mathematics

MDPI

Article

MSA-GCN: Multistage Spatio-Temporal Aggregation Graph Convolutional Networks for Traffic Flow Prediction

Ji Feng [1], Jiashuang Huang [2], Chang Guo [3,*] and Zhenquan Shi [1,*]

[1] School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China; fengji@stmail.ntu.edu.cn
[2] School of Information Science and Technology, Nantong University, Nantong 226019, China; hjshdym@163.com
[3] School of Science, Nantong University, Nantong 226019, China
* Correspondence: guo.ch@ntu.edu.cn (C.G.); szq@ntu.edu.cn (Z.S.)

Abstract: Timely and accurate traffic flow prediction is crucial for stabilizing road conditions, reducing environmental pollution, and mitigating economic losses. While current graph convolution methods have achieved certain results, they do not fully leverage the true advantages of graph convolution. There is still room for improvement in simultaneously addressing multi-graph convolution, optimizing graphs, and simulating road conditions. Based on this, this paper proposes MSA-GCN: Multistage Spatio-Temporal Aggregation Graph Convolutional Networks for Traffic Flow Prediction. This method overcomes the aforementioned issues by dividing the process into different stages and achieves promising prediction results. In the first stage, we construct a latent similarity adjacency matrix and address the randomness interference features in similarity features through two optimizations using the proposed ConvGRU Attention Layer (CGAL module) and the Causal Similarity Capture Module (CSC module), which includes Granger causality tests. In the second stage, we mine the potential correlation between roads using the Correlation Completion Module (CC module) to create a global correlation adjacency matrix as a complement for potential correlations. In the third stage, we utilize the proposed Auto-LRU autoencoder to pre-train various weather features, encoding them into the model's prediction process to enhance its ability to simulate the real world and improve interpretability. Finally, in the fourth stage, we fuse these features and use a Bidirectional Gated Recurrent Unit (BiGRU) to model time dependencies, outputting the prediction results through a linear layer. Our model demonstrates a performance improvement of 29.33%, 27.03%, and 23.07% on three real-world datasets (PEMSD8, LOSLOOP, and SZAREA) compared to advanced baseline methods, and various ablation experiments validate the effectiveness of each stage and module.

Keywords: traffic flow prediction; multistage; randomness interference features; pre-training; spatio-temporal correlation

MSC: 68T01

check for updates

Citation: Feng, J.; Huang, J.; Guo, C.; Shi, Z. MSA-GCN: Multistage Spatio-Temporal Aggregation Graph Convolutional Networks for Traffic Flow Prediction. *Mathematics* **2024**, *12*, 3338. https://doi.org/10.3390/math12213338

Academic Editor: Jonathan Blackledge

Received: 7 October 2024
Revised: 20 October 2024
Accepted: 23 October 2024
Published: 24 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The continuous improvement of living standards has led to the gradual replacement of traditional modes of transportation by automobiles, thereby exerting significant pressure on traffic infrastructure. As an important part of an intelligent traffic management system [1–4], accurate and real-time traffic flow prediction must be applied to traffic management. The biggest challenge to traffic flow prediction remains the complex spatio-temporal correlation modeling.

Traditional algorithms focus on the time dependence of traffic flow. This refers to the fact that traffic characteristics (speed, volume, occupancy, etc.) change over time, and that there is a link between the current value and a previous or even more distant time break. The Autoregressive Integral Moving Average Model (ARIMA) [5,6] and exponential

smoothing method (ES) [7] were used as representative methods to simulate future traffic trends. However, since traffic flow data usually have non-linear spatio-temporal variation patterns, complex seasonality [8], and periodicity [9], traditional models have difficulty capturing these complex traffic patterns, and the data of the predicted target node depend not only on the data of the previous moment, but also on the historical data of the current node over a long period of time, as well as the historical data of the nodes related to it. With the development of deep learning algorithms, the prediction accuracy of recurrent neural networks (RNNs) [10,11] has been further improved due to the feature whereby they can retain the information from the previous longer time step and pass it to the subsequent time step. By comparison, traditional methods only focus on a small period of time before the prediction moment. However, RNNs have the problem of gradient vanishing or explosion when dealing with long-term dependencies, so GRUs and Long Short-Term Memory (LSTM), which are based on gating units to control the information transfer and have a certain degree of mitigation function for the problem of gradient vanishing and explosion, have been widely used in the field of traffic flow [12–14]. It is important to note that a traffic flow prediction model that focuses only on temporal dependence and ignores other factors such as spatial correlation, external events, or special circumstances will result in an oversimplified and inaccurate prediction from the model. This means that, while the model can capture periodic changes in traffic flow over time, such as daily peak periods, it will ignore important factors such as accidents, weather, road works, and special events, making it difficult for the model to cope with actual traffic congestion or changes. Therefore, a complete traffic flow prediction model needs to take into account the time dependency as well as other relevant factors to provide more accurate predictions and feasibility for practical applications.

For the above reasons, the consideration of spatial correlation has been initiated by researchers in the analysis of traffic flow data [15–17]. Spatial correlation refers to the various potential interconnections and interactions between different transportation nodes. Inspired by the field of image processing [18,19], some researchers have used a convolutional neural network [20,21] to capture spatial correlation and combine it with modules focusing on time-dependent relationship modeling, such as a Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM) modules [22,23], to predict the future traffic conditions of roads. The characteristics of the CNN lead to its inability to be more useful for road network structures in non-Euclidean space [24]. The GCN-based approach provides a new idea for mining road network data in non-Euclidean space, where nodes can pass information through edges between their neighboring nodes at each time step. This means that a node can update its state according to its neighboring nodes at the current time point, reflecting dynamic changes.

The emergence of GCNs has largely promoted the development of the field of traffic flow prediction. Researchers have abstracted the road network structure information into adjacency matrices for the purpose of capturing spatial features and achieved better prediction results. However, there are some problems with existing graph-convolution-based studies:

(1) Previous studies used the base information of road networks for spatio-temporal feature modeling, for example, modeling the connectivity graph structure of roads, the distance graph structure, and so on. These graph structures reflect the most important and basic information of roads, but there are also some potential spatio-temporal information graph structures in the road network that have not been explored [25–28]. For example, two road nodes are connected to each other, but there are fewer vehicles traveling to and from them, and the correlation is low; the connectivity graph structure cannot reflect this potential correlation information. Therefore, it is necessary to mine different potential graph structures in road networks to enrich the spatio-temporal correlation features.

(2) Some studies have used the similarity graph structure in their attempts to enrich these graph structures, where information between two nodes with higher similarity

can be cross-referenced. Some researchers have used the DTW algorithm for mining this potential graph structure so that information between nodes can be referenced to each other during graph convolution [29–31]. However, when similarity features are used as a graph structure to mine the corresponding spatio-temporal correlation features, the traffic flow features may change between different traffic nodes due to some external reasons, which may cause similarity distortion. Therefore, the problem where similarity features are interfered with by randomness also needs to be solved.

(3) Some external factors, such as weather, POI, etc., also have an impact on the prediction process of traffic flow [32–34]. Incorporating them into the model aims to better simulate the external conditions of real-world traffic and improve the interpretability of the model. Therefore, it is essential to refine and incorporate this information into the model.

(4) Based on the above problems, it is necessary to propose a model that can take into account the multi-graph structure, dig deeper into the information of the graph structure, and simulate the real-world situation.

To this end, the entire model is divided into four stages to address different issues. In the first stage, we model the spatial similarity features of traffic characteristics and use the proposed CGAL module (ConvGRU Attention Layer Module) and CSC module (Causal Similarity Capture Module) for secondary randomness filtering. In the second stage, we model the correlation between all nodes in the road network from the perspective of global correlation through the CC module, serving as a complement to overall spatio-temporal correlation. In the third stage, we utilize four types of weather data, performing preprocessing in the WFC module (Weather Factor Capture Module) and introducing an Auto-LRU pre-training network based on an encoder–decoder architecture to train the weather features, resulting in features obtained from the trained encoder. In the fourth stage, we fuse the features from the first three stages to obtain complex spatio-temporal fused features, then model temporal dependencies using BiGRU, and finally output the prediction results through a fully connected layer.

(1) This paper presents a method called MSA-GCN: Multistage Spatio-Temporal Aggregation Graph Convolutional Networks for Traffic Flow Prediction. The aim is to address potential spatio-temporal correlation features in a phased manner by incorporating multiple graph structures, mitigating the interference of randomness on similarity features, and integrating weather information to ultimately predict traffic characteristics.

(2) In the first stage, this paper designs the CGAL module (ConvGRU Attention Layer Module) and the CSC module (Causal Similarity Capture Module) to model similarity features and perform the first round of randomness filtering, alongside a second round using weak causal relationship node information to reduce the randomness's impact. In the second stage, the CC module (Correlation Completion Module) is designed to construct a correlation adjacency matrix for capturing global correlation features.

(3) In the third stage, this paper introduces an Auto-LRU pre-training network within the WFC module (Weather Factor Capture Module) to perform encoding and decoding operations on weather features. This process learns the latent representations of the features, filtering out noise in the data to enhance overall data quality.

The rest of the paper is organized as follows: Section 2 introduces the classical approach of the three stages of the traffic flow prediction model; Section 3 gives the relevant definitions of the traffic flow prediction model; Section 4 introduces the framework of the MSA-GCN model, as well as the principle roles of the individual modules in the model; and Section 5 describes the relevant setups of the experiments and compares them with some advanced and traditional models. The experimental results are analyzed, and the roles of each module and the advancement of the model are summarized through ablation experiments. In Section 6, we summarize the work performed.

