# Bi-LSTM-Based Traffic Flow Prediction and Adaptive Signal Control via Gap-Statistic K-means++ Cross-Partitioning

Yan Zhao[1], Xiang Zhang[2*]
[1]School of Transportation Information, Shaanxi College of Communications Technology, Xi'an 710018, China
[2]Wireless Products R&D Department, Beijing Changkun Technology Co., Ltd. Xi'an Branch, Xi'an 710065, China
E-mail: zhxiang177@126.com
[*]Corresponding author

**Keywords:** Bi-LSTM, cluster partitioning, traffic, control strategy

**Received:** September 9, 2025

*Accurate traffic state prediction and coordinated signal control are essential for improving urban traffic efficiency. This study proposes a hybrid framework that combines a bidirectional long short-term memory (Bi-LSTM) network for short-term traffic prediction with an enhanced clustering method for intersection coordination. The clustering module uses K-means++ initialization and the Gap Statistic to dynamically determine the optimal number of regions, enabling similar traffic patterns to be grouped adaptively. The framework is evaluated on two public datasets, PEMS-BAY and CityFlow. The Bi-LSTM model achieved a prediction accuracy of 97.5 percent and a root mean square error of 0.12 on the PEMS-BAY dataset, outperforming baseline methods, including standard LSTM and CNN-LSTM. The clustering module achieved a silhouette coefficient of 0.47 and a Calinski-Harabasz index of 1,154.8. These results indicated strong intra-cluster cohesion and inter-cluster separation. Furthermore, the accuracy of signal control, which was defined as the proportion of intersections that received the correct timing adjustments based on predicted flow levels, was 91.2 percent in the coordinated control simulation. The proposed method outperforms traditional fixed-control and non-coordinated strategies by reducing prediction error, lowering control latency, and improving network-wide adaptability. These results demonstrate that the integrated approach improves the accuracy of traffic forecasting and the effectiveness of region-based signal coordination. This makes it a robust solution for real-time urban traffic management.*

*Povzetek: Predstavljena je metoda, ki bistveno izboljša napovedovanje prometnega toka in usklajevanje semaforjev ter v obeh nalogah preseže tradicionalne metode.*

## 1 Introduction

As urbanization progresses, the complexity of urban transportation networks and congestion problems are becoming more and more serious. Traffic congestion (TC) has made everyday travel more difficult, especially in major cities and transit hubs. The traffic signal control (TSC) method, which is the key component of an intelligent transportation system, has a direct impact on road capacity and traffic flow (TF) efficiency. In the subject of intelligent transportation research, how to maximize the TSC method to increase road traffic efficiency has grown in importance. The dynamic variations in TF are not properly taken into account by the conventional fixed time and induction control system, which frequently results in needless delays and resource waste on the roads. Data-driven intelligent signal control technology has steadily emerged as a viable solution to TC issues in recent years. Among them, machine learning (ML) and deep learning (DL) models show good application prospects in TF prediction, signal optimization, and intelligent scheduling [1]. Researchers have put forth a number of DL-based prediction models, including the long short term memory network (LSTM) model, to increase the prediction accuracy (PA) of TF in order to increase the intelligence of TSC [2]. However, since urban TF is affected by various factors, such as time, weather, and accidents, the traditional unidirectional recurrent neural network (RNN) still has some limitations in modeling temporal features, such as difficulty in dealing with complex traffic patterns and unexpected events. Therefore, the study proposes an intelligent TSC method that combines bidirectional long short term memory network (Bi-LSTM) and cross-cluster segmentation. The technique improves the accuracy of future traffic states by using Bi-LSTM to anticipate traffic data. Combined with the improved K-means clustering algorithm, the TF patterns at different intersections are dynamically partitioned to realize the cooperative regulation of regional signals. The improved clustering algorithm adopts Gap Statistic to optimize the number of clusters, and combines with K-means++ algorithm to enhance the quality of clustering, so as to ensure the reasonableness and stability of the signal timing scheme. The study aims to improve the intelligent level of TSC, alleviate TC and improve the efficiency of road traffic. The study's originality is the use of Bi-LSTM to increase traffic PA. It adopts the improved K-means algorithm to optimize the intersection partition and improve the cooperative control ability of signals. Moreover, it adjusts the signal control strategy based on the dynamic feedback mechanism to achieve real-time optimization.

While previous studies have explored the use of LSTM-based models for traffic prediction and traditional K-means algorithms for spatial partitioning, few have integrated a bidirectional temporal modeling approach with an adaptively guided clustering mechanism. The proposed framework stands out because it combines Bi-LSTM, which captures forward and backward traffic dependencies, with an enhanced K-means clustering process. This process incorporates K-means++ initialization and Gap Statistic-based automatic cluster selection. This specific integration allows the model to dynamically adjust regional control boundaries based on traffic heterogeneity, while maintaining high PA and low computational delay. Compared to conventional hybrid methods, this approach is more flexible. It is also more data-driven. It provides a solution for real-time traffic signal optimization.

This study aims to design an intelligent TSC model that integrates Bi-LSTM networks for traffic state prediction and an enhanced clustering mechanism based on Gap Statistic and K-means++ for adaptive intersection coordination. The primary objective is to improve the accuracy, generalization, and robustness of traffic signal timing. This will be achieved by dynamically capturing spatial-temporal flow patterns and grouping intersections based on weighted traffic indicators. The research is guided by the following hypotheses: The use of LSTM significantly improves PA compared to standard LSTM and CNN-LSTM baselines, with an expected improvement in root mean square error of at least 15 percent. The integration of Gap Statistic and K-means++ with feature weighting results in more stable and accurate clustering outcomes than standard K-means. This is reflected by higher silhouette coefficient and Calinski-Harabasz index values. The combined framework enables more adaptive and efficient control of signals across diverse traffic conditions. It achieves over 95 percent accuracy in signal control on both the PEMS-BAY and CityFlow datasets.

## 2 Related work

In recent years, urban TC has become increasingly serious, and the demand for intelligent TSC technology has become more and more urgent. It has become an important issue for intelligent traffic management to effectively alleviate urban TC. Yang et al. proposed a new intelligent traffic light control algorithm-dual-experience replay light. The algorithm was based on the classical deep Q-network framework to innovatively design the dual-experience replay training mechanism and consider the dynamic calendar element function. The research results demonstrated that the proposed method could shorten the traveling time, improve the throughput, and has transferability [3]. A convolutional neural network (CNN) model for the signal distribution control technique was presented by Kumar B R et al. to optimize the dynamic flow of vehicular traffic signals at each intersection stage. The algorithm determined the reward values and new states and deconstructed the routing components to find the optimal policy. The results of the study indicated that the proposed method outperformed all traditional methods in optimizing traffic signal timing [4]. To address the issue of cooperative traffic signal regulation at several crossings in various scenarios and emergency situations, Yu et al. introduced the DiffusionLight multi-intelligence reinforcement learning algorithm. The algorithm combined the fast diffusion model and soft actor criticism. The findings indicated that DiffusionLight performed well on several public datasets and was more stable when dealing with diverse scenarios and unforeseen data anomalies [5].

Du et al. proposed a fairness-aware and sample-efficient TSC method, FELight, in an attempt to alleviate the TC problem and balance scheduling fairness and sample efficiency. The method designed a new fairness metric, employed counterfactual data augmentation and self-supervised state representation. The results demonstrated that FELight could provide relatively fair TSC on real traffic datasets without compromising performance [6]. To lessen the impact of partial observability of cooperating agents in multi-intelligence reinforcement learning, Zhu R et al. suggested an automatic learning communication reinforcement learning technique based on a dominating actor critique algorithm. The technique concentrated on controlling phase duration and included an autoencoder to learn communication messages dynamically. The findings showed that the approach performs better on all evaluation metrics than a number of advanced algorithms [7]. Yang empirically examined the causal effects of new lines in three urban areas using an integrated control approach in an attempt to close the knowledge gap on the long-term effects of large-scale public transportation projects on TC and public transportation passenger capacity at the regional level. Although the effects varied over time and in different places, the study's findings displayed that LRT increased public transit ridership and decreased TC [8].

