2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

MCSTA: Multi-dimensional Collaborative Spatial-Temporal Attention Model for Traffic Flow Prediction

Dazhi Zhao[1], Jinlai Zhang[2(✉)], Kejia Wang[3], Wenguang Wu[2]

[1] School of Physics and Electronic Science, Changsha University of Science and Technology, Changsha, 410114, Hunan, China
[2] College of Mechanical and Vehicle Engineering, Changsha University of Science and Technology, Changsha, 410114, Hunan, China
[3] School of Computer Science and Technology, Changsha University of Science and Technology, Changsha, 410114, Hunan, China

Abstract. Traffic flow prediction is critical for the effective management and public safety of modern cities; however, it remains a challenging task. The intricate spatiotemporal dependencies in traffic data and the trade-off between computational efficiency and predictive accuracy in existing models have long been key challenges. To address these issues, we propose a novel attention-based model built upon the Transformer architecture, termed the Multi-dimensional Collaborative Spatial-Temporal Attention Model (MCSTA). Our model introduces several innovations: first, we design a Lightweight Multi-dimensional Cooperative Enhanced Attention (LMCEA) mechanism to capture spatiotemporal relationships across multiple dimensions. Additionally, we propose Non-dimensionality Reduction Local Cross-Channel Attention (NDLCCA), which leverages 1D convolution to model local cross-channel interactions while circumventing dimensionality reduction operations. This approach significantly reduces computational complexity, enhances the utilization of inter-channel information, accurately captures correlations among channels, and ultimately provides more discriminative feature representations.Experimental evaluations on two real-world traffic datasets demonstrate that MCSTA outperforms state-of-the-art (SOTA) models. Compared to the baseline model, our approach achieves RMSE reductions of 3.43%, 6.63%, and 13.01% on the NYCBike dataset and 6.13%, 7.24%, and 7.49% on the NYCTaxi dataset, respectively.

Keywords: Traffic Flow Prediction, Transformer, Lightweight Attention, Cross-Channel Attention.

1 Introduction

Traffic flow prediction plays a crucial role in the efficient management and operation of modern urban transportation systems [1-5]. With the rapid growth of urbanization and the increasing number of vehicles on the road, accurately predicting traffic flow has become a key challenge for both city planners and transportation authorities. Effective traffic prediction not only helps in optimizing traffic management strategies, but it

also improves public safety by anticipating potential traffic congestions, accidents, and other safety hazards. Moreover, accurate traffic flow forecasting is an essential component of intelligent transportation systems (ITS) that can reduce road congestion, enhance fuel efficiency, and ultimately improve the quality of life for citizens in large urban areas.

Traffic flow prediction involves forecasting the number of vehicles passing through a specific area over a given period. This task is inherently complex due to the spatio-temporal nature of the data involved. In recent years, with the advancement of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), deep learning models based on these architectures have been widely employed in traffic flow prediction. With the emergence of the Transformer model, researchers [6-13] have integrated Transformers with CNNs to enhance traffic flow prediction. In these studies, Transformer networks are utilized to capture temporal dependencies, while CNNs are employed to extract spatial dependencies. Dosovitskiy et al. [14], however, demonstrated the capability of Transformer networks in extracting spatial dependencies. Spacetimeformer [15] adopts a variant similar to Informer [16] to address this issue, yet its performance remains suboptimal. Despite these advancements, existing traffic prediction models still face significant limitations. For instance, although attention-based mechanisms, such as those in Transformer models, offer a more flexible and efficient way to model spatial-temporal dependencies, they often overlook the multi-dimensional nature of traffic data. Traffic flow involves a variety of interrelated factors, such as weather, time of day, road type, and other contextual factors, that need to be considered across multiple dimensions. Additionally, the computational cost of capturing local spatial-temporal correlations, particularly in large-scale traffic networks, remains a challenge.

To address these challenges, we introduce a novel attention model for traffic flow prediction, the Multi-dimensional Collaborative Spatial-Temporal Attention Model (MCSTA). MCSTA builds on the strengths of Transformer-based architectures while introducing several key innovations designed to improve both accuracy and efficiency. At the heart of the MCSTA model are two innovative mechanisms: Lightweight Multi-dimensional Cooperative Enhanced Attention (LMCEA) and Non-dimensionality Reduction Local Cross-Channel Attention (NDLCCA). The LMCEA mechanism is designed to better capture the relationships between spatial and temporal data across multiple dimensions. Unlike traditional attention mechanisms that treat spatial and temporal dependencies separately, LMCEA considers the multi-dimensional nature of the data, allowing the model to leverage both temporal and spatial information simultaneously. This enables the model to effectively capture the intricate relationships that exist within traffic flow data, providing a richer and more discriminative feature representation. Another challenges in applying deep learning models to traffic flow prediction is the high dimensionality of the data, which leads to increased computational complexity. The NDLCCA mechanism is designed to reduce the operational complexity by using 1D convolution to capture local cross-channel interactions. This approach avoids the need for dimensionality reduction operations and ensures that the model effectively utilizes information between channels, making the predictions more accurate while maintaining computational efficiency.

We evaluate the MCSTA model on two real-world traffic datasets: the NYCBike dataset and the NYCTaxi dataset. The experimental results demonstrate that MCSTA outperforms state-of-the-art (SOTA) models, achieving significant reductions in root mean square error (RMSE) when compared to baseline models. Specifically, the RMSE of MCSTA on the NYCBike dataset was reduced by 3.43%, 6.63%, and 13.01%, while the RMSE on the NYCTaxi dataset was reduced by 6.13%, 7.24%, and 7.49% in different evaluation settings. These results confirm that MCSTA is both accurate and efficient, making it a promising model for real-time traffic flow prediction in smart cities.

The contributions of this paper can be summarized as follows:

- We propose LMCEA, which captures temporal and spatial dependencies from multiple dimensions, can adaptively capture local feature interactions, pay attention to different factors in traffic flow data, and extract more discriminative features.
- We propose NDLCCA, using 1D convolution to capture local cross-channel interactions. Avoiding dimensionality reduction greatly reduces the operational complexity, and can make more efficient use of the information between channels, accurately capture the correlation between these channels.

In the structure of this paper, Section 2 describes the prerequisites required for the experiment. Section 3 elaborates on the proposed method in detail. Section 4 conducts an in-depth analysis of the experimental results. Finally, Section 5 summarizes the conclusions of this paper.