2. Related Work

Traffic flow prediction, as a key theoretical underpinning for urban transport planning and management, has undergone significant development and evolution over the past few decades and is now dominated by a combination of methods based on machine learning and deep learning (Table 1).

Table 1. Summary of spatio-temporal traffic flow prediction models.

Methods	Road Graph Structure Type	Predicted Traffic Characteristic Categories	Dataset
T-GCN	Distance	Speed	LOSLOOP, SZ-taxi
STSGCN	Distance	Flow	PEMSD3, PEMSD4, PEMSD7, PEMSD8
T-LGGCN	Gaussian Kernel Threshold Matrix, Correlation Matrix	Speed	PEMSD4, PEMSD8
ASTGCN	Distance	Flow	PEMSD4, PEMSD8
STDN	Departure OD Matrix, Arrival OD Matrix	Flow	NYC-taxi, NYC-bike
TmS-GCN	Adjacency Matrix, Traffic Flow Propagating Matrix, Centroid Distance Matrix	Speed	SZAREA
TGC-LSTM	k-hop Neighborhood Matrix, Free-Flow Reachable Matrix	Speed	LOOPf, INRIX
T-MGCN	Topological Matrix, Weight Topological Matrix, Traffic Pattern Matrix, Functionality Matrix	Speed, Congestion	HZJTD, PEMSD10
DSTGCN	Connection, Weather, POI	Traffic Accident	Beijing-Taxi GPS
UST-GCN	Cross Spacetime Edges Matrix	Speed	PEMSD4, PEMSD7, PEMSD8
KST-GCN	Connection, POI, Weather, Knowledge Graph	Speed	SZ-taxi
STGCGRN	Gaussian Kernel Threshold Matrix	Flow	PEMSD3, PEMSD4, PEMSD7, PEMSD8

2.1. Traditional Model

The Autoregressive Integral Moving Average (ARIMA) model [35–37] is a classical time series forecasting method that is remarkably effective in capturing the trends, seasons, and autocorrelation of traffic flow data and is efficient and easy to implement when making short time forecasts. The Support Vector Machine (SVM) [38–40] can be used to solve regression-type problems and is suitable for traffic prediction tasks that contain multiple influencing factors. At the same time, it can effectively deal with non-linear relationships and map low-dimensional features to high-dimensional space for regression. The BP neural network [41,42], applied to the field of traffic flow prediction, has also achieved better prediction results, and it can effectively capture non-linear factors such as time, weather, and events, and learn the features in the input data automatically. The vector autoregressive model (VAR) [22,43] is also a traditional statistical time series modeling method which is usually used to analyze and predict the relationship between multiple related time series variables.

The main advantage of these traditional traffic flow prediction methods is that the model structure is simple and easy to implement; however, they are sensitive to the noise and outliers in the data. The actual situation is that the number of traffic data may be even larger, and limited by the accuracy of the data collection equipment, and this shortcoming may be amplified, resulting in the poor prediction of the traditional methods.

2.2. Deep Learning Model

With the rise of deep learning, a CNN [44,45] was first applied to processing image data to automatically learn and extract information about traffic features through convolutional operations. Since road networks are typically made up of non-Euclidean spatial

data, deeper spatial correlations could not be extracted. As GCNs have been widely used in classification, clustering, and prediction [46–50], researchers have found that they can achieve good performance with a lower training cost while compensating for the shortcomings of CNNs that are unable to fully and comprehensively capture spatial features and that their ability to aggregate information makes the models more relevant to real-world situations. A GCN is often used in the field of traffic flow in conjunction with other deep learning modules. The main idea of a GRU [51–53] is to effectively capture long-standing relationships in sequences through a gating mechanism and largely solve the problems of gradient vanishing and gradient explosion [54,55]. Compared with LSTM, a GRU has fewer parameters, and the training efficiency is greatly improved. A Graph Attention Network (GAT) [56,57] weighs the connectivity relations of different nodes through the attention mechanism to aggregate information in a more targeted way.

Deep learning solves the problem of the presence of noise and outliers in the data as well as a large number of data, but the use of a single deep learning module does not do a good job of capturing potential spatio-temporal correlations from different perspectives. In the case of GCNs, for example, some studies have used only a single view as input to mine spatio-temporal correlations, which led to poor predictions due to the lack of comprehensive and adequate capturing.

2.3. Composite Models for Deep Mining of Spatio-Temporal Correlations

With the widespread promotion of deep learning, researchers have found that it is helpful to learn different features in order to improve the model prediction performance by stacking different deep learning modules. Zhu [58] proposed a spatio-temporal traffic speed prediction model, T-GCN, based on a GCN and GRU which predicts the future traffic speed on a certain road by establishing the spatial correlation through the GCN and modeling the temporal dependence through the GRU. Song [59] proposed a spatio-temporal synchronous graph convolutional network, STSGCN, to effectively capture local spatio-temporal correlations through a spatio-temporal synchronous modeling mechanism. With the deepening of the research, some researchers began to dig deeper into spatio-temporal correlation based on composite models. Feng [60] proposed a T-LGGCN model that integrates global and local spatial correlation, constructs the global correlation matrix, and captures it using a GCN. For the local spatio-temporal components, the local correlation is modeled through the fully connected layer, and the GCN combined with local correlation, and the prediction result is generated through the GRU. Guo [61] proposed the ASTGCN traffic flow prediction model, which divides historical traffic data into recent, daily, and weekly cycles and predicts the traffic conditions by modeling the spatio-temporal correlation of the three channels. Yao [62] proposed the STDN model to address the problem where existing studies rely heavily on the assumption that the spatial correlation is smooth in time and introduced a flow-gating mechanism to learn the dynamic similarity between locations. Yang [63] proposed the TmS-GCN model to address the problem of traffic speed prediction at the regional level, proposing a traffic flow propagation adjacency matrix and a center-of-mass distance adjacency matrix to deepen the spatio-temporal correlation between regions. Cui [64] proposed the TGC-LSTM model to learn the physical network topology-based traffic map convolution in the road network by defining the interactions and predicting the traffic state of the whole road network. Yu [65], based on real-world accident records, meteorological conditions, and point-of-interest distribution data, proposed a new deep spatio-temporal graph convolutional network, DSTGCN, to predict traffic accidents. Lv [11] proposed the T-MGCN model, identifying multiple hidden semantic correlations and encoding non-Euclidean spatial correlations and heterogeneous semantic correlations between roads for multiple graphs, which achieved good prediction results on two traffic datasets. Amit [66] proposed the UST-GCN model, which can connect the spatial adjacency matrices of all timestamps of a specific time window into a single spatio-temporal adjacency matrix, achieving unified spatio-temporal aggregation and saving computational costs. The KST-GCN model proposed by Zhu [67]

introduces the concept of the Knowledge Graph, and, in combination with the proposed fusion cell, KF-Cell, it inputs different information into the prediction model and obtains better prediction results. The STGCGRN proposed by Zhao [16] uses a multi-head attention mechanism to capture the hidden spatial dependencies.

The combined approach of these deep learning modules outperforms both traditional and deep learning methods in predicting spatio-temporal correlations due to the modeling and capturing of spatio-temporal correlations from different perspectives and at different depths.

3. Problem Definition

This paper defines the road network graph structure as an undirected graph $G = \{V, E, A, X\}$ where V is the set of nodes of the graph G. $V = \{v_1, v_2, \ldots, v_n\}$, where V_i represents the node numbered i, and n represents the number of nodes. E is the set of edges of the graph G, and, if there exists an edge e_k between two nodes v_i and v_j, which is added to E, $|E| = p$ represents the number of elements in E and then the set of edges is denoted as $E = \{e_1, e_2, \ldots, e_p\}$. $A \in R^{n \times n}$ is the adjacency matrix, and n represents the number of nodes in the graph G; the traffic feature matrix $X \in R^{S \times n}$, where S is the total number of time slices. x_i^t represents the eigenvalue of the traffic corresponding to the i-th sensor at the moment t. The prediction problem can be defined as the learning function F:

$$F(A, (X_{t-K}, X_{t-K+1}, \ldots, X_t)) = [X_{t+1}, X_{t+2}, \ldots, X_{t+T}] \quad (1)$$

On a road topology network graph G, the eigenvalues of K time slices before the current moment t are used to predict the traffic eigenvalues of T time slices after the moment t. $[X_{t+1}, X_{t+2}, \ldots, X_{t+T}]$ are the predictions, and G and $(X_{t-K}, X_{t-K+1}, \ldots, X_t)$ are the inputs to the model.

4. MSA-GCN Model Structure

The method proposed in this paper is divided into four phases, as shown in Figure 1. In the preparation phase, this model encodes the features of the graph structure and the traffic graph features via a GCN to obtain the traffic feature representation with spatial information. The first phase contains the CGAL module and CSC module for the secondary filtering of traffic features with randomness interference in the DTW algorithm, which is described in detail in Section 4.1. The second stage contains the CC module for modeling the spatial correlation of global road nodes and obtaining the correlation feature representation complementing the potential spatio-temporal correlation features, described in detail in Section 4.2. In the third stage, the WFC module is included for fusing different weather information and capturing the key information through the pre-trained Auto-LRU network to obtain the optimized weather feature representation, which is described in detail in Section 4.3. In the fourth stage, by fusing the different spatio-temporal feature representations obtained in the first three stages and focusing on time-dependent modeling based on a BiGRU, the prediction results are outputted through the fully connected layer, which is described in detail in Section 4.4. Finally, the algorithmic description of MSA-GCN is described in Section 4.5.

