relies on a large number of experimental iterations, and the key parameter setting lacks a unified standard, which limits the generality of the model. In addition, although multi-scale time fusion can enrich time information and improve performance, it also increases the complexity of data processing, and unreasonable fusion strategies will introduce noise.

GCN-PSO models also face high training costs and scalability challenges. The matrix operation of GCN itself is complex, and the iterative optimization of PSO further increases the amount of computation and training time, and the training time and memory consumption increase dramatically in large-scale urban transportation networks, restricting real-time applications. When the network scales up or new nodes are added, the model needs to be re-optimized and trained, making it difficult to quickly adapt to new scenarios. In the face of the heterogeneity of transportation networks in different cities, the generalization ability of the model is insufficient, and a large number of adjustments and optimizations are needed. In summary, although the model is innovative, it still needs to be improved in terms of parameter tuning, computational efficiency and scalability, and in the future, we can explore the combination of new optimization algorithms, optimize fusion strategies and try distributed computing to improve the practicability and generality.

6 Conclusion

This study profoundly explores the fusion application of graph convolutional network (GCN) and particle swarm optimization (PSO) algorithm in urban traffic flow prediction. Constructing the GCN-PSO model aims to improve the accuracy and efficiency of traffic flow prediction to cope with complex and changeable traffic conditions.

(1) In the model construction process, GCN is first used to capture spatial dependencies in traffic network, which effectively reveals the interaction between different traffic nodes. Subsequently, the PSO algorithm is introduced to optimize parameters of the GCN model, which overcomes limitations of traditional optimization methods and further improves the model performance.

(2) To verify the validity of the GCN-PSO model, we conducted multiple sets of experiments. Experimental results show that optimized GCN-PSO model has significantly improved prediction accuracy compared with the single GCN model. Specifically, the mean square error (MSE) decreased by 15.3%, indicating that the model has more minor errors in predicting traffic flow and more accurate prediction results. At the same time, the average absolute error (MAE) is reduced by 12.7%, further confirming the model's superiority in prediction accuracy.

(3) In the real-time prediction scenario, the response time of the GCN-PSO model is reduced by 8.9%. This result means the model can make predictions faster when processing real-time traffic data and provide more timely information support for traffic management and travel guidance. The shortening of response time not only improves prediction efficiency but also lays a solid foundation for applying real-time traffic control systems.

In terms of model processing capabilities, it is difficult for existing models to accurately capture complex dynamic changes in the face of sudden traffic peaks caused by extreme weather and large-scale events, as well as significant differences in traffic patterns between holidays and weekdays. At the actual deployment level, due to the extremely high real-time requirements of urban traffic data, the delay of model calculation and response may lead to the lag of prediction results, which is difficult to meet the requirements of scenarios such as real-time regulation and control of traffic signals. At the same time, problems such as uneven performance of data acquisition equipment and insufficient stability of data transmission also bring challenges to the stable operation of the model. In future research, it is urgent to explore the effective representation methods of temporal heterogeneity, explore the change laws of traffic flow in different time periods, and quickly adapt the model trained in a certain city or scene to other regions with different topology and traffic rhythm through transfer learning technology, so as to enhance the adaptability and generalization ability of the model in multiple scenarios.

References

[1] H. Zhang, L. Chen, J. Cao, X. Zhang, and S. Kan, “A combined traffic flow forecasting model based on graph convolutional network and attention mechanism,” International Journal of Modern Physics C, vol. 32, no. 12, 2021. https://doi.org/10.1142/S0129183121501588

[2] A. Zhan, F. Du, Z. Chen, G. Yin, M. Wang, and Y. Zhang, “A traffic flow forecasting method based on the GA-SVR,” Journal of High-Speed Networks, vol. 28, no. 2, pp. 97-106, 2022. https://doi.org/10.3233/JHS-22068

[3] H. Zeng, C. Jiang, Y. Lan, X. Huang, J. Wang, and X. Yuan, “Long Short-Term Fusion Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting,” Electronics, vol. 12, no. 1, 2023. 10.3390/electronics12010238

[4] K. Zaraket, H. Harb, I. Bennis, A. Jaber, and A. Abouaissa, “Hyper-Flophet: A neural Prophet-based model for traffic flow forecasting in transportation systems,” Simulation Modelling Practice and Theory, vol. 134, 2024. https://doi.org/10.1016/j.simpat.2024.102954