2 Preliminary

In this section, we define the origin-destination traffic flow prediction problem.
Based on the established coordinate system, the urban area is evenly and regularly divided into $(I \times J)$ grids.

For the grid (i, j) located in the i^{th} row and j^{th} column, its inflow and outflow within the time interval t are clearly defined as follows:

$$x_t^{in,i,j} = \sum_{g_t^{end}=(i,j)} |Trs| \tag{1}$$

$$x_t^{out,i,j} = \sum_{g_t^{start}=(i,j)} |Trs| \tag{2}$$

where $g_t^{start} = (i,j))([g_t^{end} = (i,j)]$ is used to represent the geographical spatial starting [ending] coordinates of the trajectory in the region (i, j) within the time interval t. Trs: $g_t^{start} \rightarrow g_t^{end}$ represents a set of trajectories, and $|\cdot|$ represents the cardinality of the set. It should be particularly noted that the starting region and the ending region of a trajectory may be the same or different.

At the time interval t, the inflows and outflows of all $I \times J$ regions can be represented by the tensor $X_t \in \mathbb{R}^{2\times I\times J}$, where $(X_t)_{0,i,j} = x_t^{in,i,j}$ and $(X_t)_{1,i,j} = x_t^{out,i,j}$.

We conduct sparse sampling of historical traffic flows from near to far according to three corresponding time perspectives: closeness, periodicity, and trend [17]. When constructing these three perspectives, we select hours, days, and weeks as the key time

steps. For each time perspective, we select a series of key time-step traffic flow matrices and splice them together in sequence along the time axis to construct the input data:

$$X_{closeness} = [X_{t-1}, X_{t-2}, \cdots, X_{t-l_r}] \tag{3}$$

$$X_{period} = [X_{t-p_d}, X_{t-2p_d}, \cdots, X_{t-l_d * p_d}] \tag{4}$$

$$X_{trend} = [X_{t-p_w}, X_{t-2p_w}, \cdots, X_{t-l_w * p_w}] \tag{5}$$

$$X_{history} = [X_{trend}, X_{period}, X_{closeness}] \tag{6}$$

where l_r, l_d, and l_w correspond to the input lengths of the matrices of the three time perspectives respectively, and p_d and p_w are the daily cycle (24 hours) and the weekly cycle (144 hours), respectively.

In addition, external factors such as weather conditions, the day of the week, whether it is a weekend, temperature, and wind speed are all characterized by the One-Hot Encoding method, which is convenient for subsequent analysis and processing.

Based on the given historical observations $X_{history}$ and external factors, achieving accurate prediction of X_t constitutes one of the core problems of this study.

3 Methodology

In this section, we first introduce MCSTA in general. Then we introduce Lightweight Multi-dimensional Cooperative Enhanced Attention (LMCEA) and Non-dimensionality Reduction Local Cross-Channel Attention (NDLCCA).

3.1 Overview of MCSTA

The overall framework of our proposed method is illustrated in Fig. 1. First, we explicitly model the transportation system as a topological graph, where nodes represent traffic sensors or road segments, and edges encode spatial connectivity. This graph-structured representation facilitates the integration of graph convolutional networks (GCNs) with Transformer layers. The input T-step traffic flow data, along with the road network structure, is processed through a Data Embedding Layer to generate spatiotemporal feature representations. Subsequently, the model leverages two core components to enhance feature learning: Transformer-based self-attention, which integrates Lightweight Multi-dimensional Cooperative Enhanced Attention (LMCEA) with Non-dimensionality Reduction Local Cross-Channel Attention (NDLCCA). LMCEA adaptively captures spatiotemporal dependencies across multiple dimensions, dynamically focuses on local feature interactions, and collaboratively weights heterogeneous factors influencing traffic flow. NDLCCA preserves the integrity of original feature information by circumventing dimensionality reduction operations while efficiently extracting fine-grained cross-channel correlations through localized attention mechanisms.

These components iteratively refine feature representations through stacked layers, where the interaction between Key (K) and Query (Q) further optimizes global depend-

encies. Finally, the Output Layer predicts the T'-step traffic flow by synthesizing hierarchical spatiotemporal patterns, significantly enhancing prediction accuracy through discriminative feature fusion and multi-scale dependency modeling.

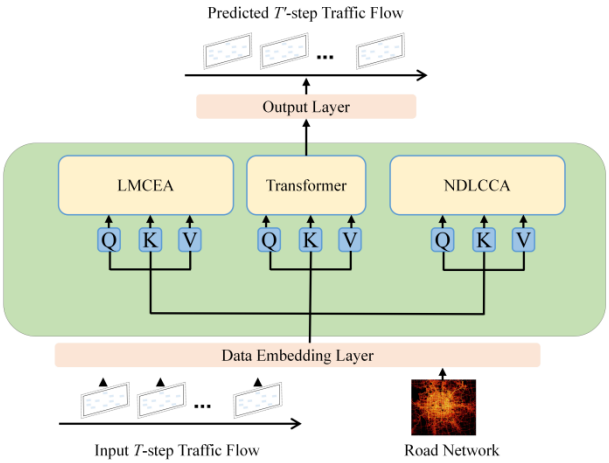

Fig. 1. Overview of the proposed MCSTA.

3.2 Lightweight Multi-dimensional Cooperative Enhanced Attention

Traffic flow prediction poses significant challenges in three key aspects: modeling intricate spatiotemporal dependencies, identifying discriminative feature representations, and ensuring robust model generalization across diverse scenarios. Prediction accuracy is inherently constrained by multiple interdependent factors, whose complex nonlinear interactions are seldom fully captured by existing methodologies. Conventional feature extraction techniques employ oversimplified approaches that are ill-suited for traffic flow prediction, while deep learning models based on CNNs [29-33] and GCNs often exhibit limited generalization capabilities in diverse traffic scenarios. These limitations collectively hinder prediction accuracy and reduce the practical applicability of such models. To address these fundamental challenges, we propose a novel module termed Lightweight Multi-dimensional Cooperative Enhanced Attention (LMCEA). LMCEA is a lightweight, efficient, and generalizable attention mechanism that can be seamlessly integrated into network architectures. As illustrated in Fig. 2, LMCEA comprises three parallel branches: the first two branches capture feature interdependencies along the spatial dimensions W and H, respectively, while the third branch models interactions between channels, thereby enhancing the network’ s ability to extract comprehensive and discriminative feature representations.