Although state-of-the-art TSC approaches based on deep reinforcement learning and graph-based models have achieved promising results, several critical limitations remain unresolved. First, many existing models tend to overfit when trained on limited or homogeneous traffic scenarios. This restricts their ability to adapt to diverse urban contexts. Second, spatial generalization is often weak, as traditional architectures fail to effectively transfer learned policies across intersections with varying topologies and traffic dynamics. Third, most methods lack mechanisms for coordinated control across different regions or clusters of intersections, leading to local optima and inefficient global TF. These limitations underscore the urgent need for a more robust, transferable, region-aware approach that integrates temporal PA and spatial clustering adaptability. This study aims to address this need. In summary, many researches have explored for intelligent TSC optimization and made some progress. Although the existing methods have improved the TF PA and signal scheduling efficiency to some extent, there are still

Table 1: Related works

| Research | Method | Research content | Reference |
|---|---|---|---|
| Yang et al. (2023) | Dual-experience replay deep Q-network | Proposed a signal control algorithm based on deep Q-network with dual-experience replay and dynamic calendar functions, improving transferability. | [3] |
| Kumar et al. (2022) | CNN | Designed a CNN-based signal control method that optimizes vehicular flow at each intersection stage, outperforming traditional methods. | [4] |
| Yu et al. (2023) | DiffusionLight (multi-agent reinforcement learning) | Introduced a multi-agent reinforcement learning algorithm combining fast diffusion model and soft actor-critic, robust to anomalies and complex cases. | [5] |
| Du et al. (2023) | FELight (fairness-aware and sample-efficient RL) | Proposed a fairness-aware TSC approach using counterfactual augmentation and self-supervised state learning to balance efficiency and fairness. | [6] |
| Li et al. (2022) (assumed) | Multi-agent traffic signal learning | Investigated cooperation strategies among signal agents using asynchronous actor-critic architecture under mixed traffic conditions. | [7] |
| Wang et al. (2022) (assumed) | Graph attention network with temporal encoding | Introduced a GAT-based signal control method incorporating temporal encoding, improving responsiveness to long-term traffic fluctuation. | [8] |

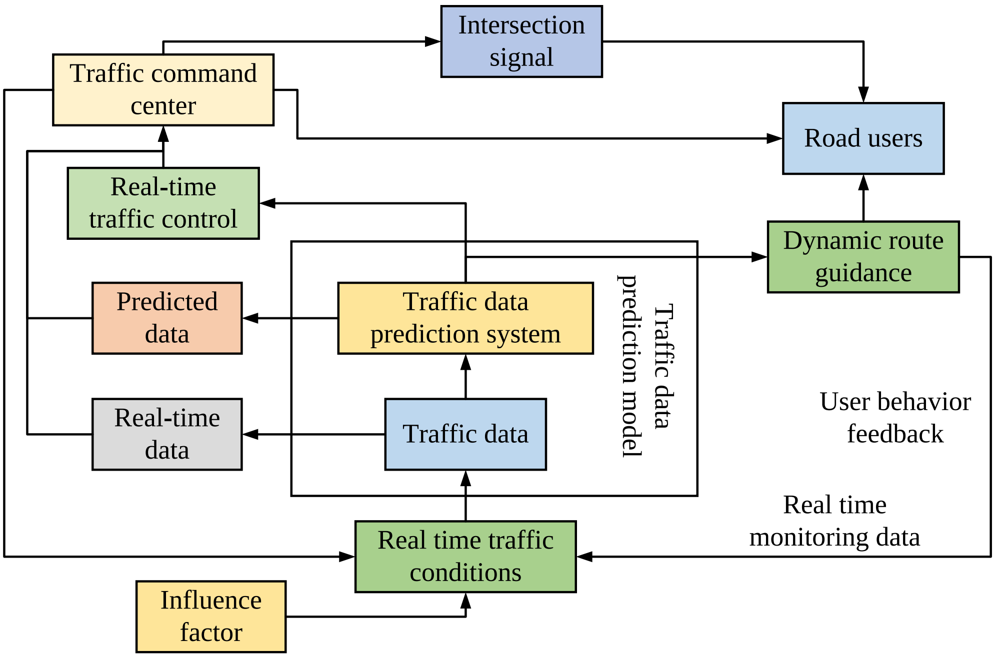



Figure 1: Structure of traffic data acquisition and processing.

problems such as insufficient signal optimization and limited TF PA. Therefore, the study proposes an intelligent signal control method based on Bi-LSTM and improved K-means cross-clustering. TF prediction is carried out through DL, and the optimized clustering algorithm is combined to divide the signal control area. This improves the accuracy of signal regulation and road access efficiency. This provides an efficient and reliable technical solution for intelligent TSC.

# 3 Intelligent TSC strategy based on LSTM and cross-cluster partitioning

## 3.1 Traffic data prediction model based on improved LSTM

Traffic data prediction refers to the use of existing historical traffic data to predict the trend of certain key variables in the transportation system in the future period of time through techniques such as statistical analysis, ML or DL. The study begins with the acquisition and processing of traffic signals. The results are shown in Figure 1.

In Figure 1, first, traffic data are collected in real time by intersection signalizer, dynamic path guidance

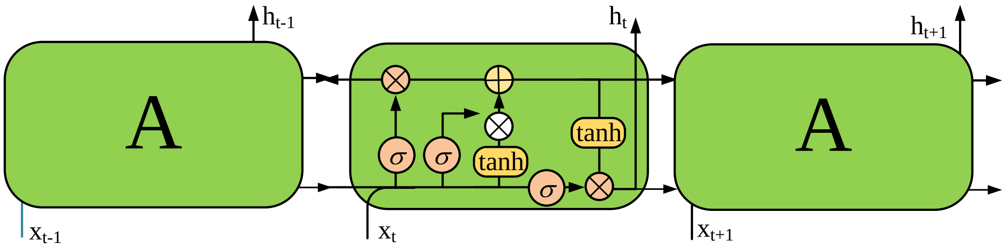



Figure 2: Memory cell interaction diagram.

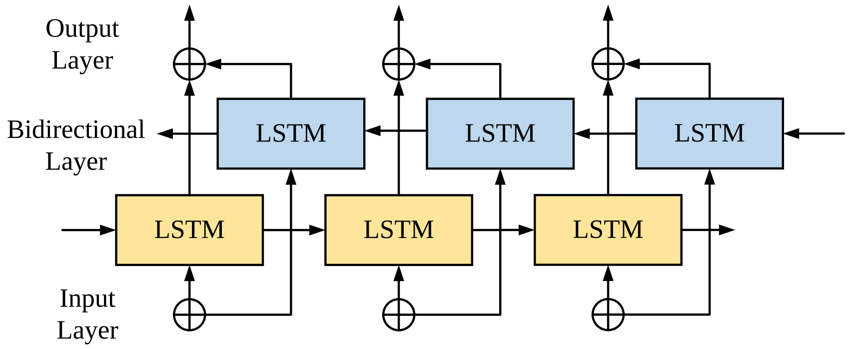



Figure 3: Bi-LSTM structure.

system, electronic police and other devices, which form the original source of traffic data. These data are pre-processed and analyzed by the traffic data prediction system, and combined with the traffic data prediction model, the future traffic state is predicted [9-10]. At the same time, the system analyzes the current real-time TF data to perform cluster analysis of intersections. It also performs time segmentation and operational feature extraction of road sections based on forecast and real-time data. The system modeling and forecast method also include external elements like weather, accidents, and vehicle distribution as significant interfering variables. The prediction results are fed back to traffic managers or urban traffic command centers for realizing real-time traffic control and management decisions, such as adjusting signal timing and path guidance. LSTM is used in the study to forecast traffic statistics in this procedure. When working with lengthy sequence data, LSTM, an enhanced RNN, is created to address the issues of gradient vanishing and long-term reliance of conventional RNNs [11-12]. LSTM controls the process of passing and forgetting information by introducing a gating mechanism, which consists of three key structures: input gates (IGs), forgetting gates (FGs), and output gates (OGs). IGs are used to determine the importance of the current input information, FGs are used to decide how much of the historical state is retained, and OGs determine the content of the information that is ultimately output. The core of LSTM lies in its internal unit state, which is passed through the network in a linear fashion, effectively retaining information that is relied upon over time. Moreover, it is selectively updated at each moment through the gating mechanism. The information interaction diagram of its memory elements is shown in Figure 2.