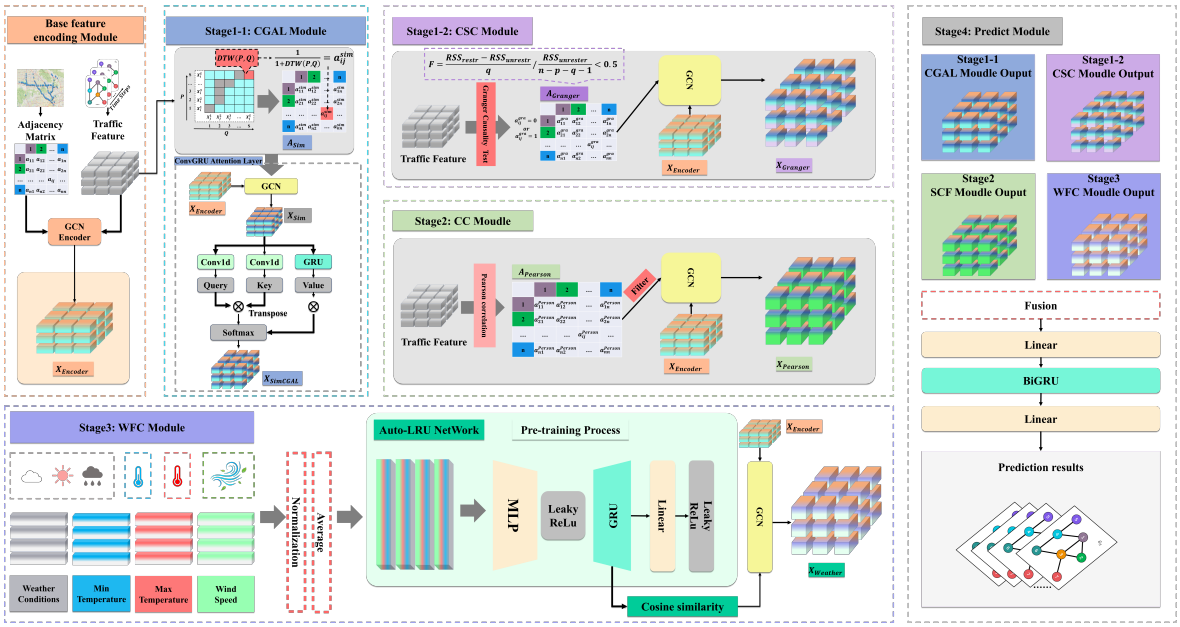


Figure 1. MSA-GCN model framework.

4.1. Stage 1 Randomness Filtering

In the first stage, as shown in Figure 2, the MSA-GCN uses a GCN to encode the original features, and the CGAL module and CSC module are designed to capture similarity features and reduce the problem of randomness interference in predictions twice.

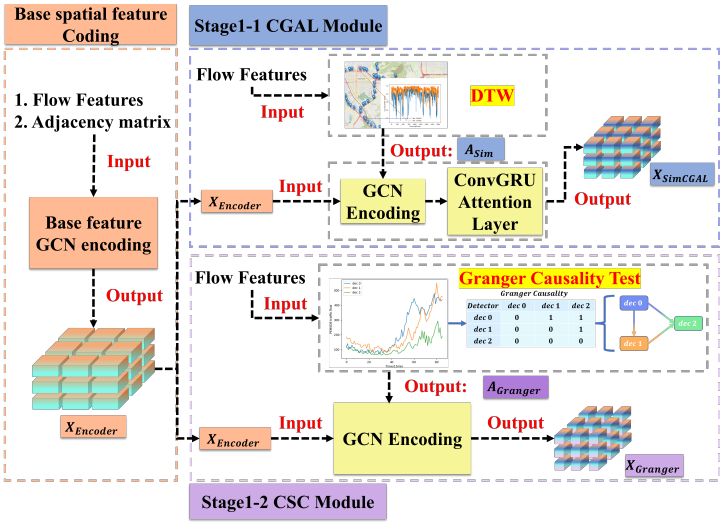


Figure 2. Stage 1 flowchart.

4.1.1. Base Spatial Feature Coding

The GCN is able to effectively capture spatial dependencies and topological information between nodes. Through the operation of graph convolution, it can combine the features and spatial information of the nodes to generate more representative features. The GCN embedding expression is shown in Equation (2).

$$X_{Encoder}^{(l+1)} = ReLU(\widetilde{A}X_{Encoder}^{(l)}W^{(l)}) \tag{2}$$

where $X_{Encoder}^{(l)}$ is the feature matrix of layer l, $X_{Encoder}^{(0)}$ is the input node features, $\widetilde{A}$ is the normalized adjacency matrix, and $W^{(l)}$ is the learnable weight matrix for layer l. A is either the distance adjacency matrix or the connectivity adjacency matrix, depending on

the specific dataset; PEMSD8 and LOSLOOP provide distance adjacency matrices, while SZAREA provides connectivity adjacency matrices. $X_{Encoder}$ will be used in the subsequent spatio-temporal feature modeling process.

4.1.2. CGAL Module: ConvGRU Attention Layer Module

As a method for solving similarity feature problems, the DTW (Dynamic Time Warping) algorithm plays an important role in the field of graph convolution research for traffic flow prediction. Traffic flow, as typical time series information, often exhibits irregular fluctuations and varying lengths. The DTW algorithm is widely used because it can effectively overcome these issues and compute the similarity between two time series. However, in the real world, sudden events may cause abrupt changes in node traffic, which may result in two nodes with originally low similarity being assigned a higher similarity through the algorithm, leading to some degree of model distortion. We refer to these sudden events as stochastic disturbances. In this section, we propose a CGAL module to perform an initial filtering of such randomness, and, in the next subsection, we use Granger causality tests for a second filtering. Below is an introduction to the DTW algorithm and the CGAL module.

$$D(i,j) = d(i,j) + \min \begin{cases} D(i-1,j) \\ D(i-1,j-1) \\ D(i,j-1) \end{cases} \tag{3}$$

For two roads P and Q corresponding to an equal length of historical traffic data time series $P = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$, $Q = \{y_1, y_2, \ldots, y_j, \ldots, y_n\}$. x_i and y_j represent the i-th and j-th element of the corresponding sequence, respectively. $d(i,j)$ represents the Euclidean distance between two traffic features. i and j represent indexed positions in X and Y, respectively. The Euclidean distance $d(i,j) = |x_i - y_j|$ between the corresponding elements x_i and y_j in the two time series is calculated before the distortion matrix D is iteratively solved as Equation (3).

$$A_{Sim} = 1/(1 + DTW(X_i, X_j)) \tag{4}$$

A smaller $DTW(P,Q)$ represents a higher similarity between P and Q and vice versa. After calculating the DTW distance, the distance is usually converted to similarity by mapping the values of the elements to the range of 0 to 1 so that the larger the value, the higher the similarity. The mapping formula is Equation (4).

As shown in Equations (5)–(7), the CGAL module encodes the similarity adjacency matrix A_{Sim} obtained through the DTW algorithm, and then uses an improved self-attention mechanism for the first round of randomness filtering. We replaced the traditional self-attention mechanism's Query, Key, and Value with 1D convolution and a GRU. This choice was made because sudden events usually cause rapid changes in traffic features over a short period, and 1D convolution, when applied to time series, performs a form of local smoothing, helping to eliminate noise signals caused by randomness. The GRU replaces the Value part of the self-attention mechanism because, through its gating mechanism, the GRU can retain key information over long periods and is less likely to be disturbed by short-term random fluctuations.

$$X_{Sim} = \mathrm{GCN}(A_{Sim}, X_{Encoder}) \tag{5}$$

$$[Q,K,V] = [\mathrm{Conv1d}(X_{Sim}, W_q), \mathrm{Conv1d}(X_{Sim}, W_k), \mathrm{GRU}(X_{Sim}, W_v)] \tag{6}$$

$$X_{SimCGAL} = \mathrm{Softmax}(\frac{Q \cdot K^T}{\sqrt{d^{\frac{1}{2}}}}) \cdot V \tag{7}$$

where Conv1d represents a 1D-CNN operation, A_{Sim} is the similarity comprehension matrix, W_q, W_k, W_v are trainable weight matrices, and Softmax converts the elements of a vector into a probability distribution such that the sum of all its elements is 1. $\sqrt{d^{\frac{1}{2}}}$ is

suitable for controlling the dot product results of Query and Key, preventing gradient vanishing or exploding, and $X_{SimCGAL}$ is the output of the CGAL module.

The initial encoding feature $X_{Encoder}$ and the fused feature X_{Sim} obtained from A_{Sim} through the DTW algorithm are filtered for randomness once in the CGAL module, which helps the model to be more focused on the extraction of key information. The randomness is later filtered a second time by the CSC module. W_q, W_k, W_v are trainable weight matrices.