LMCEA can be regarded as a highly optimized computational unit, which can achieve a precise and efficient specific transformation from an input tensor to an output tensor of the same shape.

In the top branch, **F** is first rotated counterclockwise by 90° along the H axis, and the resulting rotated feature map is denoted as $\tilde{\mathbf{F}}_W \in \mathbb{R}^{W\times H\times C}$. To accurately model the

long-distance dependency between the channel dimension C and the spatial dimension H, we innovatively introduce an optimized squeeze transformation operation. Input $\tilde{\mathbf{F}}_W$ into it, and the obtained aggregated feature map is still denoted as $\tilde{\mathbf{F}}_W \in \mathbb{R}^{W\times H\times C}$. Subsequently, by using a carefully designed excitation transformation, we deeply capture the feature interactions in the spatial dimension W. The resulting width-oriented feature weights are represented as $\hat{\mathbf{F}}_W \in \mathbb{R}^{W\times 1\times 1}$. Then, $\hat{\mathbf{F}}_W$ generates the attention weights closely related to the input in the W dimension through an efficient sigmoid activation function, which is represented as $\tilde{\mathbf{F}}_W \in \mathbb{R}^{W\times 1\times 1}$. Through an optimized element-wise multiplication operation, α_W is precisely applied to $\tilde{\mathbf{F}}_W$, thus obtaining the enhanced feature map $\mathbf{F}_W \in \mathbb{R}^{W\times H\times C}$. Finally, $\mathbf{F}_W$ is rotated clockwise by 90 degrees along the H axis to obtain a feature map $\mathbf{F}_W^{'} \in \mathbb{R}^{C\times H\times W}$ with the same shape as the original input.In the middle branch, $\mathbf{F}$ is first rotated counterclockwise by $90°$ along the W axis to obtain the rotated feature map $\hat{\mathbf{F}}_H \in \mathbb{R}^{H\times C\times W}$. To accurately depict the dependency between the channel dimension C and the spatial dimension H, and further deeply explore the feature interactions in the height direction, we sequentially apply the optimized squeeze transformation and excitation transformation to $\hat{\mathbf{F}}_H$. By doing so, we can accurately derive the aggregated feature map $\hat{\mathbf{F}}_H \in \mathbb{R}^{H\times 1\times 1}$ and the feature weights in the height direction $\check{\mathbf{F}}_H \in \mathbb{R}^{H\times 1\times 1}$ in sequence. Subsequently, $\check{\mathbf{F}}_H$ is activated by an efficient sigmoid function to generate the attention weights $\mathcal{A}_H \in \mathbb{R}^{H\times 1\times 1}$ that are highly adaptable to the input in the H dimension. $\mathcal{A}_H$ is used to precisely recalibrate $\hat{\mathbf{F}}_H$, thereby obtaining the enhanced feature map $\mathbf{F'}_H \in \mathbb{R}^{H\times C\times W}$. Finally, $\mathbf{F'}_H$ is rotated clockwise by $90°$ along the W axis to obtain a feature map $\mathbf{F''}_H \in \mathbb{R}^{C\times H\times W}$ with the same shape as the original input.

The design of the bottom branch integrates cutting-edge design concepts with the advantages of classic channel attention mechanisms, mainly focusing on deeply modeling spatial dependencies, and it can keenly capture the complex interactions between channels. $\mathbf{F}$ first generates the feature map $\hat{\mathbf{F}}_C \in \mathbb{R}^{C\times H\times W}$ through an optimized identity mapping operation. Then, $\hat{\mathbf{F}}_C$ is input into the optimized squeeze transformation and excitation transformation modules in sequence, and thus the aggregated feature map $\hat{\mathbf{F}}_C \in \mathbb{R}^{C\times 1\times 1}$ and the feature weights in the channel direction $\check{\mathbf{F}}_C \in \mathbb{R}^{C\times 1\times 1}$ are obtained. Next, an efficient sigmoid activation function is applied to $\hat{\mathbf{F}}_C$, and the channel attention weights $\mathcal{A}_C \in \mathbb{R}^{C\times 1\times 1}$ closely related to the input are precisely derived. Subsequently, $\mathcal{A}_C$ is used to precisely rescale $\hat{\mathbf{F}}_C$ to generate the enhanced feature map $\mathbf{F'}_C \in \mathbb{R}^{C\times H\times W}$. Finally, the feature map $\mathbf{F'}_C$ is precisely remapped through the optimized identity mapping function to obtain $\mathbf{F''}_C \in \mathbb{R}^{C\times H\times W}$.

Finally, by averaging the weights, all the outputs of the three branches that have been recalibrated with attention weights in different dimensions are comprehensively processed, so as to obtain the highly optimized final refined feature map. The weight averaging method fully considers the differences in the importance of the outputs of each branch. Compared with the simple average, it can more effectively integrate the key

information of each dimension, significantly improve the model's ability to extract and represent complex traffic flow features, and thus comprehensively enhance the model performance.

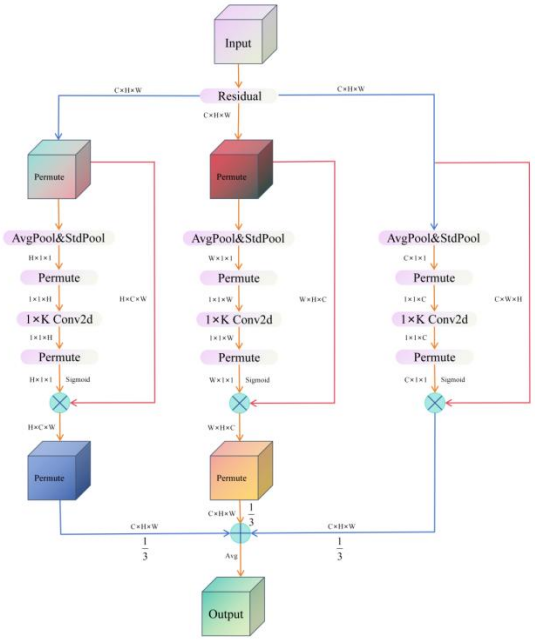

Fig. 2. The structure of LMCEA. ⊗ represents broadcast element-level multiplication, and ⊕ represents broadcast element-level summation.