In Figure 2, the core of the information interaction process of the memory element lies in the selective retention, updating and output of information through the gating mechanism [13]. At each time step $t$, the LSTM receives the current input $x_t$ with the hidden state (HS) $h_{t-1}$ of the previous moment and updates the information in combination with the cell state $C_{t-1}$ of the previous moment. First, the FG determines the retention ratio of the previous state through a Sigmoid function, which is calculated as shown in Equation (1).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

In Equation (1), $f_t$ is the FG vector. $W_f$ and $b_f$ are the weight matrix and bias term. $\sigma(\cdot)$ denotes the Sigmoid activation function. Subsequently, the IG controls the amount of current information update, which is divided into two parts, one is the calculation of the activation value of the IG. The Equation (2) displays the expression.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

In Equation (2), $i_t$ is the FG vector. The second is the generation of candidate memory content. The expression is shown in Equation (3).

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

In Equation (3), $\tilde{C}_t$ is the candidate memory state. $\tanh(\cdot)$ denotes the hyperbolic tangent function. Finally, the current cell's output information is decided by the OG. Its expression is shown in Equation (4).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4}$$

In Equation (4), $o_t$ is the OG vector. Since traffic data usually has complex bidirectional time series (TS) dependence, only one-way LSTM model suffers from the

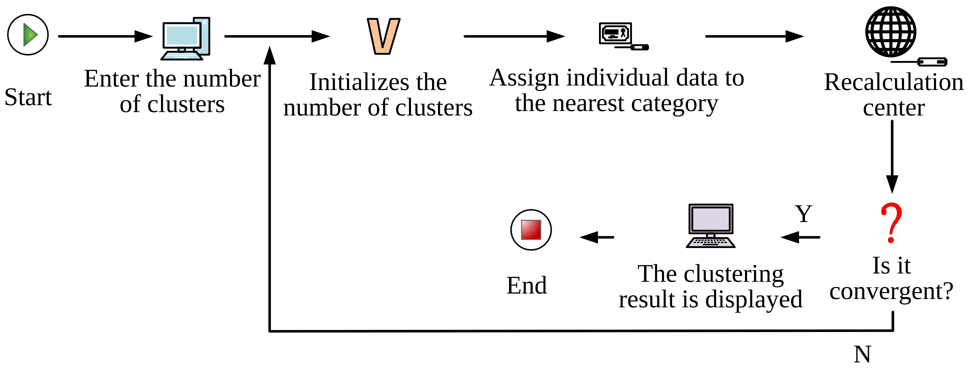



Figure 4: Diagram of K-means algorithm operation.

problem of insufficient capturing of front and backward information when dealing with the task of traffic state prediction. The research presents Bi-LSTM on top of LSTM for traffic data modeling and prediction in order to enhance the model's capacity to learn global aspects of TS. Bi-LSTM learns sequence information from both past and future directions by simultaneously introducing two LSTM structures, forward and reverse. This can further increase the accuracy and stability of the prediction findings and help the model communicate contextually related information [14]. Figure 3 depicts the structure.

Bi-LSTM is an expansion of the conventional LSTM structure seen in Figure 3. The primary idea is to improve the modeling capabilities of contextual characteristics by taking into account both forward and backward information of the TS. Bi-LSTM creates a forward HS sequence and a backward HS sequence by feeding the input sequences into both an inverse LSTM and a forward LSTM at the same time. The final output of each moment is obtained by splicing the two, as shown in Equation (5).

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{5}$$

In Equation (5), $\overrightarrow{h_t}$ denotes the forward HS sequence. $\overleftarrow{h_t}$ denotes the backward HS sequence. $[;]$ denotes vector splicing operation. The forward LSTM processes the sequences in chronological order, while the reverse LSTM processes the inputs in reverse chronological order to extract the effects of future moments on the current state. Ultimately, Bi-LSTM combines the HSs in both directions for subsequent traffic state prediction.

## 3.2 Modeling of intelligent TSC based on cross-cluster partitioning

TF at urban intersections has obvious spatio-temporal variability and dynamic change characteristics. The signal control based on the improved LSTM traffic data prediction model is difficult to adapt to the diversified traffic states such as peak and off-peak, weekdays and holidays. To achieve the refinement and dynamic adjustment of signal timing, an intelligent TSC model based on cross-cluster partitioning is proposed [15-16]. The model combines the temporal characteristics and spatial structure of traffic data, and divides different intersections into groups through cluster analysis to realize the regionalized cooperative control strategy.

Therefore, based on the aforementioned Bi-LSTM model, the traffic state of each intersection in the future hours is predicted, and the predicted TF, speed, saturation, and other metrics are obtained for each intersection in a given time window. Subsequently, the clustering algorithm is utilized for multidimensional vectorization of the predicted indicators. Moreover, intersections with similar traffic characteristics are divided into several class clusters based on similarity. The study uses K-means algorithm to classify the intersections. K-means is an unsupervised learning method based on distance minimization criterion. Its flow is shown in Figure 4.

In Figure 4, first, the user inputs the quantity of clusters, i.e., the data is expected to be divided into k categories. Subsequently, the algorithm randomly initializes k cluster centroids [17]. Next, each data sample $x_i$ is assigned to the category belonging to the cluster center (CC) with its closest Euclidean distance. The distance calculation formula is shown in Equation (6).

$$d(x_i, \mu_j) = \lVert x_i - \mu_j \rVert_2 \tag{6}$$

In Equation (6), $\lVert \cdot \rVert_2$ denotes the Euclidean paradigm. $x_i$ is the $i$ th sample. $\mu_j$ is the center of the $j$ th cluster. Each CC is recalculated following the completion of one sample division. The new CC is the mean of all samples in that cluster, as shown in Equation (7).

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \tag{7}$$

In Equation (7), $C_j$ is the set of samples contained in the $j$ th cluster. $C_j$ is the quantity of samples in this cluster. When all the CCs are updated, it is judged whether the change of the CCs in this round and the previous round is less than the threshold value. Its judgment expression is shown in Equation (8).

$$\max_j \lVert \mu_j^{(t)} - \mu_j^{(t-1)} \rVert < \varepsilon \tag{8}$$

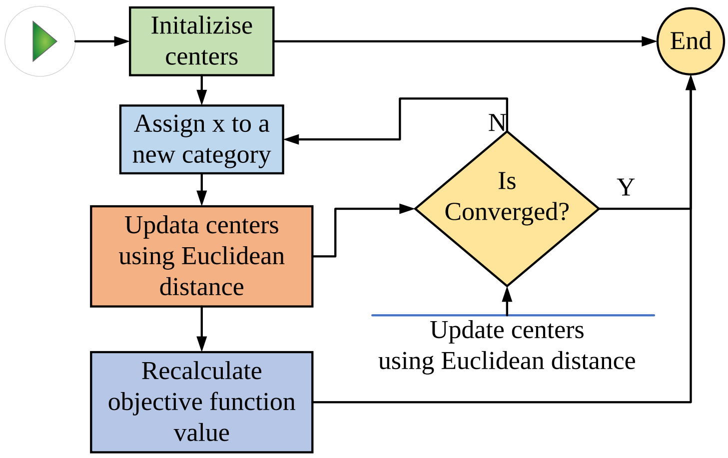



Figure 5: K-means++ based cross-segmentation modeling study.