4.1.3. CSC Module: Causal Similarity Capture Module

In this section, the proposed model performs a second filtering through the CSC module for randomness issues not considered by the DTW algorithm and randomness disturbances in the data themselves. After the CGAL module performs the first round of randomness filtering, there may still be some noise or short-term changes caused by sudden events that have not been completely filtered out. In the CSC module, the Granger causality test further filters out those features that, despite being correlated in the short term, do not have a causal relationship. This provides a stricter filtering criterion for the CGAL module, ensuring that the features ultimately input into the model are more stable and reliable.

The application of the Granger causality test to time series is used to test whether a certain time series of data is predictive of another time series. Firstly, given any two time series M and N in the matrix of historical traffic characteristics, restricted and unrestricted models are built and fitted. The restricted model contains the current variables to be tested and their historical lagged values but no other predictor variables. The unrestricted model contains all the independent variables of the restricted model and includes other possible predictor variables.

$$\begin{cases} M_t = a_1 M_{t-1} + a_2 M_{t-2} + \ldots + a_p M_{t-p} + e_1 \\ MN_t = a_1 M_{t-1} + a_1 M_{t-2} + \ldots + a_p M_{t-p} + b_1 N_{1,t} + \ldots + b_q N_{q,t} + e_2 \end{cases} \tag{8}$$

where M_t is the current variable to be tested, M_{t-1} to M_{t-p} are its values in the past p time slices, $a_0, a_1, \ldots, a_p$ are the coefficients of the "limiting model", and e_1 is the error term. $n_{1,t}$ to $N_{q,t}$ are the predictor variables of all possibilities, the predictor variables $b_0, b_1, \ldots, b_q$ are the coefficients of the unrestricted model, and e_2 is the error term. The two models are then fitted separately by least squares or other fitting methods, where p and q are the lag orders of M and N, respectively.

Next, the residual sum of squares RSS_{restr} and $RSS_{unrestr}$ are calculated and the F-statistic is computed from the residual sum of squares:

$$RSS_{restr} = sum(e_1^2) \tag{9}$$

$$RSS_{unrestr} = sum\left(e_2^2\right) \tag{10}$$

$$F = \frac{RSS_{restr} - RSS_{unrestr}}{q} / \frac{RSS_{unrester}}{n - p - q - 1} \tag{11}$$

where n is the number of time steps in the time series.

Based on the calculated p-value of the F-statistic, if it is less than the pre-determined significance level of 0.05, the original hypothesis is accepted, i.e., there is a causal relationship, and the value of the corresponding element in the $A_{Granger}$ is $a_{ij}^{gra} = 1$. Otherwise, the original hypothesis is rejected, i.e., there is no causal relationship, and $a_{ij}^{gra} = 0$. Then, $A_{Granger}$ and $X_{Encoder}$ are input into the GCN to encode into the feature matrix $X_{Granger}$ with causality, as shown in Equation (12).

$$X_{Granger} = \mathrm{GCN}(A_{Granger}, X_{Encoder}) \tag{12}$$

where $X_{Granegr}$ is the last output of the CSC module and the first stage.

4.2. Stage 2 CC Module: Correlation Completion Module

At this stage, we complemented potential spatial correlations by constructing a Pearson correlation adjacency matrix $A_{Pearson}$ with low correlation filtering and spatial feature coding ($X_{Pearson}$).

$$A_{Pearson} = \begin{cases} r_{V_xV_y} = \frac{Cov(V_x,V_y)}{S_X S_Y} & \text{Correlation between } x \text{ and } y \\ 0 & r_{V_xV_y} < \lambda_{dataset} \end{cases} \tag{13}$$

$$X_{Pearson} = \text{GCN}(A_{Pearson}, X_{Encoder}) \tag{14}$$

The time series samples of any two sensors are $V_x = \{X_1, X_2, \ldots, X_n\}$, $V_y = \{Y_1, Y_2, \ldots, Y_n\}$. $Cov(V_x, V_y)$ represents the covariance of the two samples, and S_X and S_Y represent the standard deviation of the two samples, respectively. $r_{V_xV_y}$ represents the magnitude of the correlation between the two sensors computed by sensor x and sensor y using historical traffic data, and a_{xy}^{cor} is the magnitude of the correlation between sensors x and y after filtering with a high threshold. The $\lambda_{dataset}$ is the corresponding threshold value chosen according to different datasets. $X_{Pearson}$ is also the last output of the CC module and the first stage.

4.3. Stage 3 WFC Module: Weather Factor Capture

The type of weather (sunny, cloudy, light rain), high and low temperatures, and wind speed are all external factors that can influence the change in traffic status. The WFC module includes several components, including data mapping, Auto-LRU network, and result mapping components.

The PEMSD8 dataset exhibits a large difference in the traffic flow collected by the same sensor on two days with a 10-degree difference in maximum temperature (19 July and 27 July). Traffic volumes for a day with a maximum temperature of 38 degrees Celsius were generally lower than traffic volumes for a day with a maximum temperature of 28 degrees Celsius. These external factors are also introduced to deeply explore the potential changing patterns of traffic data, make the model more relevant to the actual situation, and obtain more accurate prediction results.

To this end, for the PEMSD8 dataset, we used the data of Bao et al. [68], accessed through https://www.ventusky.com (accessed on 28 September 2023), to fill in some of the missing data, and, finally, a total of four types of data are used: daily maximum temperature, daily minimum temperature, daily weather conditions, and daily wind speed magnitude based on the convolutional neural network and fully connected neural network. An Auto-LRU network structure is built to mine the intrinsic patterns of these data, and they are fused and mapped into weather influencing factors that are added to the training of the model:

$$X_{Fusion} = \text{Mean}[\text{Normalization}(X_{MaxT}, X_{MinT}, X_{Wind}, X_{WeatherType})] \tag{15}$$

$$X_{Encoder}^{Weather} = F_{af}(\text{Linear}(X_{Fusion})) \tag{16}$$

$$X_{Decoder}^{Weather} = \text{GRU}(X_{Encoder}^{Weather}) \tag{17}$$

$$X_{Decoder}^{Output} = F_{af}(\text{Linear}(X_{Decoder}^{Weather})) \tag{18}$$

As shown in Equations (15)–(18), different weather information is first mapped into four feature matrices, $X_{WeatherType}$, X_{MaxT}, X_{MinT}, and $X_{Wind} \in R^{S \times n}$, and fused after they are normalized. Then, they are fused in the unsupervised training of the Auto-LRU after encoder and decoder layers to map the data to a low-dimensional space and then reduce them to achieve the purpose of feature extraction, removing redundant information and generating new samples. F_{af} is the activation function LeakyReLU.

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{19}$$

$$A_{Weather} = \cos(\theta)(X_{Decoder}^{Output}) \tag{20}$$

$$X_{Weather} = \mathrm{GCN}(A_{Weather}, X_{Encoder}) \tag{21}$$

$$X_{Weather} = \mathrm{CNN}(X_{Weather}) \tag{22}$$

As shown in Equation (18), after the training, the generated new sample is extracted as the weather feature $X_{Decoder}^{Output}$, and, as shown in Equations (19)–(22), the weather correlation adjacency matrix $A_{Weather}$ is derived using cosine similarity $\cos(\theta)$. Subsequently, it is input into the GCN together with $X_{Encoder}$ for spatial feature encoding to obtain the weather feature $X_{Weather}$, which is the final output of the WFC module.

4.4. Stage 4 Prediction Module

In this stage, the four spatio-temporal features $X_{SimCGAL}, X_{Granger}, X_{Pearson}, X_{Weather}$ obtained in the first three stages are fused, and then after being dimensionally transformed using the linear layer, they are inputted into the BiGRU for the temporal dependence modeling, and, finally, the predictions are outputted through the linear layer. The prediction process is shown in Equations (23)–(28), where Equations (25)–(27) represent the computational formulas for the BiGRU network.

$$X^{Fusion} = X_{SimCGAL} \odot X_{Granger} \odot X_{Pearson} + Train.K \odot X_{Weather} \tag{23}$$

$$X^{Fusion} = \mathrm{Linear}(X^{Fusion}) \tag{24}$$

$$\overrightarrow{h_t} = \mathrm{GRU}(X_t^{Fusion}, \overrightarrow{h_{t-1}})) \tag{25}$$

$$\overleftarrow{h_t} = \mathrm{GRU}(X_t^{Fusion}, \overleftarrow{h_{t-1}})) \tag{26}$$

$$h_t = [\overrightarrow{h_t} | \overleftarrow{h_t}] \tag{27}$$

$$\hat{y}_{pre} = \mathrm{Linear}(h_t) \tag{28}$$

where X_{Fusion} is a complex feature that fuses several spatio-temporal features, $Train.K$ is a trainable parameter, *Linear* stands for the linear layer, and $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ stand for the forward GRU and backward GRU, respectively. "|" represents the splicing operation, and $\hat{y}_{pre}$ represents the prediction result.

4.5. MSA-GCN Model Training Process

$$\mathrm{Loss} = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{29}$$

The MSA-GCN model operation algorithm flow is shown in Algorithm 1. In stage 1, the base information (distance, connectivity, and eigenvalues to be predicted) is first encoded using the GCN to obtain the base feature $X_{Encoder}$. Then, the similarity adjacency matrix is solved in the CGAL module and the first randomness filtering and key feature extraction are performed using the GCN combined with the $X_{Encoder}$ and CGAL network to obtain the corresponding feature $X_{SimCGAL}$. In the CSC module, the causal adjacency matrix $A_{Granger}$ obtained by the Granger causality test is used to combine $X_{Encoder}$ to obtain the feature $X_{Granger}$ with a causal relationship for the subsequent second chance filtering.