3.3 Non-dimensionality Reduction Local Cross-Channel Attention

In conventional channel information processing, traditional dimensionality reduction operations disrupt the correspondence between channels and their associated weights, leading to inaccurate information transmission and impairing the ability to capture correlations among various factors. Furthermore, in local cross-channel interactive learning, existing models suffer from excessive parameterization, high computational costs, and a limited capacity to capture only simple adjacent channel relationships, making it challenging to extract long-range local dependencies.

To address these limitations, we propose the Non-dimensionality Reduction Local Cross-Channel Attention (NDLCCA) mechanism. As illustrated in Fig. 3, the input to the NDLCCA module consists of data with dimensions $W \times H \times C$, where W denotes width, H represents height, and C corresponds to the number of channels.

The input data first passes through a Global Average Pooling (GAP) layer. The GAP operation averages the spatial information within each channel, compressing the spatial dimension and thereby obtaining a set of feature vectors with global statistical infor-

mation. Non-dimensional reduction operation helps aggregate feature information, reduces computational load, and enhances the robustness of the features. The set of feature vectors processed by GAP then enters an operation layer controlled by parameter k. Parameter k is used to adjust the feature mapping and transformation, enabling preliminary screening and transformation of features based on the characteristics of the input data, making the features more targeted in subsequent processing. The features processed by the k operation layer are then passed to an activation and normalization layer controlled by parameter σ. Parameter σ is mainly used to adjust the distribution of features, ensuring that the feature values are within an appropriate range through activation functions and normalization operations, preparing for subsequent convolution and feature interaction operations. Subsequently, the data is split into two parallel branches. One branch passes through multiple alternating convolutional layers and Local Cross-Channel Interaction (LCCI) modules. Convolutional layers extract local spatial and channel features by sliding different convolution kernels over the feature data. The LCCI module focuses on mining the interaction relationships among different channels within the same local region. The other branch passes through Adaptive Kernel Size Determination Mechanism (AKSDM) module. After processing by the two parallel paths, the data is merged at the fusion node with the initial data, generating an output with a dimension still of $W \times H \times C$.

1) Local Cross-Channel Interaction

As shown in Fig. 4, in the LCCI module, the input features are initially transformed through a convolutional layer. Then, the transformed features undergo a split operation, dividing them into multiple sub-feature sets. Each sub-feature set is processed by 1D convolution (C1D) to further extract the feature information within the channels. Finally, the processed sub-feature sets are merged through a concatenation (Concat) operation and passed through another convolutional layer for feature fusion and adjustment. This design can effectively enhance the feature representation ability and capture the complex inter-channel dependencies.

2) Adaptive Kernel Size Determination Mechanism

As shown in Fig. 4, in the AKSDM module, the input features are first preliminarily regularized through a convolutional layer to unify the feature expression form. After passing through the convolutional layer, it enters a cascaded structure of N Blocks. Each Block consists of two convolutional layers, with the convolution kernels of each layer being 3×3. The first convolutional layer is used for the preliminary feature extraction of the input features, and after the first convolutional layer, there is a BN layer, which is used to accelerate the network convergence and reduce the internal covariate shift. The second convolutional layer is used to further extract features, and after the second convolutional layer, there is also a BN layer. The input features are added to the output of the second convolutional layer, and then the output is passed through a ReLU activation function. The output of N Blocks is concatenated with the convolutional layer through Concat operation, and then the multi-scale features are integrated through the convolution operation to output the global features and interaction information, achieving the hierarchical extraction of multi-scale features.

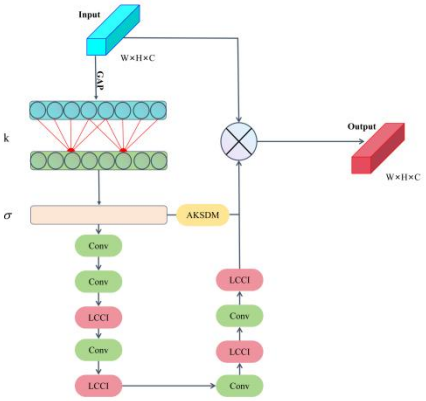

Fig. 3. The structure of NDLCCA.

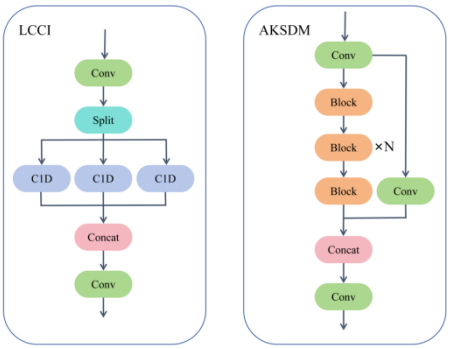

Fig. 4. The diagram of LCCI and AKSDM.

4 Experimental Results

4.1 Datasets

We compared the proposed model to 13 baseline models on two real-world datasets from New York City, NYCBike and NYCTaxi.

NYCBike. The data are shared bike data from New York City. The time range strarts from January 1, 2018, and ends on December 31, 2020, and it contains approximately 56 million track records. We divided New York City into 192 zones. We created three separate subdatasets with time windows of 30 minutes, 60 minutes, and 90 minutes.

NYCTaxi. The data are yellow taxi data from New York City. The time range starts from January 1, 2013, and ends on December 31, 2015, and it contains approximately 416 million track records. We divided New York City into 192 zones. We created three separate subdatasets with time windows of 30 minutes, 60 minutes, and 90 minutes.

For both datasets, we designated all data except the final eight weeks as the training set, the first four weeks of the last eight weeks as the validation set, and the remaining four weeks as the test set.

4.2 Main Results

Table 1 presents the evaluation results for the NYCBike dataset, while Table 2 summarizes the evaluation results for the NYCTaxi dataset. Here, we trained the model directly on datasets with prediction horizons of 30, 60, and 90 minutes. Subsequently, we fine-tuned the 60 minutes and 90 minutes models using pre-trained parameters from the 30-minute dataset, with the corresponding results denoted as 60 min^* and 90 min^*. To facilitate a visual comparison of accuracy levels with previous studies, we selected the classical CNN-based model ST-ResNet as the zero baseline. Table 3 reports the relative error index of MCSTA with respect to ST-ResNet on the NYCBike dataset, and Table 4 provides the corresponding results for the NYCTaxi dataset.