[5] G. S. Li, L. Yang, S. C. Bai, X. Y. Song, Y. J. Ren, and S. Q. Liu, “BIKAGCN: Knowledge-Aware Recommendations Under Bi-layer Graph Convolutional Networks,” Neural Processing Letters, vol. 56, no. 1, 2024. https://doi.org/10.1007/s11063-024-11475-6

[6] R. Kumar, J. M. Moreira, and J. Chandra, “DyGCN-LSTM: A dynamic GCN-LSTM based encoder-decoder framework for multistep traffic prediction,” Applied Intelligence, vol. 53, no. 21, pp. 25388-25411, 2023. 10.1007/s10489-023-04871-3

[7] W. Q. Huang, F. Hao, J. X. Shang, W. Y. Yu, S. K. Zeng, C. Bisogni, and V. Loia, “Dual-LightGCN: Dual light graph convolutional network for discriminative recommendation,” Computer Communications, vol. 204, pp. 89-100, 2023.

[8] A. Godó, S. Q. Wu, F. Okura, Y. Makihara, M. Ikeda, S. Sato, M. Suzuki, Y. Satake, D. Taomoto, and Y. Yagi, “PPGCN: Phase-Aligned Periodic Graph Convolutional Network for Dual-Task-Based Cognitive Impairment Detection,” Ieee Access, vol. 12, pp. 37679-37691, 2024. 10.1109/ACCESS.2024.3371517.

[9] F. Feng, C. Wei, B. Zhao, Y. Lv, and Y. He, “Research on Lane-Changing Decision Making and Planning of Autonomous Vehicles Based on GCN and Multi-Segment Polynomial Curve Optimization,” Sensors, vol. 24, no. 5, 2024. https://doi.org/10.3390/s24051439

[10] J. R. Challapalli, and N. Devarakonda, “Effectual pre-processing with quantization error elimination in pose detector with the aid of image-guided progressive graph convolution network (IGP-GCN) for multi-person pose estimation,” Machine Learning-Science and Technology, vol. 4, no. 2, 2023. 10.1088/2632-2153/acc9fc

[11] S. N. Anjiri, D. R. Ding, and Y. Song, “HyGate-GCN: Hybrid-Gate-Based Graph Convolutional Networks with dynamical ratings estimation for personalized POI recommendation,” Expert Systems with Applications, vol. 258, 2024. https://doi.org/10.1016/j.eswa.2024.125217

[12] K. Subramanian, and P. Kandhasamy, “Mining high utility itemsets using Genetic Algorithm Based-Particle Swarm Optimization (GA-PSO),” Journal of Intelligent & Fuzzy Systems, vol. 44, no. 1, pp. 1169-1189, 2023. https://doi.org/10.3233/JIFS-220871

[13] M. Saad, R. N. Enam, and R. Qureshi, “Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data application,” Frontiers in Big Data, vol. 7, 2024. https://doi.org/10.3389/fdata.2024.1358486

[14] P. K. K. Ranganna, S. G. Matt, C. L. Chen, A. B. Jayachandra, and Y. Y. Deng, “Fitness Sharing Chaotic Particle Swarm Optimization (FSCPSO): A Metaheuristic Approach for Allocating Dynamic Virtual Machine (VM) in Fog Computing Architecture,” Cmc-Computers Materials & Continua, vol. 80, no. 2, 2024. 10.32604/cmc.2024.051634

[15] Mas Omar, Fitri Yakub, Shahrum Shah Abdullah, Muhamad Sharifuddin Abd Rahim, Ainaa Hanis Zuhairi, and Niranjana Govindan, "One-step vs horizon-step training strategies for multi-step traffic flow forecasting with direct particle swarm optimization grid search support vector regression and long short-term memory," Expert Systems with Applications, vol. 252, pp. 124154, 2024. https://doi.org/10.1016/j.eswa.2024.124154

[16] T. Zhou, B. Huang, R. Li, X. Liu, and Z. Huang, “An attention-based deep learning model for citywide traffic flow forecasting,” International Journal of Digital Earth, vol. 15, no. 1, pp. 323-344, 2022. https://www.tandfonline.com/doi/full/10.1080/17538947.2022.2028912