Algorithm 1 MSA-GCN

Input: X_{Flow}: Traffic flow features; X_{Speed}: Traffic speed features; $A_{connection}$: Connection adjacency; features; $A_{Distance}$: Distance adjacency; $WeatherInformation$.
Output: $\hat{y}_{pre}$

Basic Information-GCN Code

$X_{Encoder} \leftarrow \mathrm{GCN}(A_{Connection}$ or $A_{Distance}, X_{Flow}$ or $X_{Speed})$ **//The underlying adjacency matrix and the traffic features to be predicted are encoded as features containing spatial information by means of graph convolution**.

Stage1.1 CGAL module: ConvGRU Attention Layer

$A_{Sim} \leftarrow \frac{1}{1+\mathrm{DTW}(X_{Flow}\ or\ X_{Speed})}$ **//Solve for the magnitude of feature similarity between road nodes**.

$X_{Sim} \leftarrow \mathrm{GCN}(A_{Sim}, X_{Encoder})$ **//Similarity feature coding**.

$[Q, K, V] \leftarrow [\mathrm{Conv1d}(X_{Sim}, W_q), \mathrm{Conv1d}(X_{Sim}, W_k), \mathrm{GRU}(X_{Sim}, W_v)]$

$X_{SimCGAL} \leftarrow \mathrm{Softmax}(\frac{Q \cdot K^T}{\sqrt{d^{\frac{1}{2}}}}) \cdot V$ **//Randomness is filtered for the first time**.

Stage1.2 CSC module: Causal Similarity Capture Module

$A_{Granger} \leftarrow \mathrm{Granger}(X^i, X^j)$ **//Calculate the causality matrix**.

$X_{Granger} \leftarrow \mathrm{GCN}(A_{Granger}, X_{Encoder})$ **//causality feature coding, subsequent to the prediction result fusing it with X_{Sim} for a second randomness interference filtering**.

Stage2 CC module: Correlation Completion Module

$A_{Pearson} \leftarrow \mathrm{Pearson}(X_{Flow}$ or $X_{Speed})$ **//Calculate the Pearson correlation neighbourhood matrix**.

$X_{Pearson} \leftarrow \mathrm{GCN}(A_{Pearson}, X_{Encoder})$ **//Correlation feature coding**.

Stage3 WFC module: Weather Factor Capture
Auto-LRU module: Pre-training

$[X_{MaxT}, X_{MinT}, X_{Wind}, X_{WeatherType}] \leftarrow$ DataMapping[WeatherInformation]

$X_{Fusion} \leftarrow \mathrm{Mean}[\mathrm{Normalization}(X_{MaxT}, X_{MinT}, X_{Wind}, X_{WeatherType})]$

$X_{Encoder}^{Weather} \leftarrow F_{af}(\mathrm{MLP}(X_{Fusion}))$

$X_{Decoder}^{Weather} \leftarrow \mathrm{GRU}(X_{Encoder}^{Weather})$

$X_{OtherLayer} \leftarrow F_{af}(\mathrm{MLP}(X_{Decoder}^{Weather}))$

Cosine similarity calculation

$X_{new}^{Weather} \leftarrow$ Auto-LRU.$Decoder_{Output}$

$\cos(\theta) \leftarrow \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$

$A_{Weather} \leftarrow \cos(\theta)(X_{new}^{Weather})$

$X_{Weather} \leftarrow \mathrm{GCN}(A_{Weather}, X_{Encoder})$

$X_{Weather} \leftarrow \mathrm{CNN}(X_{Weather})$**//Weather feature coding**.

Stage4 Prediction module

$X^{Fusion} \leftarrow X_{SimCGAL} \odot X_{Granger} \odot X_{Pearson} + Train.K \odot X_{Weather}$

$X^{Fusion} \leftarrow \mathrm{Linear}(X^{Fusion})$

$\overrightarrow{h_t} \leftarrow \mathrm{GRU}(X_t^{Fusion}, \overrightarrow{h_{t-1}}))$

$\overleftarrow{h_t} \leftarrow \mathrm{GRU}(X_t^{Fusion}, \overleftarrow{h_{t-1}}))$

$h_t \leftarrow [\overrightarrow{h_t} | \overleftarrow{h_t}]$

$\hat{y}_{pre} \leftarrow \mathrm{Linear}(h_t)$

In stage 2, the CC module solves for potential correlations between nodes in the road network. The correlation adjacency matrix $A_{Pearson}$ is firstly derived and encoded using the GCN and $X_{Encoder}$ to obtain the correlation feature $X_{Pearson}$.

In stage 3, the WFC module models the weather factors affecting people's travel. The weather factors are fused and trained by four different weather factors using the Auto-LRU, a pre-trained network with unsupervised training, and the data are subjected to a down-sampling and up-sampling process to extract the key features. At the end of the training, the weather correlation adjacency matrix $A_{Weather}$ is derived based on cosine similarity using the newly generated features $X_{new}^{Weather}$ and encoded using the GCN joined with the $X_{Encoder}$ to obtain the output $X_{Weather}$ of the WFC module.

In stage 4, the prediction stage, several obtained features are fused and predicted using the BiGRU, and the prediction results are output through the fully connected layer.

The loss function used during training is shown in Equation (29), where y is the true value, and $\hat{y}$ is the model predicted value.

5. Experiments

5.1. Datasets

This paper used three real-world datasets, PEMSD8, LOSLOOP, and SZAREA, covering both traffic speed and traffic flow data types for evaluating the high and low performance of the model. The parameters of the different datasets are shown in Table 2. The adjacency matrix type is the graph structure that comes with the dataset, consisting of a distance adjacency matrix representing the distance between nodes and a connectivity adjacency matrix labeled with 0 and 1 representing whether two nodes are connected. The adjacency matrix used by $X_{Encoder}$ is based on the adjacency matrix provided by the dataset, both of which are adjacency matrices that can reflect the structure of the road network.

Table 2. Description of dataset parameters.

Dataset	PEMSD8	SZAREA	LOSLOOP
Data Type	Flow	Speed	Speed
Adjacency Matrix Type	Distance Adjacency Matrix	Connectivity Adjacency Matrix	Distance Adjacency Matrix
Location	San Bernardino	Shenzhen	Los Angeles
Country	America	China	America
Geographic Prediction Type	Road	Area	Road
Sensors	170	78	207
Time Steps	17,856	480	2016
Time Interval	5 min	15 min	5 min
Time Span	1 July–31 August 2016	1–24 January 2019	1–7 March 2012
Weather	Yes	No	No

(1) Traffic flow datasets

- PEMSD8: The dataset is derived from traffic data collected by 170 sensors of the Caltrans Performance Evaluation System on 8 selected roadways in San Bernadino. Data count intervals were 5 min intervals. The data acquisition period was from 1 July 2016 to 31 August 2016.

(2) Traffic speed datasets

- LOSLOOP: This dataset is derived from 207 sensors of the Caltrans Performance Evaluation System on Los Angeles County freeways. Traffic speeds were counted every 5 min, and the data counts cover the period from 1 to 7 March 2012.
- SZAREA: The dataset is derived from cab GPS data in Shenzhen, China, collected from a total of about 30,000 cabs, dividing the urban area of Shenzhen into 78 zones. The data are counted every 15 min. This paper uses the published data section for 21–24 January 2019.

To verify the performance of the MSA-GCN model, the traffic features predicted by the MSA-GCN include flow (PEMSD8) and speed (LOSLOOP, SZAREA). To test whether the MSA-GCN can handle features at the regional level, SZAREA is a regional-level dataset.

Additionally, the PEMSD8 dataset includes weather data, which are used in this paper to capture weather features. The corresponding hyperparameter settings for each dataset are shown in Table 3, where HistoryLength represents the length of the history time slice used for training, HiddenDim represents the dimension of the hidden layer, and $\lambda_{dataset}$ represents the filtering threshold for the Pearson's correlation coefficient. As shown in Figure 3, we determined a threshold range [0.0–0.6] in advance, tested the performance comparison of each dataset under different thresholds and the same prediction benchmark, and determined the final threshold for filtering the Pearson correlation matrix. This helped us to determine the final threshold for filtering the Pearson correlation matrix. The relevant parameter settings for the Auto-LRU are shown in Table 4.

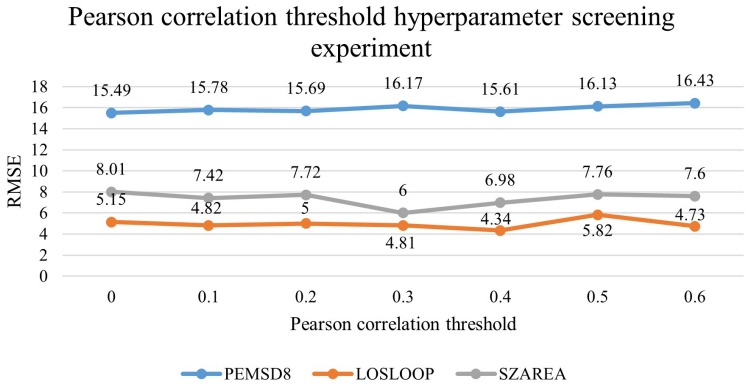


Figure 3. Pearson threshold screening experiment.

Table 3. MSA-GCN model hyperparameter settings for different datasets.