MCSTA consistently outperformed all competing models. As shown in Table 1 and Table 3, on the NYCBike dataset, MCSTA fine-tuned at 60 and 90 minutes achieved RMSE values of 3.66 and 5.28, representing relative reductions of 6.63% and 13.01% compared to the zero baseline. Similarly, as presented in Table 2 and Table 4, on the NYCTaxi dataset, MCSTA trained for 30 minutes achieved RMSE and MAE values of 9.34 and 3.33, with relative reductions of 6.13% and 7.24% compared to the zero baseline. These results consistently surpassed all baseline models. The RMSE and MAE values obtained from direct training on the 90-minute dataset were 28.72 and 8.90, respectively, with MCSTA being the only model achieving an MAE below 9.00 across all baselines. Furthermore, MCSTA attained the highest proportion of optimal values among all competing models, outperforming state-of-the-art (SOTA) approaches in traffic flow prediction.

Table 1. The Prediction Results on NYCBike.

	Models	30 min		60 min		90 min		60 min*		90 min*	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Classical Models	HA	8.90	3.00	17.43	5.76	25.62	8.47	17.43	5.76	25.62	8.47
	ARIMA	10.46	3.52	8.32	2.85	8.97	3.12	8.32	2.85	8.97	3.12
	SimpleExpSmoothing	9.04	3.13	8.07	2.74	12.72	5.44	8.07	2.74	12.72	5.44
GCN	GCN [20]	2.88	1.25	5.29	2.24	7.48	3.10	5.05	2.25	7.62	3.13
	STGCN [21]	2.71	1.16	5.02	2.03	7.66	2.98	4.94	2.02	7.58	2.98
	ASTGCN [22]	2.36	1.08	4.09	1.84	6.17	2.63	4.05	1.82	6.24	2.69
CNN	ConvLSTM [23]	2.30	1.02	4.09	1.75	6.23	2.55	3.85	1.67	5.69	2.43
	ST-ResNet[24]	2.33	1.04	4.10	1.76	6.38	2.64	3.92	1.71	6.07	2.54
	LMST3D-ResNet[25]	2.33	1.03	3.98	1.70	6.06	2.55	3.90	1.69	5.92	2.47
Transformer	Traffic transformer[26]	2.44	1.08	4.30	1.88	6.15	2.69	4.12	1.71	5.86	2.38
	Spacetimeformer[27]	2.34	1.06	3.97	1.74	5.70	2.46	3.83	1.67	5.59	2.38
	Bi-STAT[28]	2.28	**0.99**	3.95	1.64	5.78	**2.28**	3.86	1.63	5.82	2.32
	ProSTformer[18]	**2.24**	**0.99**	**3.82**	1.65	5.62	2.38	3.67	**1.58**	5.29	2.21
	MCSTA (ours)	2.25	1.00	3.83	**1.64**	**5.61**	2.35	**3.66**	1.63	**5.28**	**2.20**

Table 2. The Prediction Results on NYCTaxi.

	Models	30 min		60 min		90 min		60 min*		90 min*	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Classical Models	HA	44.22	12.43	87.25	24.24	129.4	35.84	87.25	24.24	129.4	35.84
	ARIMA	116.2	35.81	104.9	32.02	93.58	28.52	104.9	32.02	93.58	28.52
	SimpleExpSmooth-ing	86.08	27.01	89.49	30.05	65.09	18.69	89.49	30.05	65.09	18.69
GCN	GCN [20]	11.98	4.25	23.72	8.23	35.56	12.65	22.64	7.92	32.49	11.16
	STGCN [21]	11.24	3.90	22.54	7.30	35.57	11.00	21.99	7.20	35.57	11.19
	ASTGCN [22]	10.53	3.74	21.76	7.57	34.69	11.07	21.35	7.02	33.89	10.07
CNN	ConvLSTM [23]	11.72	3.87	22.69	6.71	43.52	10.28	24.25	7.14	46.59	11.00
	ST-ResNet[24]	9.95	3.59	19.47	6.68	30.40	9.76	19.45	6.54	28.59	9.22
	LMST3D-ResNet[25]	10.05	3.53	19.39	6.43	30.09	9.93	18.84	6.27	28.51	9.05
Trans-former	Traffic trans-former[26]	10.84	3.81	21.59	7.01	33.63	10.75	20.29	6.73	30.90	9.55
	Spacetime-former[27]	10.10	3.69	18.85	6.55	32.16	10.11	19.43	6.83	30.86	9.96
	Bi-STAT[28]	9.58	3.41	18.52	6.18	28.80	9.21	18.39	6.14	28.85	9.14
	ProSTformer[18]	9.38	3.34	**18.29**	**5.98**	28.90	9.18	**17.53**	**5.83**	26.58	**8.41**
	MCSTA (ours)	**9.34**	**3.33**	18.71	6.05	**28.72**	**8.90**	17.93	5.84	**26.45**	8.52

Table 3. Results of Relative Error on NYCBike.

	Models	30 min		60 min		90 min	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
Classical Models	HA	281.97%	188.46%	344.64%	236.84%	322.08%	244.09%
	ARIMA	348.93%	203.85%	112.24%	66.67%	47.78%	22.83%
	SimpleExpSmoothing	287.98%	200.96%	105.87%	60.23%	109.56%	114.17%
GCN	GCN [20]	23.61%	20.19%	28.83%	30.99%	23.23%	22.05%
	STGCN [21]	16.31%	11.54%	26.02%	18.13%	24.88%	17.32%
	ASTGCN [22]	1.29%	3.85%	3.32%	6.43%	1.65%	3.54%
CNN	ConvLSTM [23]	-1.29%	-1.92%	-1.79%	-2.34%	-6.26%	-4.33%
	ST-ResNet[24]	0	0	0	0	0	0
	LMST3D-ResNet[25]	0.00%	-0.96%	-0.51%	-1.17%	-2.47%	-2.76%
Trans-former	Traffic transformer[26]	4.72%	3.85%	5.10%	0.00%	-3.46%	-6.30%
	Spacetimeformer[27]	0.43%	1.92%	-2.30%	-2.34%	-7.91%	-6.30%
	Bi-STAT[28]	-2.15%	**4.81%**	-1.53%	-4.68%	-4.78%	10.24%
	ProSTformer[18]	**-3.86%**	**-4.81%**	-6.38%	**-7.60%**	-12.85%	-12.99%
	MCSTA (ours)	-3.43%	-3.85%	**-6.63%**	-4.68%	**-13.01%**	**-13.39%**

Table 4. Results of Relative Error on NYCTaxi.