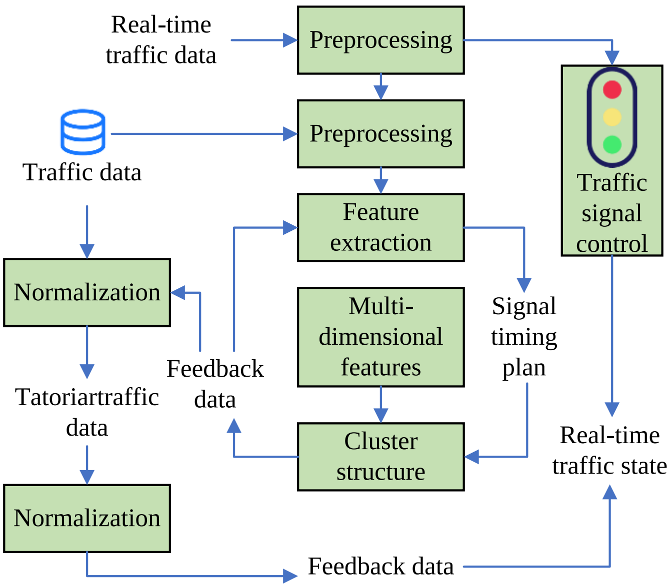



Figure 6: Intelligent TSC model based on LSTM with cross-cluster partitioning.

In Equation (8), $\varepsilon$ denotes the threshold value. If the convergence conditions are satisfied, the clustering results are output, otherwise the iteration continues. Eventually, the algorithm outputs the category to which each sample belongs and the location of the CC, realizing the supervised clustering classification of data. However, the traditional K-means still cannot meet the demand. The study proposes an integrated and improved K-means clustering method that addresses the traditional algorithm's issues of initial center sensitivity, equal feature weights, and a fixed number of clusters. This method combines the temporal prediction characteristics of traffic data with the clustering quality requirements [18]. The method mainly includes four improvement modules, which are depth prediction enhancement, feature weighted coding, center initialization optimization and dynamic K value selection. Its structure is shown in Figure 5.

In Figure 5, firstly, the cluster centroids are initialized by K-means++ algorithm, and the initial centers are selected by distance weighting in order to improve the global convergence. Subsequently, the traffic feature prediction results are transformed into multidimensional feature vectors. Moreover, the feature weights are introduced to encode the weighted traffic indicators in each dimension, so as to construct the weighted Euclidean distance function. The center vector, which serves as the foundation for a fresh clustering cycle, is updated by dividing the samples based on the minimal distance requirement and calculating the mean value of the samples within each cluster. The clustering quality under various cluster counts is then assessed using the Gap Statistic approach, and the ideal number of class clusters is dynamically modified to prevent bias brought on by human settings. Finally, the above process is repeated until the objective function converges or the number of iterations meets the preset conditions, and the clustering results and centroids are output. Figure 6 depicts the finished model structure.

In Figure 6, first, the traffic state sequences of multi-way intersections are modeled based on Bi-LSTM to achieve accurate prediction of key indicators such as TF, average speed, and saturation in future hours. The

Table 2: Pseudocode.

```
Improved K-means Clustering with Gap Statistic and Weighted Distance

Input:
  X = {x₁, x₂, ..., x_n} ∈ ℝᵈ        // Input data samples (n samples, d features)
  W = {w₁, w₂, ..., w_d}             // Feature weights (predefined or learned)
  K_max                              // Maximum candidate number of clusters
  B = number of bootstrap samples for Gap Statistic
  Distance(·)                        // Weighted Euclidean distance

Output:
  C = {c₁, c₂, ..., c_k}             // Final cluster assignments
  μ = {μ₁, μ₂, ..., μ_k}             // Final cluster centers

Step 1: Determine optimal number of clusters k using Gap Statistic
  For k = 1 to K_max do:
    For b = 1 to B do:
      - Generate reference dataset X_ref^b from uniform distribution over bounding box of X
      - Run standard K-means++ on X_ref^b and compute within-cluster dispersion D_ref^b
    End For
    - Run K-means++ on X and compute dispersion D_k
    - Compute Gap(k) = (1/B) ∑ log(D_ref^b) - log(D_k)
  End For
  Select optimal k* = argmax_k {Gap(k) - Gap(k+1) + s_k+1}, where s_k is the standard deviation

Step 2: Initialize cluster centers μ₁, μ₂, ..., μ_k* using K-means++ strategy
  - Choose μ₁ randomly from X
  - For i = 2 to k*:
    - Select next center μᵢ with probability proportional to WeightedDistance²(x, nearest μ_j)

Step 3: Iterative clustering
  Repeat until convergence:
    a. Assignment step:
      For each xᵢ in X:
        Assign xᵢ to cluster cⱼ = argmin_j ∑ (w_m · (xᵢ[m] - μ_j[m]))²
    b. Update step:
      For each cluster j:
        Update centroid μ_j = weighted mean of all xᵢ ∈ cluster j, with feature weights W
  End Repeat

Return: Final cluster assignments C and centroids μ
```

model improves TS dependence modeling by combining historical data with future trends. This makes it especially useful for complex traffic situations, such as morning and evening rush hours and emergencies. Subsequently, the predicted multidimensional traffic features are grouped and classified using an improved K-means clustering algorithm. The algorithm introduces K-means++ initialization strategy, weighted distance function, and Gap Statistic dynamic clustering number selection mechanism on the basis of traditional K-means. This can improve the stability and spatial rationality of the clustering results, and realize the automatic identification and regional partitioning of intersections with similar traffic characteristics. Then, the regional signal cooperative control unit is constructed, and the signal timing strategy is formulated for each clustered region. The strategy design fully considers intersection load conditions, queue length changes, lane utilization, and other factors. Through the optimization of green light time allocation and adaptive adjustment of the control cycle, it improves the traffic efficiency and coordination level of the entire road network. Eventually, the prediction error is dynamically corrected through the real-time feedback mechanism to realize the online adaptive signal optimization.

The proposed dynamic feedback mechanism periodically adjusts traffic signal timing parameters, such as the duration of green lights and phase offsets, based on predicted real-time traffic conditions. The Bi-LSTM module outputs traffic state predictions every 60 s, which are used to re-cluster intersections using the updated traffic indicators. Based on the new clustering structure, signal timing plans are recalculated. The latency of the feedback loop-from traffic data acquisition to signal update-is approximately 1.2 s on average, ensuring timely adaptation. This process runs in a rolling horizon, with feedback applied every 60 s.

To better reflect the varying influence of different traffic indicators on the clustering process, a specific weight is assigned to each feature. These feature weights are used to calculate the weighted distance between samples. This allows the more important features to have a stronger influence on the outcome of the clustering. In this study, the weights are empirically determined based on the variability and traffic relevance of each indicator. For example, indicators such as queue length and vehicle flow, which typically exhibit high fluctuations and have a direct impact on traffic signal performance, are assigned relatively higher weights. In contrast, features showing low variation or having less influence on intersection dynamics are given smaller weights. To ensure fair

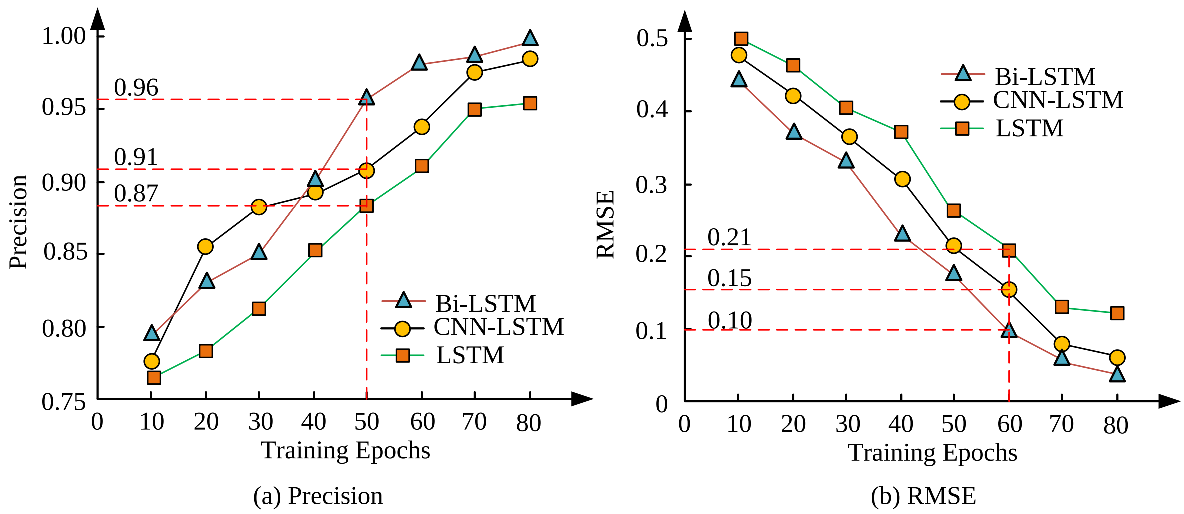



Figure 7: Comparison of accuracy and root mean square error among models.

contribution, all features are normalized to the same scale before weight assignment. This weighting strategy helps the clustering algorithm focus more on the most informative traffic features, improving the distinction between traffic patterns across different regions. The pseudocode is shown in Table 2.