[17] Y. Zhao, Y. Lin, H. Wen, T. Wei, X. Jin, and H. Wan, “Spatial-Temporal Position-Aware Graph Convolution Networks for Traffic Flow Forecasting,” Ieee Transactions on Intelligent Transportation Systems, vol. 24, no. 8, pp. 8650-8666, 2023. 10.1109/TITS.2022.3220089

[18] W. Zhang, K. Zhu, S. Zhang, Q. Chen, and J. Xu, “Dynamic graph convolutional networks based on spatiotemporal data embedding for traffic flow forecasting,” Knowledge-Based Systems, vol. 250, 2022. https://doi.org/10.1016/j.knosys.2022.109028

[19] S. Zhang, Y. Guo, P. Zhao, C. Zheng, and X. Chen, “A Graph-Based Temporal Attention Framework for Multi-Sensor Traffic Flow Forecasting,” Ieee Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 7743-7758, 2022. 10.1109/TITS.2021.3072118

[20] Quanzhi Liu, Shuang Wu, and Peng Zhang, "Statistical Analysis of Urban Traffic Flow Using Deep Learning," Informatica, vol. 48, no. 5, 2024. ttps://doi.org/10.31449/inf.v48i5.5393

[21] Q. Zhang, W. Chang, C. Li, C. Yin, Y. Su, and P. Xiao, “Attention-based spatial-temporal graph transformer for traffic flow forecasting,” Neural Computing & Applications, vol. 35, no. 29, pp. 21827-21839, 2023. https://doi.org/10.1007/s00521-023-08951-w

[22] H. Zhang, H. Wang, L. Chen, T. Zhao, and S. Kan, “Traffic Flow Forecasting Based on Transformer with Diffusion Graph Attention Network,” International Journal of Automotive Technology, vol. 25, no. 3, pp. 455-468, 2024. https://doi.org/10.1007/s12239-024-00036-4

[23] H. Zhang, S. Kan, J. Cao, L. Chen, and T. Zhao, “A traffic flow-forecasting model based on multi-head spatio-temporal attention and adaptive graph convolutional networks,” International Journal of Modern Physics C, vol. 33, no. 10, 2022. https://doi.org/10.1142/S0129183122501376

[24] H. Zhang, L. Chen, J. Cao, X. Zhang, S. Kan, and T. Zhao, “Traffic Flow Forecasting of Graph Convolutional Network Based on Spatio-Temporal Attention Mechanism,” International Journal of Automotive Technology, vol. 24, no. 4, pp. 1013-1023, 2023. https://doi.org/10.1007/s12239-023-0083-9

[25] L. Yu, Z. Wang, W. Yang, Z. Qu, and C. Ren, “IEDSFAN: information enhancement and dynamic-static fusion attention network for traffic flow forecasting,” Complex & Intelligent Systems, vol. 11, no. 1, 2025. https://doi.org/10.1007/s40747-024-01663-1

[26] Z. Ye, H. Wang, K. Przystupa, J. Majewski, N. Hots, and J. Su, “Dynamic Spatio-Temporal Hypergraph Convolutional Network for Traffic Flow Forecasting,” Electronics, vol. 13, no. 22, 2024. https://doi.org/10.3390/electronics13224435

[27] T. Yan, H. Gong, Y. Zhan, and Y. Xia, “CAG-NSPDE: Continuous adaptive graph neural stochastic partial differential equations for traffic flow forecasting,” Neurocomputing, vol. 603, 2024. https://doi.org/10.1016/j.neucom.2024.12825

[28] Z. Xu, Z. Lv, B. Chu, and J. Li, “A Fast Spatial-temporal Information Compression algorithm for online real-time forecasting of traffic flow with complex nonlinear patterns,” Chaos Solitons & Fractals, vol. 182, 2024. https://doi.org/10.1016/j.chaos.2024.114852

[29] Y. Xu, L. Han, T. Zhu, L. Sun, B. Du, and W. Lv, “Generic Dynamic Graph Convolutional Network for traffic flow forecasting,” Information Fusion,

vol. 100, 2023. https://doi.org/10.1016/j.inffus.2023.101946

[30] Fatemeh Rezaei Aderyani and S. Jamshid Mousavi, "Machine learning-based rainfall forecasting in real-time optimal operation of urban drainage systems," Journal of Hydrology, vol. 645, pp. 132118, 2024. https://doi.org/10.1016/j.jhydrol.2024.132118