	Models	30 min		60 min		90 min	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
Classical Models	HA	334.42%	246.24%	348.13%	283.87%	325.56%	267.21%
	ARIMA	1068.04%	897.49%	439.54%	379.34%	207.83%	192.21%
	SimpleExpSmoothing	764.92%	652.37%	359.63%	349.85%	114.11%	91.50%
GCN	GCN [20]	20.40%	18.38%	16.40%	21.10%	13.64%	21.04%
	STGCN [21]	12.96%	8.64%	13.06%	10.09%	24.41%	19.31%
	ASTGCN [22]	5.83%	4.18%	9.77%	7.34%	18.54%	16.05%
CNN	ConvLSTM [23]	17.79%	7.80%	16.66%	2.60%	52.22%	11.50
	ST-ResNet[24]	0	0	0	0	0	0
	LMST3D-ResNet[25]	1.01%	-1.67%	-3.14%	-4.13%	-0.28%	-1.84%
Trans-former	Traffic transformer[26]	8.94%	6.13%	4.32%	2.91%	8.08%	3.58%
	Spacetimeformer[27]	1.51%	2.79%	-3.08%	0.15%	7.94%	8.03%
	Bi-STAT[28]	-3.72%	-5.01%	-5.45%	-6.12%	0.73%	-0.87%
	ProSTformer[18]	-5.73%	-6.96%	**-9.87%**	**-10.86%**	-7.03%	**-8.79%**
	MCSTA (ours)	**-6.13%**	**-7.24%**	-7.81%	-10.70%	**-7.49%**	-7.59%

4.3 Results Analysis

In order to further explore the performance stability and adaptability of MCSTA, we conducted an in-depth analysis of the experimental results, focusing on analyzing multiple key dimensions such as pre-training results analysis, applicability analysis, attention weight analysis, and efficiency analysis:

1) Pre-training Results Analysis: To explore the role of pre-training in traffic flow prediction models, we investigate the effects of pre-training and fine-tuning on the performance of MCSTA models across different duration datasets. In the experiment, we pre-trained the model on 30 minutes datasets and fine-tuned the resulting parameters on 60 minutes and 90 minutes datasets. As shown in Fig. 5 and Fig. 6, pre-trained and fine-tuned models generally perform better in most cases. Taking the NYCTaxi dataset as an example, on 60 minutes data, only the ConvLSTM and Spacetimeformer models, after pre-training, exhibited worse results than direct training. On 90 minutes data, only the ConvLSTM and Bi-STAT models, after pre-training, showed worse performance than direct training. The RMSE results for other models, after pre-training and fine-

tuning, were superior to those of direct training. This suggests that pre-training can help models capture data features more efficiently, thereby enhancing prediction accuracy.

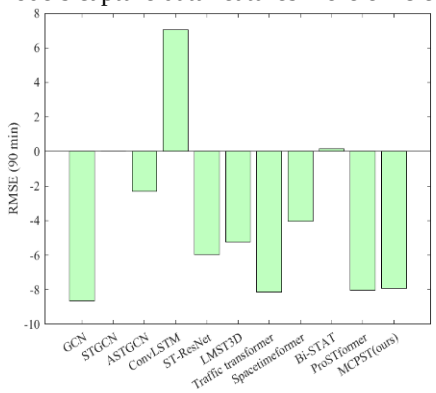

Fig. 5. The error reduction of RMSE on 60 min NYCTaxi with pre-training than with direct training. The negative values mean the RMSE errors decrease.

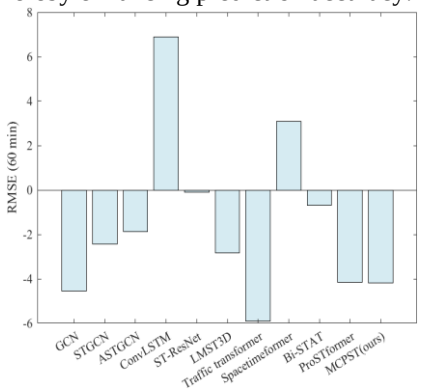

Fig. 6. The error reduction of RMSE on 90 min NYCTaxi with pre-training than with direct training. The negative values mean the RMSE errors decrease.

2) Applicability Analysis: As illustrated in Fig. 7, an in-depth analysis of the model's prediction results reveals that MCSTA exhibits outstanding performance in handling complex traffic flow data. Taking New York City traffic as a case study, the significant regional variations in traffic flow pose substantial challenges for accurate prediction. In the city’s core areas, such as bustling commercial centers and transportation hubs, traffic flow is highly dense, whereas in remote suburbs or restricted zones, it remains sparse. Despite this complexity, MCSTA effectively distinguishes traffic characteristics across different regions. For key areas with high traffic density, the model closely aligns with actual flow trends, providing precise predictions. As shown in Fig. 7 and Fig. 8, between 10:00 – 10:30 a.m. and 6:00 – 6:30 p.m., traffic surges significantly during weekday morning and evening rush hours. MCSTA accurately captures these peak fluctuations and rhythmic changes, producing predictions that closely correspond with real-world data. Moreover, the model performs robustly in regions with low traffic volume. Even during periods where traffic is nearly nonexistent, MCSTA reliably differentiates between true low-flow conditions and potential misjudgments, effectively preventing overestimation or false predictions. This capability is crucial for practical applications, as it enables urban traffic planners to make informed decisions, optimally allocate transportation resources, and avoid unnecessary investments in low-traffic regions. To further illustrate the model’s predictive performance, we conducted a detailed analysis of traffic flow across different time intervals. Whether capturing short-term fluctuations or long-term trends, MCSTA consistently demonstrates stable and accurate predictions.

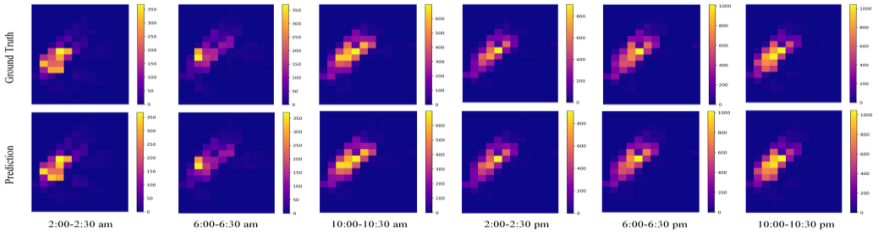

Fig. 7. The prediction results of MCSTA on 30 min NYCTaxi.