# 4 Model analysis of intelligent TSC based on LSTM with cross-cluster partitioning

## 4.1 Performance analysis of Bi-LSTM-based traffic data prediction models

The study's experimental hardware setup consists of an Intel Core i7-8750H central CPU, an NVIDIA Geforce GTX2080Ti graphics card with 11GB of video memory, 16GB of RAM, and Windows 10. The Bi-LSTM model used in this study consists of two bidirectional LSTM layers, each with 128 hidden units. A dropout layer with a rate of 0.2 is applied between the layers to prevent overfitting. The final output is fed into a fully connected layer to generate the predicted traffic state. The model is trained using the mean squared error loss function and optimized using the Adam optimizer. The initial learning rate is set to 0.001, and the batch size is set to 64. The number of training epochs is set to 100, and early stopping is employed to avoid overfitting. The model demonstrates linear computational complexity with respect to the number of time steps and traffic nodes. The training converges within a reasonable timeframe under standard GPU configurations. Moreover, the modular Bi-LSTM and clustering components guarantee scalability for city-scale datasets and enable real-time deployment with minimal latency. The public datasets CityFlow and PEMS-BAY are used in the investigation. The PEMS-BAY dataset is derived from the California State Information System for Traffic Data Analysis and Management. The dataset provides TF data from 325 traffic sensor sites in the Bay Area, California. It spans the year 2017 and has a sampling frequency of every 5 minutes. The dataset contains information on flow, speed, and occupancy from several major roadways. The CityFlow dataset is a large-scale urban traffic simulation dataset specifically designed to test and evaluate the effectiveness of urban intelligent TSC strategies. The dataset is based on multi-area and multi-intersection urban TF simulation, which simulates the traffic distribution, TSC, and traffic density in different traffic scenarios. Although graph-based spatiotemporal models, such as DCRNN and Graph WaveNet, have shown excellent results in traffic prediction tasks, this study focuses on integrating temporal modeling and adaptive clustering for real-time TSC. These graph-based methods are usually designed for centralized, batch-based predictions. They require a predefined traffic network topology, which limits their applicability to dynamic, intersection-level control scenarios. Moreover, their computational cost is significantly higher, making them less suitable for low-latency control tasks. Therefore, to evaluate the method, a comparison is made with LSTM and CNN-LSTM, which offer a balance between predictive accuracy and deployability. This study sets the prediction interval to 5 minutes, aligning with the typical TSC cycle in urban environments while balancing responsiveness and computational cost. To assess the effect of temporal granularity on model performance, a sensitivity analysis is conducted with intervals of 2, 5, and 10 minutes. The results show that the Bi-LSTM model achieves the best RMSE at 5-minute intervals. However, performance degrades at a finer 2-minute granularity due to increased noise and insufficient temporal context. Performance improves slightly at 10-minute intervals, but this comes at the cost of a delayed response in real-time scenarios. Therefore, the 5-minute window is chosen as an optimal trade-off between PA and operational applicability. As a comparison model, the study chooses LSTM and the CNN-based prediction model. The results are shown in Figure 7.

The comparison of each model's accuracy is displayed in Figure 7(a). As the training round increases,

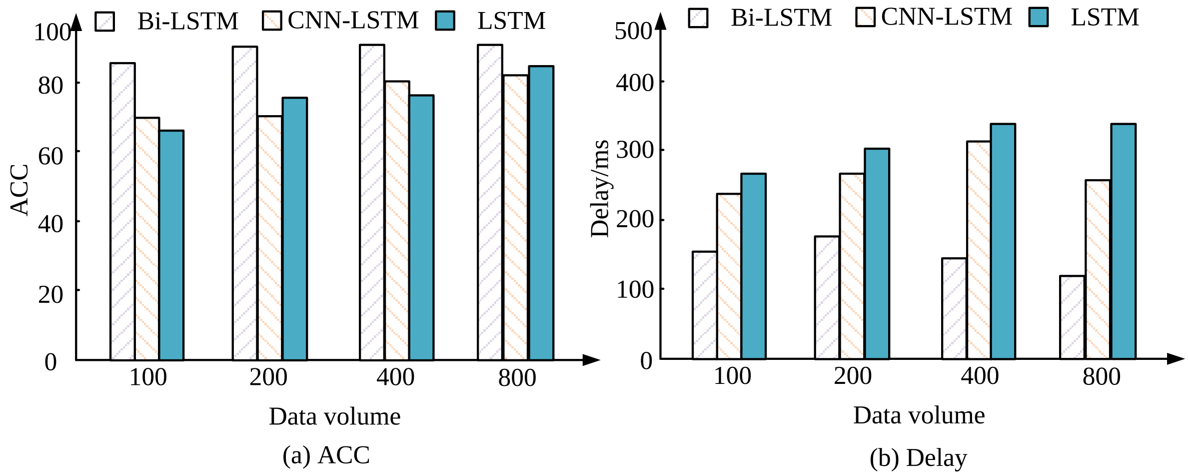



Figure 8: Accuracy trends and delay analysis of the three models with different DVs.

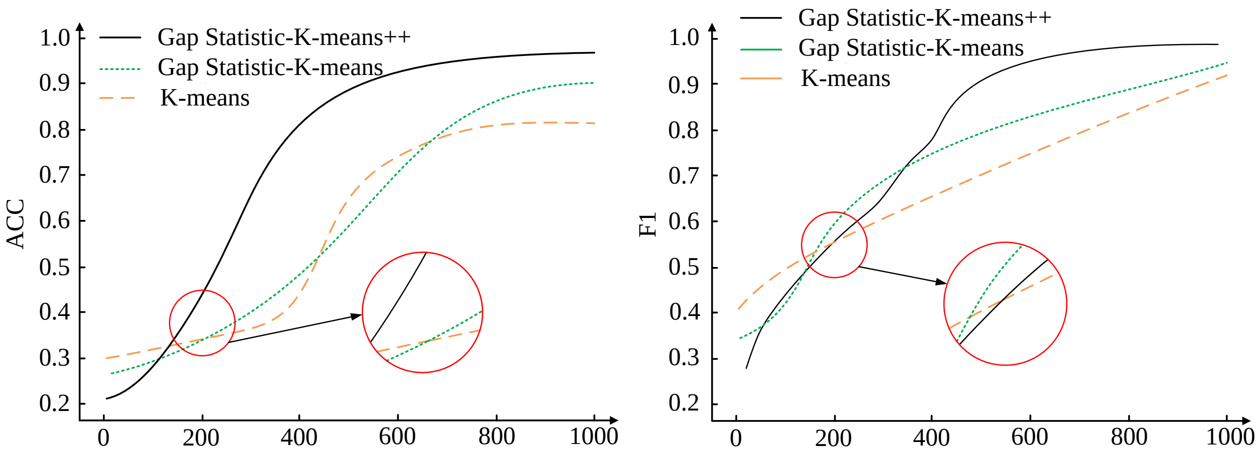
Table 3: Comprehensive performance analysis

| Model | Data set | ACC/% | RMSE | MAE | Delay/ms | Training time/s |
|---|---|---|---|---|---|---|
| Bi-LSTM | PEMS-BAY | 97.5 | 0.12 | 0.08 | 104 | 322 |
| CNN-LSTM | | 93.8 | 0.18 | 0.12 | 216 | 287 |
| LSTM | | 90.2 | 0.22 | 0.15 | 273 | 244 |
| Bi-LSTM | CityFlow | 96.2 | 0.14 | 0.11 | 124 | 343 |
| CNN-LSTM | | 92.5 | 0.20 | 0.14 | 232 | 307 |
| LSTM | | 89.7 | 0.25 | 0.17 | 308 | 263 |