Dataset	PEMSD8	SZAREA	LOSLOOP
Learning Rate	0.001	0.001	0.001
Epoch	20	300	200
HistoryLength	6	6	6
BatchSize	256	64	128
HiddenDim	64	32	256
Split Ratio	0.92	0.9	0.8
$\lambda_{dataset}$	0	0.4	0.3

Table 4. Auto-LRU model hyperparameter settings for PEMSD8.

Dataset	Learning Rate	Epoch	BatchSize	Optimizer	Loss
Value	0.001	50	2048	Adam	MSE

5.2. Methods of Comparison

5.2.1. Complex Deep Learning Models

- SMGCN [69]: SMGCN achieves traffic prediction by extracting the spatio-temporal correlations of fine- and coarse-grained traffic maps and using cross-scale fusion techniques to enable the transfer of information between different granularities.
- STFGCN [70]: STFGCN is a traffic flow prediction method that improves prediction performance by combining node-specific graph convolution operations, adaptive adjacency matrices, and a Transformer-based global temporal correlation learning module.
- DeFedSTTN [71]: DeFedSTTN is a spatial-temporal transformer network model in a decentralized federated learning framework.
- UAT-GCN [72]: UAT-GCN is a novel deep learning method for traffic speed prediction that captures both the spatial and temporal dependencies of traffic data and quantifies

prediction uncertainty and improves prediction accuracy and reliability through the combination of graph convolutional networks and gated cyclic units.

- DST-GraphSAGE [73]: DST-GraphSAGE is an innovative deep learning framework that combines GraphSAGE models and attention mechanisms to dynamically capture spatial and temporal dependencies in traffic networks.
- AGFCRN [74]: AGFCRN is an adaptive graph fusion convolutional recurrent network for traffic flow prediction.
- AST-InceptionNet [75]: AST-InceptionNet is a multi-scale adaptive spatio-temporal graph neural network model.
- DDGCRN [76]: DDGCRN is a Decomposition Dynamic Graph Convolutional Recurrent Network for traffic prediction.
- GDGCN [77]: GDGCN is a General Dynamic Graph Convolutional Network for traffic flow prediction.
- MSSTAT [78]: MSSTAT is a multi-scale spatial time-aware Transformer architecture for improving the accuracy of traffic prediction through multi-scale feature extraction and spatial heterogeneity modeling.
- Traffic-GGNN [55]: Traffic-GGNN is a novel attention-mechanism-enhanced spatial-temporal gated graph neural network for traffic flow prediction.
- Tms-GCN [63]: TmS-GCN is a regional-level spatio-temporal graph convolutional prediction model which mainly consists of a GCN and GRU. It is a GPS-based regional-level traffic flow prediction model which had good results in predicting the traffic flow in 78 zones in Shenzhen by zip code using the SZAREA dataset.
- ASTGCN [61]: ASTGCN is a spatio-temporal graph convolutional network model based on an attention mechanism which effectively captures the dynamic spatio-temporal features of traffic data by fusing the spatial-temporal attention mechanism and spatio-temporal convolutional operations.
- T-GCN [58]: T-GCN is a new temporal graph convolutional network model combining graph convolutional networks (GCNs) and gated recurrent units (GRUs).
- STGCN [79]: It is used to predict the traffic flow, and integrates a graph convolutional network and Gated Linear Unit (GLU) through a spatio-temporal convolution module for modeling and obtains good prediction results.
- GRU: The GRU predictive traffic flow model achieves accurate predictions of future traffic flows by using gate mechanisms (update gates and reset gates) to capture long- and short-term dependencies in the traffic flow time series.
- GCN: The GCN predictive traffic flow model captures spatial dependencies by applying convolution operations to nodes (e.g., road segments, intersections) of the traffic network graph, aggregating information from neighboring nodes and combining it with data in the time dimension for traffic flow prediction.

5.2.2. Traditional Models

- ARIMA: It is a combination of an autoregressive (AR) model and moving average (MA) model. It is an extended form of the AR model which introduces different operations and moving average terms based on the AR model.
- SVR: This model maps historical traffic flow data into a high-dimensional space and subsequently builds a regression model from the eigenvalues in the space.
- HA: This model is based on using the average of historical data to predict future values. In the field of traffic flow, the model first obtains a weighted average of the collected historical traffic data, calculates the average traffic characteristics of different time slices, and then makes a prediction of the future traffic flow based on the traffic characteristics of each time slice.

5.3. Evaluation Indicators

This paper uses the following three evaluation metrics to assess the performance of the model: the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and

Mean Absolute Percentage Error (MAPE). When their values are smaller, it means that the performance of the model is better and vice versa.

$$RMSE = \left[\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\right]^{\frac{1}{2}} \tag{30}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{31}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| * 100\% \tag{32}$$

where N is the number of nodes, y_i is the true value of node i, and $\hat{y}_i$ is the predicted value of node i by the model.

5.4. Comparative Experiment

As shown in Tables 5–7, and Figure 4, overall, the MSA-GCN achieves a good prediction performance improvement compared to the state-of-the-art baseline on all three datasets. As an example, the percentage of RMSE improvement is 29.33% compared to the baseline on the PEMSD8 dataset, 27.03% compared to the baseline on the LOSLOOP dataset, and 23.07% compared to the baseline on the SZAREA dataset. The MSA-GCN is based on multiple potential spatio-temporal correlation mining, and it solves a contingency problem in the prediction data themselves and the DTW algorithm. Secondary filtering is performed. On the SZAREA dataset, compared with some models that do not use graph structure, e.g., ARIMA, HA, GRU, etc., the MSA-GCN achieves good results, with a comparative improvement of 40.35%, 36.50%, and 35.76%, respectively. Compared to pure time series data, traffic flow data are also largely affected by the structure of the road network, which reflects the importance of modeling the graph structure. Among them, the ARIMA model achieves poor prediction results for the SZAREA dataset due to the requirement for smoothness of the data, without which poor prediction occurs. In the case of the traffic network, in particular, the speed feature changes with the vehicles on the road. Its eigenvalues change a lot, so the ARIMA model cannot be well applied to this scenario for prediction. In addition, the GCN shows a strong performance in information aggregation, especially in aggregating the information of neighboring nodes, which achieves greater results. However, compared with the time series model, the traffic flow data have both significant temporal and spatial characteristics, and the prediction performance of the GCN, which has a weaker ability to model the sequence information, is poorer, and the MSA-GCN improves the performance by 42.85%. For the SZAREA dataset, the T-GCN and the TmS-GCN combine both the GCN and the GRU, which means that the models have temporal and spatial capabilities, and they achieve the best performance compared with the traditional models.

Compared to models like STGCN, TGCN, ASTGCN, and GraphWaveNet, which only consider basic graph structures, the MSA-GCN takes a more comprehensive approach by considering multiple potential spatio-temporal correlations from the perspectives of similarity, causality, and correlation. As a result, the prediction performance of the MSA-GCN for the LOSLOOP dataset improved by 30.1% compared to the T-GCN. Similarly, the classic methods ASTGCN and GraphWaveNet only use road infrastructure as the graph structure, failing to leverage more potential graph structures to explore deeper latent spatio-temporal correlations. This limits their ability to assign accurate weights to the relationships between nodes during training. Both the MSA-GCN and STGCN use a GCN to model spatial correlations. However, for temporal dependency modeling, the STGCN uses one-dimensional convolution and a Gated Linear Unit (GLU). A GRU is better at handling long time sequences compared to a one-dimensional convolution and GLU, especially under the premise of using historical feature sequences of the same length, and processes each time step's input incrementally, offering more flexibility than the one-dimensional convolution and GLU, which are better suited for handling local

features. Therefore, the MSA-GCN outperforms STGCN on the LOSLOOP dataset by 30.1%. In addition, incorporating weather features into the prediction process of the model enables the MSA-GCN to better simulate real-world road conditions, enhancing the model's interpretability. Through comparative experiments and subsequent ablation studies, it could be observed that, with the inclusion of weather features, the performance of the model on the PESMD8 dataset was further improved.

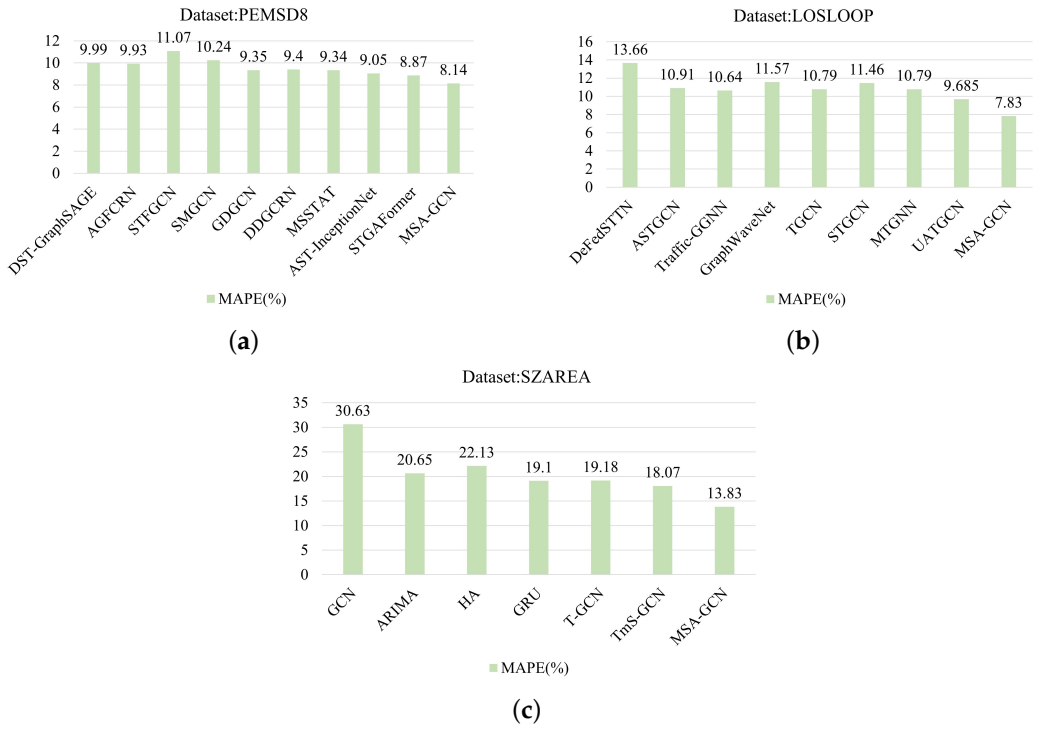


Figure 4. Comparisonof average prediction performance over 12 time steps. (**a**) PEMSD8, (**b**) LOSLOOP, (**c**) SZAREA.