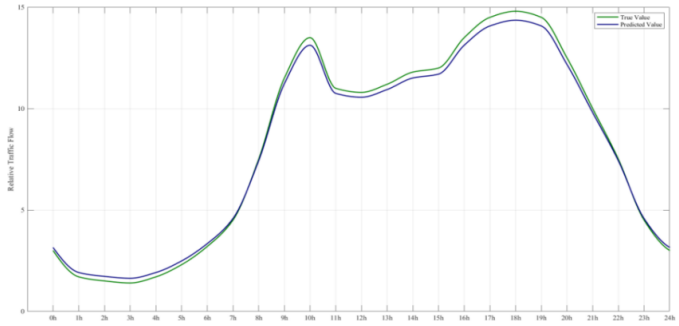

Fig. 8. The true value and predicted value in relative traffic flow in New York City during the day.

4) Efficiency Analysis: As shown in Fig. 9 and Fig. 10, we present the GPU memory usage and total training time for all models during the training phase. For the MCSTA model, both GPU memory usage and training time remain within acceptable limits, while delivering optimal performance. To facilitate comparison and reference, the time complexity and space complexity of each model are summarized in Table 5. In the traffic prediction task addressed in this study, there are no strict requirements regarding real-time inference performance or computing resources. Therefore, we prioritize the accuracy between the predicted and actual values as the core evaluation metric, specifically conducting a quantitative evaluation using RMSE and MAE. This approach aligns with the methodology adopted in most related research.

Table 5. The efficiency comparison of different models.

Models	30 min NYCBikc			30 min NYCTaxi		
	Paramcters	FLOPS	Iteration time(s)	Paramcters	FLOPS	Iteration time(s)
GCN [18]	6.914K	3.834M	1	6.914K	3.834M	1
STGCN [36]	133.666K	169.439M	40	133.666K	169.439M	7
ASTGCN [12]	89.478K	67.830M	7	89.478K	67.830M	7
ConvLSTM [26]	1.038M	2.389G	3	16.538M	38.093G	12
ST-ResNet[39]	2.690M	516.469M	3	2.690M	516.469M	12
LMST3D-ResNet[8]	4.005M	3.073G	4	4.005M	3.073G	4
Traffic trans-former[5]	169.990M	1.020G	3	169.990M	1.020G	3
Spacetime-former[11]	92.290K	210.727M	10	92.290K	210.727M	10
Bi-STAT[7]	551.756K	1.338G	17	551.756K	1.338G	17
ProSTformer[33]	3.623M	1.484G	12	14.273M	2.851G	10
MCSTA (ours)	3.521M	1.377G	12	15.269M	2.649G	12

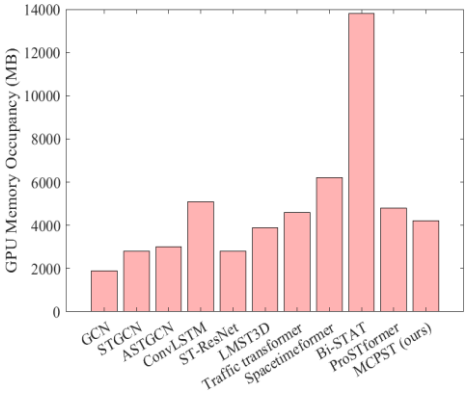

Fig. 9. GPU-Memory occupancy on 30 min NYCTaxi.

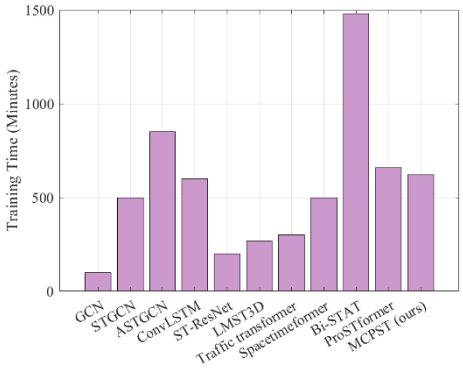

Fig. 10. Training time on 30 min NYCTaxi.

4.4 Ablation Experiments

As shown in Table 6, we conducted additional experiments on MCSTA on NYCBike, taking into account ablation factors.

1) The effect of LMCEA: We removed the LMCEA module from the MCSTA model, leaving only the basic attention mechanism, represented as $MCSTA^{\dagger}$. This modification aims to assess the contribution of the LMCEA module's ability to capture feature dependencies in different dimensions to the overall performance of the model. As shown in Table 6, the model's performance significantly decreases after the removal

of the LMCEA module, indicating that the LMCEA module plays a crucial role in mining the spatial-temporal feature correlations of traffic data and enabling multi-dimensional collaborative prediction.

2) Applicability Analysis: We removed the NDLCCA module from the MCSTA model, leaving only the basic attention mechanism, represented as $MCSTA^{\dagger\dagger}$. This adjustment aims to evaluate the NDLCCA module's ability to prevent dimensional degradation and capture the contribution of cross-channel interactions to the model's overall performance in an efficient manner. As shown in Table 6, the model's performance significantly decreases after the removal of the NDLCCA module, indicating that the NDLCCA module plays a vital role in local cross-channel interactive learning and enhancing prediction accuracy.

Table 6. Ablation Experiment on NYCBike.

Models	30 min		60 min		90 min	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
MCSTA	**2.25**	**1.00**	**3.66**	**1.63**	**5.28**	**2.20**
$MCSTA^{\dagger}$	2.27	1.04	3.82	1.64	5.55	2.29
$MCSTA^{\dagger\dagger}$	2.29	1.11	3.73	1.66	5.75	2.37

5 Conclusion

In this paper, we propose MCSTA based on two key innovations. First, we introduce LMCEA to capture the relationship between temporal and spatial data from multiple dimensions, thereby improving the predictive accuracy of the model. Second, we propose NDLCCA, which uses 1D convolution to capture local cross-channel interactions. By avoiding dimensionality reduction, NDLCCA significantly reduces operational complexity, enhances the effective use of information between channels, and accurately captures correlations between these channels, ultimately providing more discriminative feature information and improving the prediction accuracy. Compared with the baseline model, the RMSE of our model on the NYCBike dataset was reduced by 3.43%, 6.63%, and 13.01%, while the RMSE on the NYCTaxi dataset was reduced by 6.13%, 7.24%, and 7.49%, respectively. These results demonstrate that MCSTA outperformed SOTA models across three subdatasets of NYCBike and NYCTaxi.