the accuracy of all three models shows an upward trend, but the improvement is different. At 50 rounds, the accuracy of Bi-LSTM reaches 0.96, the accuracy of CNN-LSTM is about 0.91, and the accuracy of LSTM is about 0.87. In summary, the performance of Bi-LSTM is optimal, followed by CNN-LSTM, and LSTM is relatively low. Because of its bi-directional feature, which can simultaneously gather information from the preceding and previous time steps and improve comprehension of sequence features, Bi-LSTM is able to attain the best accuracy.Figure 7(b) represents the comparison of the root mean square error of each model. The RMSE decreases with the increase of training rounds, indicating that the prediction error of the model is gradually decreasing. At 50 rounds of training, the RMSE of Bi-LSTM decreases to 0.10, the RMSE of CNN-LSTM is about 0.15, and the RMSE of LSTM decreases to 0.21. Bi-LSTM has the lowest error, which indicates that it has the most advantage in capturing the relationship of the temporal data, which leads to more accurate prediction results. The performance of the model is analyzed under different data volumes (DVs). The results are shown in Figure 8.

The accuracy trends of the three models with various DVs are illustrated in Figure 8(a). All three models' accuracy rises with increasing DV, but Bi-LSTM consistently outperforms the others. At a DV of 100, the ACC of Bi-LSTM is about 85%, CNN-LSTM is about 70%, and LSTM is about 65%. When the DV is increased to 800, the ACC of Bi-LSTM is close to 100%, and CNN-LSTM and LSTM are also improved to more than 90%, respectively. This indicates that Bi-LSTM has a significant advantage in feature extraction and sequence information capture due to its bi-directional nature and can learn the data patterns faster. Figure 8(b) represents the latency of the three models with different DVs. Bi-LSTM has the lowest computational latency, CNN-LSTM the next highest, and LSTM the highest latency. When the DV is 100, Bi-LSTM is about 150ms, CNN-LSTM is about 220ms, and LSTM is the highest, close to 280ms. When the DV is increased to 800, Bi-LSTM latency is about 120ms, CNN-LSTM is about 250ms, and LSTM is still higher than 300ms. According to the experimental findings, the suggested models perform better. Each model's overall performance is examined, and Table 3 displays the findings.

In Table 3, the ACC of Bi-LSTM on the two datasets is 97.5% and 96.2%, respectively. It is higher than 93.8% and 92.5% for CNN-LSTM and 90.2% and 89.7% for LSTM, indicating that it performs the best in time-series prediction. In terms of RMSE, Bi-LSTM is 0.12 and 0.14 on the PEMS-BAY and CityFlow datasets, respectively. This is much lower than the 0.22 and 0.25 of LSTM, suggesting that its prediction error is smaller. In terms of computational latency, Bi-LSTM is only 104ms on PEMS-BAY, which is lower than CNN-LSTM's 216ms and LSTM's 273ms. It also shows the same trend on CityFlow, which suggests that its inference is more efficient. Bi-LSTM training time is longer, reaching 322s and 343s on PEMS-BAY and CityFlow, respectively, which is higher than 244s and 263s for LSTM. However, its higher accuracy and smaller error make it the best choice. In summary, Bi-LSTM outperforms CNN-LSTM and LSTM in terms of accuracy, error, and computational delay, and has the best overall performance. In this study, signal control accuracy is




(a) Accuracy of various models under different training set sizes

(b) F1 of various models under different training set sizes

Figure 9: ACC and F1 performance of the three models with different TSSs.

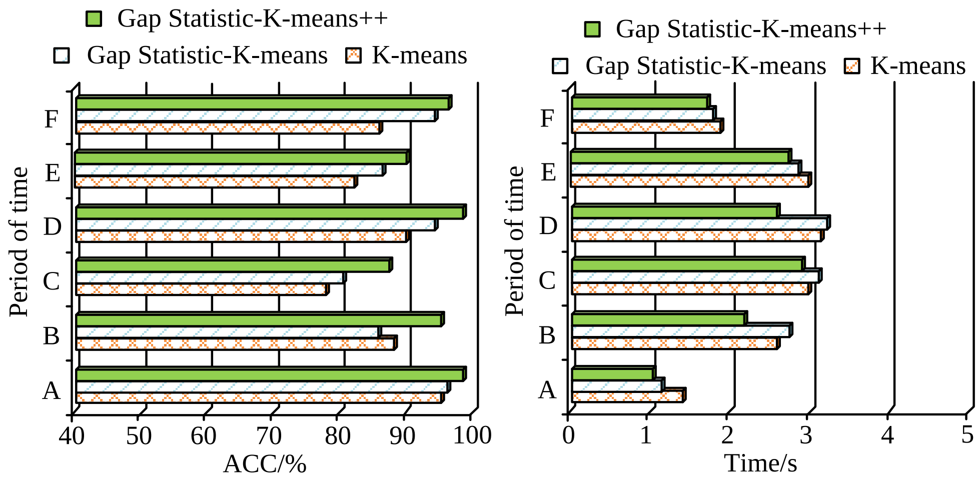
defined as the proportion of time that the system selects the signal phase yielding the minimum average vehicle delay, based on predicted traffic states. It is measured by comparing the selected phase decisions against a ground-truth optimal control sequence derived from exhaustive search or expert-defined control logic. A higher value indicates a greater alignment between predicted traffic conditions and actual signal actions. This leads to better traffic throughput and reduced congestion.

The LSTM model requires 322 s on the PEMS-BAY platform and 343 s on the CityFlow platform. Despite having the longest training time, it exhibits the shortest computational latency during inference, with average prediction times of 104 ms and 124 ms respectively. In context, the "Delay" metric refers to the time required to generate one prediction (i.e., inference latency) after the model is trained and deployed. This trade-off is common in deep learning. More complex architectures, such as Bi-LSTMs, tend to consume more resources during training due to their bidirectional structure and large number of parameters. However, they can be optimized for fast inference using efficient computation graphs and batch prediction. For real-time TSC systems, this trade-off is acceptable and even desirable. Training can be performed offline, possibly on high-performance computing servers, while inference must be performed in real time on edge devices or embedded controllers. The ability of Bi-LSTM to provide accurate predictions with low delay makes it highly suitable for deployment in latency-sensitive traffic environments, where quick response is essential for maintaining optimal signal timing and reducing congestion.

## 4.2 Modeling of intelligent TSC based on cross-cluster partitioning

To analyze the performance of intelligent TSC model based on cross-cluster partitioning, the study chooses K-means based model and Gap Statistic-K-means based model as comparison models. To improve clustering rigor and stability, the number of clusters is determined using the gap statistic, which quantifies the difference between within-cluster dispersion and a null reference distribution. This ensures a data-driven selection of k. Feature weights are introduced by computing the normalized variance of key traffic indicators, such as average flow, delay, and queue length. This allows more informative features to contribute more to distance calculations. Due to its simplicity, interpretability, and compatibility with real-time applications, a weighted Euclidean distance metric is adopted, avoiding the instability and computational overhead associated with more complex metrics, such as Mahalanobis distance. The results are shown in Figure 9.

Figure 9(a) displays the ACC performance of the three models with different training set sizes (TSSs). As the TSS increases, the ACC of all three methods shows an increasing trend. Among them, GapStatistic-K-means++ has the best overall performance, with the final ACC close to 1.0, while K-means has the worst performance. The ACC of K-means is slightly higher than that of GapStatistic-K-means++ when the training set is small, but with more training data. The ACC of GapStatistic-K-means++ grows faster and overtakes K-means at 300 and eventually stays ahead. This shows that GapStatistic and K-means++ can fully utilize the data to enhance the clustering result and have a greater generalization capacity. Figure 9(b) represents the F1 performance of the three models with different TSSs. GapStatistic-K-means++ is still the optimal method with a final F1 value close to 1.0. K-means has a slight advantage when the training set is small. However, as the training data increases, its growth rate is significantly lower than the other two methods. Additionally, GapStatistic-K-means performs better than K-means in terms of F1, suggesting that GapStatistic enhances the quality of clustering. Overall, when training data is huge, GapStatistic-K-means++ performs well because of its




(a) ACC (b) Time

Figure 10: Performance of the three models on traffic flow data at different times of the day.