Table 5. Experimental results on the PEMSD8 dataset (flow).

Dataset	**PEMSD8**		
Model	**RMSE**	**MAE**	**MAPE (%)**
DST-GraphSAGE	24.87	15.27	9.99
AGFCRN	24.63	15.52	9.93
STFGCN	24.35	15.23	11.07
SMGCN	24.30	15.44	10.24
GDGCN	23.87	14.82	9.35
DDGCRN	23.75	14.40	9.40
MSSTAT	23.44	14.03	9.34
AST-InceptionNet	23.34	13.72	9.05
STGAFormer	22.43	13.06	8.87
MSA-GCN	**15.85**	**10.76**	**8.14**
Improvement	**29.33%**	**17.61%**	**8.22%**

This paper uses an improved self-attention mechanism called the ConvGRU Attention Layer which performs the first random optimization of similarity features and then applies a Granger causality test for a second round of random optimization. Compared to the distance space attention module proposed by STGAFormer, our MSA-GCN captures and optimizes more aspects of potential graph structures, specifically reflected in the first and second phases, where it captures similarity and correlation and optimizes the impact of randomness twice in the first phase. As a result, our MSA-GCN improves prediction

performance on the PEMSD8 dataset by 29.33% compared STGAFormer. STFGCN and SMGCN also use the DTW algorithm to construct similarity for predicting traffic features, aiming to overcome distance and adjacent connection limitations; however, they do not further optimize the obtained similarity features, indicating that their model performance could be enhanced. The GDGCN focuses on the dynamic evolution of temporal features through an improved GCN while neglecting the diversity of potential spatial information, making the MSA-GCN superior in performance. DST-GraphSAGE emphasizes simultaneously extracting spatial and temporal correlations but does not focus on mining potential spatio-temporal correlations. In contrast, the MSA-GCN, which utilizes a BiGRU to extract temporal dependencies, places greater emphasis on the diversity and optimization of spatial correlations, thus outperforming DST-GraphSAGE in terms of graph structure diversity and optimization. The DeFedSTTN model employs attention mechanisms to capture temporal dependencies, but the adjustment of attention weights may lead to training instability, especially when the unprocessed data contain significant noise, potentially affecting the model's predictive performance. In contrast, the MSA-GCN adopts a multistage, modular approach to optimize multiple graph structures, combining features after two rounds of denoising (randomness) through the CGAL module and Granger causality testing with global correlations, ultimately inputting them into the BiGRU for temporal dependency modeling and outputting prediction results. Therefore, the MSA-GCN considers the problem more deeply and comprehensively, resulting in a prediction performance improvement of 40.66% compared to the DeFedSTTN model.

Table 6. Experimental results on the LOSLOOP dataset (speed).

Dataset	**LOSLOOP**		
Model	**RMSE**	**MAE**	**MAPE (%)**
DeFedSTTN	7.55	3.02	13.66
ASTGCN	6.80	4.05	10.91
Traffic-GGNN	6.67	4.02	10.64
GraphWaveNet	6.45	3.75	11.57
T-GCN	6.41	3.95	10.79
STGCN	6.41	3.71	11.46
MTGNN	6.33	4.21	10.79
UATGCN	6.14	3.62	9.68
MSA-GCN	**4.48**	**3.09**	**7.83**
Improvement	**27.03%**	**14.64%**	**19.15%**

Table 7. Experimental results on the SZAREA dataset (speed).

Dataset	**SZAREA**		
Model	**RMSE**	**MAE**	**MAPE**
GCN	10.50	6.55	30.63
ARIMA	10.06	6.00	20.65
HA	9.45	4.94	22.13
GRU	9.34	4.71	19.10
T-GCN	7.98	4.43	19.18
TmS-GCN	7.80	4.26	18.07
MSA-GCN	**6.00**	**3.31**	**13.83**
Improvement	**23.07%**	**22.30%**	**23.46%**

As shown in Table 8, this paper tests the short-term and long-term prediction performance of the MSA-GCN on three datasets with prediction baselines of 15 min, 30 min, 45 min, and 60 min. It can be seen that the short-term and long-term prediction performance is good, and is also ahead of the compared baselines, and the RMSE floats are controlled to be less than 1, which proves the stability of the model in predicting the data. The RMSE is controlled within 1, which proves the stability of the model prediction data.

Table 8. Long- and short-term forecasting performance.

Dataset		PEMSD8, LOSLOOP, SZAREA			
Models	**Metrics**	**15 min**	**30 min**	**45 min**	**60 min**
PEMSD8	RMSE	15.46	15.57	16.02	16.33
	MAE	10.44	10.56	10.89	11.15
	MAPE	7.88	8.01	8.05	8.62
LOSLOOP	RMSE	4.38	4.57	4.46	4.53
	MAE	3.00	3.15	3.10	3.14
	MAPE	7.59	7.93	7.95	7.86
SZAREA	RMSE	6.00	6.02	5.98	6.00
	MAE	3.30	3.30	3.31	3.33
	MAPE	14.03	13.7	14.01	13.60

5.5. Ablation Experiments and Model Performance Analysis

In this section, the setup of the ablation experiments, the results, and the model performance (number of parameters, training time) will be presented and analyzed.

5.5.1. Experimental Setup

Two types of ablation experiments were performed on the MSA-GCN: (1) experiments using only one type of feature, and (2) experiments ablating one specific feature. These experiments are designed to discuss the roles and interconnections of each module.

- Only X_{Sim}: This ablation experiment only retains as spatio-temporal features the prediction of the $X_{Encoder}$ obtained after encoding the features using the underlying information X_{Sim} and A_{Sim} jointly with the feature extraction using the GCN, with the aim of testing the performance of the model without any stochasticity filtering. The CGAL module, CSC module, CC module, and WFC module are not included.
- Only $X_{SimCGAL}$: This ablation experiment retains only the results of the first filtering of contingency using the CGAL module, i.e., the similarity adjacency matrix A_{Sim} and the GCN-encoded base features $X_{Encoder}$ modeled after feature extraction by the CGAL module $X_{SimCGAL}$ are used for prediction. The performance of feature prediction after only one chance filtering is examined, without the potential relevant complement of the CC module, and without considering the influence of the second randomness filtering of the CSC module and the external factors of the WFC module.
- Only $X_{Granger}$: This ablation experiment retains only the causal feature $X_{Granger}$ for second chance filtering, and examines the prediction performance when using only this feature from the perspective of joint multi-graph prediction, disregarding the first filtering of chance by the CGAL module, disregarding the CC module and the WFC module, and filtering the coding feature $X_{Encoder}$ by chance only once.
- Only $X_{SimCGAL} \odot X_{Granger}$: This ablation experiment retains only the spatio-temporal features filtered by chance twice. The cases of potential correlation–completion performance of the CC module and the influence of external weather factors of the WFC module are not taken into account.
- Only $X_{Pearson}$: This ablation experiment only considers the $X_{Pearson}$ features obtained by solving for the Pearson correlation in the CC module and encoding them to test the impact of a small number of potential spatio-temporal feature preconditions on the model prediction performance from the perspective of joint prediction of multiple graphs. The CGAL module, CSC module, and WFC module are not considered.
- Only $X_{Weather}$: This ablation experiment only considers the model performance of the MSA-GCN when the weather feature $X_{Weather}$ is used as a spatio-temporal feature.
- w/o $X_{Weather}$: This ablation experiment removes the weather features from the third stage to explore the impact on MSA-GCN when external weather features are not simulated.

- w/o $X_{Pearson}$: This ablation experiment removes the global correlation features from the second stage to explore the impact on the MSA-GCN when this potential spatial correlation is absent.
- w/o $X_{Granger}$: This ablation experiment removes the second module, the CSC module, from the first stage to explore the impact on the MSA-GCN when the second round of stochastic filtering is not performed.
- w/o $X_{SimCGAL}$: This ablation experiment removes the first module, the CGAL module, from the first stage to explore the impact on the MSA-GCN when only features from the Granger causality test are used in combination with features from the other two stages.

5.5.2. Results of the Multi-Temporal Feature Ablation Experiment

The results of the ablation experiments on different datasets are shown in Tables 9–11. For the two considerations of the ablation experimental setup, the following analyses are made in conjunction with the experimental results.