In future work, we aim to further enhance the accuracy of MCSTA in predicting traffic flow and expand its application to a wider range of complex real-world scenarios to verify its prediction performance in diverse environments.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (No. 62403076 and 52472399), the Humanities and Social Science Fund of Ministry of Education (No. 24YJCZH416), Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province (New energy intelligent vehicle technology,

2024RC1029) and Hunan NSF Basic Research Project for Young Students (No. 2025JJ60893).

References

1. Zhou, S., Wei, C., Song, C., Pan, X., Chang, W., Yang, L.: Short-term traffic flow prediction of the smart city using 5g internet of vehicles based on edge computing. IEEE Transactions on Intelligent Transportation Systems 24(2), 2229 – 2238 (2022)
2. Navarro-Espinoza, A., Lopez-Bonilla, O.R., Garcia-Guerrero, E.E., Tlelo-Cuautle, E., Lopez-Mancilla, D., Hernandez-Mejia, C., Inzunza-Gonzalez, E.: Traffic flow prediction for smart traffic lights using machine learning algorithms. Technologies 10(1), 5 (2022)
3. Tao, X., Cheng, L., Zhang, R., Chan, W., Chao, H., Qin, J.: Towards green innovation in smart cities: Leveraging traffic flow prediction with machine learning algorithms for sustainable transportation systems. Sustainability 16(1), 251 (2023)
4. A. Chahal, P. Gulia, N. S. Gill, and I. Priyadarshini (2023) A hybrid univariate traffic congestion prediction model for iot-enabled smart city. Information14 (5), 268.
5. Sayed, S.A., Abdel-Hamid, Y., Hefny, H.A.: Artificial intelligence-based traffic flow prediction: a comprehensive review. Journal of Electrical Systems and Information Technology 10(1), 13 (2023)
6. K. I. Ata, M. K. Hassan, A. G. Ismaeel, S. A. A. Al-Haddad, S. Alani, et al. (2024) A multilayer cnn-gruship model based on transformer for spatial- temporal traffic flow prediction. Ann Shams Engineering Journal15 (12), 103045.
7. Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.J., Xiong, H.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020)
8. H. Xu, Q. Hu, G. Tan, Y. Zhang, and Z. Lin (2023) A multi-layer model based on transformer and deep learning for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems25 (1), 443 – 451.
9. Lin, H., Jia, W., Sun, Y., You, Y.: Spatial-temporal self-attention network for flow prediction. arXiv preprint arXiv:1912.07663 (2019)
10. Wang, T., Chen, J., Lu, J., Liu, K., Zhu, A., Snoussi, H., Zhang, B.: Synchronous spatiotemporal graph transformer: A new framework for traffic data prediction. IEEE Transactions on Neural Networks and Learning Systems 34(12), 10589 – 10599 (2022)
11. Xie, Y., Niu, J., Zhang, Y., Ren, F.: Multisize patched spatial-temporal transformer network for short-and long-term crowd flow prediction. IEEE Transactions on Intelligent Transportation Systems 23(11), 21548 – 21568 (2022)
12. Yan, X., Gan, X., Wang, R., Qin, T.: Self-attention eidetic 3d-lstm: Video prediction models for traffic flow forecasting. Neurocomputing 509, 167 – 176 (2022)
13. Vaswani, A.: Attention is all you need. Advances in Neural Information Processing Systems (2017)
14. A. Dosovitskiy (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
15. J. Grigsby, Z. Wang, N. Nguyen, and Y. Qi (2021) Long-range transformers for dynamic spatiotemporal forecasting. arXiv preprint arXiv:2109.12218.
16. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 11106 – 11115 (2021)

17. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
18. Yan, X., Gan, X., Tang, J., Zhang, D., Wang, R.: Prostformer: Progressive space-time self-attention model for short-term traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems (2024)
19. Loshchilov, I., Hutter, F., et al.: Fixing weight decay regularization in adam. arXiv preprint arXiv:1711.05101 5 (2017)
20. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
21. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
22. S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 922 – 929.
23. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems 28 (2015)
24. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: Dun-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems. pp. 1 – 4 (2016)
25. Y. Chen, X. Zou, K. Li, K. Li, X. Yang, and C. Chen (2021) Multiple local 3d cnns for region-based prediction in smart cities. Information Sciences542, 476 – 491.
26. L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu (2020) Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting. Transactions in GIS24 (3), 736 – 755.
27. J. Grigsby, Z. Wang, N. Nguyen, and Y. Qi (2021) Long-range transformers for dynamic spatiotemporal forecasting. arXiv preprint arXiv:2109.12218.
28. C. Chen, Y. Liu, L. Chen, and C. Zhang (2022) Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting. IEEE Transactions on Neural Networks and Learning Systems34 (10), 6913 – 6925.
29. Zhang, J., Meng, Y., Wei, J., Chen, J., & Qin, J.: A novel hybrid deep learning model for sugar price forecasting based on time series decomposition. Mathematical Problems in Engineering, 6507688.(2021).
30. Zhang, J., Meng, Y., Wu, J., Qin, J., Yao, T., & Yu, S.: Monitoring sugar crystallization with deep neural networks. Journal of Food Engineering, vol. 280, 109965.(2020).
31. Wu, X., Meng, Y., Zhang, J., Wei, J., & Zhai, X.: Amodal segmentation of cane sugar crystal via deep neural networks. Journal of Food Engineering, vol 348, 111435.(2023).
32. Lu, G., He, D., & Zhang, J. Energy-saving optimization method of urban rail transit based on improved differential evolution algorithm. Sensors, vol 23, 378. (2022).
33. Wu, J., Zhang, J., Zhu, J., Wang, F., Si, B., Huang, Y., ... & Meng, Y.: Lightweight peach detection using partial convolution and improved Non-maximum suppression. Journal of Visual Communication and Image Representation, 104495. (2025).