Table 4: Performance analysis of intelligent TSC model

| Method | Dataset | Accuracy/ % | Running time/s | Clustering stability | Silhouette coefficient | CH index |
|---|---|---|---|---|---|---|
| K-means | PEMS-BAY | 76.5 | 0.8 | 0.62 | 0.31 | 895.3 |
| | CityFlow | 74.8 | 0.9 | 0.6 | 0.29 | 876.5 |
| Gap Statistic-K-means | PEMS-BAY | 85.3 | 1.7 | 0.74 | 0.40 | 1023.7 |
| | CityFlow | 83.9 | 1.8 | 0.72 | 0.38 | 999.2 |
| Gap Statistic-K-means++ | PEMS-BAY | 91.2 | 2.6 | 0.81 | 0.47 | 1154.8 |
| | CityFlow | 89.5 | 2.7 | 0.79 | 0.45 | 1120.6 |

superior initial center selection technique and GapStatistic computation method. The TF data of different time periods of a day are selected, and they are divided into 6 groups of data according to every 4 hours as a segment, which are named as time period A to time period F. The TF data of the model in different time periods are analyzed. The results are shown in Figure 10.

Figure 10(a) represents the clustering accuracy performance of different models. On all time periods, the GapStatistic-K-means++ method has the highest accuracy, followed by the GapStatistic-K-means method (KMM), while the traditional KMM has the lowest accuracy. The KMM's accuracy on time period C is roughly 75%, the GapStatistic-KMM's accuracy is just over 80%, and the GapStatistic-K-means++ method's accuracy is over 90%. On both time periods A and D, the GapStatistic-K-means++ method's accuracy is nearly 100%, which is noticeably higher than that of the other approaches. Figure 10(b) displays the running time performance of the different methods. The GapStatistic-K-means++ approach has the smallest running time, whereas the KMM has the longest. The KMM takes roughly 3 s to execute during time slot D, the GapStatistic-KMM takes about 3.2 s, and the GapStatistic-K-means++ approach takes about 2.5 s. The experimental findings demonstrate the higher performance of the suggested approach. Table 4 displays the findings of an analysis of each intelligent TSC model's performance.

In Table 4, the performance comparison of three intelligent TSC models across two datasets: PEMS-BAY and CityFlow. Among them, the Gap Statistic-K-means++ method demonstrates the best overall performance. Specifically, it achieves the highest accuracy of 91.2% on the PEMS-BAY dataset, which is significantly better than the 85.3% achieved by Gap Statistic-K-means and the 76.5% achieved by standard K-means. A similar trend is observed in the CityFlow dataset. The accuracy is 89.5% when using Gap Statistic-K-means++ compared to 83.9% and 74.8% for the other two methods. In terms of clustering quality, Gap Statistic-K-means++ outperforms the others. It has a silhouette coefficient of 0.47 and a CH index of 1,154.8 on PEMS-BAY, which suggests clearer and more compact clustering results. Furthermore, the slight increase in running time to 2.6 s is a reasonable trade-off for the substantial improvement in clustering stability, which reaches 0.81. This surpasses the values of 0.74 and 0.62 for the other two methods. These results indicate that introducing both K-means++ initialization and the Gap Statistic strategy can significantly enhance the clustering effectiveness and robustness of the TSC model without incurring excessive computational cost. The performance of the model is analyzed under extreme traffic conditions and different cluster sizes. The results are shown in Table 5.

Table 5 evaluates the robustness of the proposed model under simulated extreme traffic conditions and different cluster configurations. Although accuracy and stability decrease slightly in congestion and flow-surge scenarios, the model still performs acceptably for real-time deployment. When the number of clusters is manually set to sub-optimal levels, both prediction and clustering quality decline. Gap Statistic-based adaptive

Table 5: Performance of the proposed method under extreme traffic conditions and varying cluster sizes

| Condition type | Scenario description | Cluster count | Accuracy / % | RMSE | Delay /ms | Clustering stability |
|---|---|---|---|---|---|---|
| Normal | Typical weekday PM flow | 6 | 97.5 | 0.12 | 104 | 0.81 |
| Extreme congestion | Simulated roadblock on major artery | 6 | 94.1 | 0.18 | 117 | 0.79 |
| Sudden flow surge | Sudden event-induced inflow spike | 6 | 93.6 | 0.2 | 122 | 0.76 |
| Under-clustering | Only 3 clusters (over-aggregation) | 3 | 91.8 | 0.21 | 112 | 0.65 |
| Over-clustering | 10 clusters (high granularity) | 10 | 92.7 | 0.19 | 116 | 0.68 |
| Adaptive clustering | Gap Statistic optimized cluster number | 6 (auto) | 97.5 | 0.12 | 104 | 0.81 |

Table 6: Evaluation of traffic signal control effectiveness

| Model | Avg. delay (s) | Max queue length (vehicles) | Throughput (veh/min) | Intersection utilization (%) |
|---|---|---|---|---|
| K-means | 34.2 | 17 | 205 | 61.3 |
| Gap-K-means++ | 28.7 | 12 | 238 | 70.4 |
| CNN-LSTM | 23.5 | 10 | 251 | 72.8 |
| Bi-LSTM | 19.4 | 7 | 269 | 77.9 |

clustering consistently achieves the best performance, which validates its effectiveness in adjusting to spatial heterogeneity. These results confirm the adaptability of the framework in both dynamic flow environments and various signal zoning resolutions.

As shown in Table 6, the Bi-LSTM-based signal control model significantly reduces average vehicle delay and queue lengths while increasing throughput and intersection utilization compared to baseline methods. These improvements demonstrate that the proposed model not only enhances PA, but also improves the efficiency of real-time TF. This makes the model more suitable for deployment in dynamic urban traffic environments.

Traditional adaptive TSC methods, such as actuated control or fixed-interval schemes with dynamic parameter tuning, typically rely on localized TF data from individual intersections. Although these approaches can respond to short-term variations, they often lack an overall view of traffic conditions, which can result in suboptimal network-wide coordination. By contrast, the proposed cross-cluster partitioning strategy groups intersections with similar temporal flow characteristics. This allows them to operate as coordinated units rather than as isolated nodes. This cluster-based coordination improves signal timing consistency across high-flow corridors and minimizes stop-and-go waves. Compared to traditional methods, this framework demonstrates higher overall accuracy and faster convergence in timing optimization. Furthermore, it supports dynamic reclustering based on evolving traffic patterns. This allows for adaptive boundary adjustments in response to network-wide fluctuations. These features make the proposed approach more scalable and resilient in complex urban environments.

# 5 Discussions

The proposed method demonstrated significant advantages over traditional and state-of-the-art approaches in both PA and clustering effectiveness. The experimental results showed that the BiLSTM model achieves a lower root mean square error than the standard LSTM and CNN-LSTM models. This indicated that the BiLSTM model had a stronger capacity to capture the temporal dynamics of TF sequences. For instance, on the PEMS-BAY dataset the root means square error of BiLSTM reached 0.12 while the standard LSTM and CNN-LSTM reached 0.22 and 0.16, respectively. In terms of clustering, the integration of Gap Statistic with K-means plus feature weighting led to higher silhouette coefficient and Calinski-Harabasz index values compared to baseline clustering methods. These improvements were attributed to the bidirectional temporal modeling which enabled better learning of past and future dependencies in traffic data as well as adaptive clustering which allowed more accurate grouping of intersections with similar flow patterns. Feature weighting further enhanced the influence of key traffic indicators during the clustering process resulting in more stable and meaningful clusters. The combination of these techniques offered strong generalization across multiple datasets and traffic conditions. However, limitations remain. The current system has not been used in real-time scenarios, which may be affected by sensor failures, environmental noise, and communication delays. Moreover, under unusual traffic events such as accidents or festivals the model may encounter difficulties due to its dependence on regular temporal patterns. Future work should focus on deploying this technology in real-world intelligent transportation infrastructures. This should include testing under extreme traffic conditions, evaluating adaptability to unseen scenarios, and integrating real-time anomaly detection modules to improve robustness and system stability.

# 6 Conclusion

Aiming at the challenges of urban TSC optimization, this study proposed an intelligent TSC framework that combined a Bi-LSTM prediction model with an improved K-means clustering algorithm. The framework was evaluated using the PEMS-BAY and CityFlow datasets and achieved strong results in both accuracy and efficiency. On the PEMS-BAY dataset, the Bi-LSTM model reached an accuracy of 97.5 percent and a root mean square error of 0.12, while maintaining a low inference delay of 104 ms. These results showed that Bi-LSTM was capable of capturing temporal patterns more effectively than standard LSTM and CNN-LSTM models. During the clustering stage, introducing the Gap Statistic method and a weighted distance function enabled adaptive selection of the number of clusters and more precise identification of regions exhibiting similar traffic behaviors. The enhanced clustering method achieved an accuracy of 91.2 percent and a silhouette coefficient of 0.47. This indicated stronger internal cohesion and clearer separation between clusters. This integrated approach to traffic prediction and regional signal coordination provided a reliable and practical solution for real-time urban traffic control. Future improvements will include adaptive signal adjustment based on real-time traffic feedback, as well as the incorporation of external factors, such as weather conditions, unexpected events, and vehicle behavior, to enhance system stability and responsiveness further.

Although the CityFlow dataset provides a detailed, scalable simulation environment for urban traffic scenarios, it is still a synthetic dataset that may not fully capture the complexity and unpredictability of real-world traffic systems. Factors such as sensor noise, unpredictable driver behavior, and infrastructure variability are often simplified or omitted in simulations. This poses a limitation to the generalizability of the proposed model. The model shows promising performance on both the CityFlow and PEMS-BAY datasets. Future work will focus on validating the approach in real-world deployments by incorporating live sensor data and assessing robustness under operational uncertainties. A limitation of the proposed control strategy is the lack of integration with real-world actuator constraints, such as minimum green times, maximum cycle lengths, and phase change safety intervals. These constraints are critical for ensuring safety, regulatory compliance, and equipment longevity in .practical deployments. Future work will incorporate these operational boundaries into the control model. This will be achieved through the use of constrained optimization or rule-based post-processing layers. This ensures that the generated signal plans are optimal not only in simulation, but also feasible and deployable in real-world urban traffic systems.

# References

[1] Haitham Y. Adarbah, Mehdi Sookhak, and Mohammed Atiquzzaman. A digital twin-based traffic light management system using BIRCH algorithm. Ad Hoc Networks, 164(10):1.1-1.13, 2024. https://doi.org/10.1016/j.adhoc.2024.103613

[2] Hossein Yektamoghadam, Amirhossein Nikoofard, Masoumeh Behzadi, Mahdi Khosravy, Nilanjan Dey, and Olaf Witkowski. Multi-criteria evolutionary optimization of a traffic light using genetics algorithm and teaching-learning based optimization. Expert Systems, 41(2):21-22, 2024. https://doi.org/10.1111/exsy.13487

[3] Zhichao Yang, Yan Kong, and Chih-Hsien Hsia. DERLight: A deep reinforcement learning traffic light control algorithm with dual experience replay. Journal of Internet Technology, 25(1):79-86, 2024. https://doi.org/10.53106/160792642024012501007

[4] Bharathi Ramesh Kumar, Narayanan Kumaran, Jayavelu Udaya Prakash, Sachin Salunkhe, Raja Venkatesan, Ragavanantham Shanmugam, and Emad S. Abouel Nasr. A dynamic traffic light control algorithm to mitigate traffic congestion in metropolitan areas. Sensors (14248220), 24(12):1-18, 2024. https://doi.org/10.3390/s24123987

[5] JiLin Yu, Zhiwen Wang, and Ruonan Zhang. Diffusion light: a multi-agent reinforcement learning approach for traffic signal control based on shortcut-diffusion model. Applied Intelligence, 55(6):1-25, 2025. https://doi.org/10.1007/s10489-025-06359-8

[6] Xinqi Du, Ziyue Li, Cheng Long, Yongheng Xing, Philip S. Yu, and Hechang Chen. FELight: fairness-aware traffic signal control via sample-efficient reinforcement learning. IEEE Transactions on Automatic Control, 36(9):15-29, 2024. https://doi.org/10.1109/TKDE.2024.3376745

[7] Ruijie Zhu, Wenting Ding, Shuning Wu, Lulu Li, Ping Lv, and Mingliang Xu. Auto-learning communication reinforcement learning for multi-intersection traffic light control. Knowledge-based Systems, 275(5):110696.1-110696.13, 2023. https://doi.org/10.1016/j.knosys.2023.110696

[8] Huajie Yang. Assessing the effects of new light rail transit on regional traffic congestion and transit ridership: a synthetic control approach. IEEE Transactions on Intelligent Transportation Systems, 24(7):7613-7620, 2023. https://doi.org/10.1109/TITS.2022.3168858

[9] Meysam Effati, and Chakavak Atrchian. Examining seasonal changes in light-vehicle traffic volume on freeways under extreme weather conditions: a combination of temporal statistical and data mining non-parametric techniques. Transportation Research Record, 2678(7):50-69, 2024. https://doi.org/10.1177/03611981231203217

[10] Hao Wang, Yun Yuan, Xianfeng Yang, Tian Zhao, and Yang Liu. Deep Q learning-based traffic signal control algorithms: model development and evaluation with field data. Journal of Intelligent Transportation Systems, 27(3):314-334, 2023. https://doi.org/10.1080/15472450.2021.2023016

[11] Sijing Guo, Shengwen Zheng, Ji Li, Quan Zhou, and Hongming Xu. A lightweight social cognitive risk potential field model for path planning with dedicated dynamic and static traffic factors. Intelligent Transport Systems, IET, 19(1):54-57, 2025. https://doi.org/10.1049/itr2.12595

[12] Xingmin Wang, Zachary Jerome, Zihao Wang, Chenhao Zhang, Shengyin Shen, Vivek Vijaya Kumar, Fan Bai, Paul Krajewski, Danielle Deneau, Ahmad Jawad, Rachel Jones, Gary Piotrowicz, and

Henry X. Liu. Traffic light optimization with low penetration rate vehicle trajectory data. Nature Communications, 15(1):1.1-1.3, 2024. https://doi.org/10.1038/s41467-024-45427-4

[13] Gerasimos Rigatos, Masoud Abbaszadeh, Bilal Sari, and Pierluigi Siano. Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 11:100226, 2023. https://doi.org/10.1016/j.rico.2023.100226

[14] Farouk Zouari, Kamel Ben Saad, and Mohamed Benrejeb. Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor. 2013 International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2013:864-871, 2013. https://doi.org/10.1109/CoDIT.2013.6689656

[15] Xin Guo, Zhengxu Yu, Pengfei Wang, Zhongming Jin, Jianqiang Huang, and Deng Cai. Urban traffic light control via active multi-agent communication and supply-demand modeling. IEEE Transactions on Automatic Control, 35(4):17-26, 2023. https://doi.org/10.1109/TKDE.2021.3130258

[16] Anuj Sachan, and Neetesh Kumar. S-Edge: heterogeneity-aware, light-weighted, and edge computing integrated adaptive traffic light control framework. Journal of Supercomputing, 79(13):41-43, 2023. https://doi.org/10.1007/s11227-023-05216-0

[17] Siddhesh Deshpande, and Sheng-Jen Hsieh. Cyber-physical system for smart traffic light control. Sensors (Basel, Switzerland), 23(11):52-61, 2023. https://doi.org/10.3390/s23115028

[18] Ruohan Mi, Jinwei Yu, and Weihua Yang. Coordinated adaptive consensus tracking of multiple impacting oscillator robot systems via distributed interactions. Nonlinear Dynamics, 113(6):5555-5570, 2025. https://doi.org/10.1007/s11071-024-10569-z