Table 9. PEMSD8: Ablation experiment result.

Dataset		PEMSD8			
Type of Ablation	**Metrics**	**15 min**	**30 min**	**45 min**	**60 min**
Only X_{Sim}	RMSE	47.19	46.96	47.02	47.62
	MAE	33.16	32.93	33.00	33.47
	MAPE	24.85	24.93	24.83	25.24
Only $X_{SimCGAL}$	RMSE	19.13	19.32	18.32	18.89
	MAE	13.55	13.86	12.99	13.41
	MAPE	9.18	9.05	8.84	9.24
Only $X_{SimCGAL} \odot X_{Granger}$	RMSE	17.60	18.03	17.15	18.71
	MAE	12.54	12.50	11.71	13.25
	MAPE	8.67	9.08	8.80	9.65
Only $X_{Granger}$	RMSE	63.09	62.66	64.22	63.03
	MAE	45.54	44.96	45.87	45.36
	MAPE	35.58	35.63	37.14	35.01
Only $X_{Pearson}$	RMSE	43.83	43.69	43.97	44.05
	MAE	30.48	30.68	30.59	30.86
	MAPE	22.82	22.35	24.19	24.07
Only $X_{Weather}$	RMSE	62.55	61.76	62.94	62.17
	MAE	43.91	43.54	44.30	44.05
	MAPE	32.39	32.22	35.31	33.03
w/o $X_{Weather}$	RMSE	17.08	16.15	20.78	17.29
	MAE	11.87	11.02	14.99	11.78
	MAPE	8.47	8.53	9.60	8.68
w/o $X_{Pearson}$	RMSE	16.38	17.27	16.56	19.42
	MAE	11.23	12.02	11.31	13.93
	MAPE	7.98	8.16	8.27	9.76
w/o $X_{Granger}$	RMSE	19.03	16.11	16.29	16.42
	MAE	13.60	10.96	11.08	11.28
	MAPE	8.61	8.06	8.38	8.59
w/o $X_{SimCGAL}$	RMSE	58.88	57.98	57.29	58.91
	MAE	41.35	40.89	40.64	41.45
	MAPE	34.54	33.45	33.22	34.04
MSA-GCN	RMSE	**15.46**	**15.57**	**16.02**	**16.33**
	MAE	**10.44**	**10.56**	**10.89**	**11.15**
	MAPE	**7.88**	**8.01**	**8.05**	**8.62**

Table 10. LOSLOOP: Ablation experiment result.

Dataset		LOSLOOP			
Type of Ablation	**Metrics**	**15 min**	**30 min**	**45 min**	**60 min**
Only X_{Sim}	RMSE	4.77	5.34	5.02	4.85
	MAE	3.38	3.91	3.64	3.37
	MAPE	7.91	9.96	8.92	8.67
Only $X_{SimCGAL}$	RMSE	4.58	4.71	5.00	4.81
	MAE	3.23	3.20	3.44	3.31
	MAPE	8.00	8.18	8.97	8.35
Only $X_{SimCGAL} \odot X_{Granger}$	RMSE	4.69	4.77	4.58	4.59
	MAE	3.34	3.32	3.21	3.21
	MAPE	8.10	8.66	8.04	7.98
Only $X_{Granger}$	RMSE	7.18	7.60	7.24	7.51
	MAE	4.64	4.94	4.66	4.86
	MAPE	12.47	14.07	12.54	13.78
Only $X_{Pearson}$	RMSE	7.12	6.46	6.65	7.05
	MAE	4.98	4.35	4.36	4.67
	MAPE	12.99	11.70	11.51	12.30
w/o $X_{Pearson}$	RMSE	4.69	4.77	4.58	4.59
	MAE	3.34	3.32	3.21	3.21
	MAPE	8.10	8.66	8.04	7.98
w/o $X_{Granger}$	RMSE	4.56	4.60	4.56	4.52
	MAE	3.25	3.26	3.15	3.16
	MAPE	7.99	7.98	7.81	7.86
w/o $X_{SimCGAL}$	RMSE	6.56	6.58	6.45	6.47
	MAE	4.35	4.33	4.22	4.11
	MAPE	11.63	11.41	11.19	11.37
MSA-GCN	RMSE	**4.38**	**4.57**	**4.46**	**4.53**
	MAE	**3.00**	**3.15**	**3.10**	**3.14**
	MAPE	**7.59**	**7.93**	**7.95**	**7.86**

In Figure 5, this paper visualizes the RMSE values for each time step of each dataset. Combined with various types of ablation experiments, it can be seen that the model has good stability with small changes in error for both short-term and long-term prediction, and has strong long-term prediction ability. Where * represents Hadamard product.

First, using the PEMSD8 dataset as an example, we explored the results of the MSA-GCN ablation experiments when only one type of feature was used. By examining $OnlyX_{Sim}$, $OnlyX_{Granger}$, $OnlyX_{Pearson}$, and $OnlyX_{Weather}$, it became evident that using a single feature without any further processing significantly degrades model performance. This is because the core characteristic of the GCN is updating the information of a node by aggregating the importance of its neighbors. These graph structures, when unoptimized, especially $X_{Granger}$, lead to a 308.08% performance drop compared to the full MSA-GCN model. The reason is that the Granger causality matrix contains a large number of elements with a value of 1, indicating that many nodes can be used as mutually predictive sequences. In the second module of the first stage, the CSC module, the number of node pairs without a causal relationship that is filtered out is not very large. These elements with a value of 1 negatively affect the information aggregation in the GCN, causing each node to treat all its neighbors as equally important and accept all their information indiscriminately, which overly smooths the process and introduces a lot of noise, resulting in poor outcomes. Similarly, for $X_{Weather}$, the matrix has many values close to 1, and using it alone without further processing leads to similar problems. On the other hand, X_{Sim} and $X_{Pearson}$ perform better than $X_{Granger}$ and $X_{Weather}$ when used individually. This is because the values of similarity and correlation, calculated through the DTW algorithm and Pearson correlation algorithm, reflect the underlying spatial correlations to some extent. Moreover, these values range between 0 and 1, making them more accurate, thus yielding better results. After excluding the weather factor, we separately tested Only $X_{SimCGAL}$ and

Only $X_{SimCGAL} \odot X_{Granger}$ as inputs to the MSA-GCN. Comparing their results with X_{Sim}, we can see that each step of stochastic optimization significantly improves the prediction performance of the MSA-GCN. When the CGAL module in the first stage performs the first round of filtering for randomness, $X_{SimCGAL}$ achieves a 59.46% improvement over X_{Sim}. In the second round of filtering for randomness, performed by the CSC module in the first stage, $X_{SimCGAL} \odot X_{Granger}$ shows a 62.70% improvement over X_{Sim}. The above analysis provides insights into the results of the ablation experiments for individual graph structures and examines whether the two modules in the first stage—CGAL and CSC—enhance the prediction performance of the MSA-GCN by filtering randomness, especially in the absence of the third-stage weather factor.

Table 11. SZAREA: Ablation experiment result.

Dataset		**SZAREA**			
Type of Ablation	**Metrics**	**15 min**	**30 min**	**45 min**	**60 min**
Only X_{Sim}	RMSE	7.90	7.85	7.85	7.81
	MAE	4.47	4.44	4.29	4.31
	MAPE	18.75	18.55	17.97	17.78
Only $X_{SimCGAL}$	RMSE	6.22	6.26	6.11	6.19
	MAE	3.53	3.51	3.36	3.46
	MAPE	14.75	14.60	13.36	13.82
Only $X_{SimCGAL} \odot X_{Granger}$	RMSE	6.07	6.09	6.01	6.05
	MAE	3.37	3.38	3.32	3.34
	MAPE	14.08	14.26	14.11	13.58
Only $X_{Granger}$	RMSE	9.76	9.29	8.66	8.74
	MAE	5.77	5.26	4.94	5.00
	MAPE	23.32	22.25	20.19	20.75
Only $X_{Pearson}$	RMSE	6.56	6.39	6.49	6.44
	MAE	3.89	3.83	3.80	3.77
	MAPE	16.24	15.74	15.11	15.60
w/o $X_{Pearson}$	RMSE	6.07	6.09	6.01	6.05
	MAE	3.37	3.38	3.32	3.34
	MAPE	14.08	14.26	14.11	13.58
w/o $X_{Granger}$	RMSE	7.82	7.65	7.91	7.69
	MAE	4.40	4.19	4.36	4.26
	MAPE	18.56	17.39	17.40	17.36
w/o $X_{SimCGAL}$	RMSE	7.74	7.90	7.93	7.85
	MAE	4.45	4.58	4.56	4.46
	MAPE	18.83	18.84	19.12	19.33
MSA-GCN	RMSE	**6.00**	**6.02**	**6.00**	**6.00**
	MAE	**3.30**	**3.30**	**3.33**	**3.33**
	MAPE	**14.03**	**13.7**	**14.01**	**13.60**

Next, using the PEMSD8 dataset as an example, we discuss the phase-wise optimization of the MSA-GCN and the superiority of its multi-graph structure when a single feature is removed. When we remove the simulation of spatio-temporal features in the third stage, specifically, the $X_{Weather}$ feature, the model's performance drops by 10.47% compared to the full MSA-GCN model. Reducing the simulation of weather leads to the model's inability to capture real-world road conditions. When we remove the CC module in the second stage, which eliminates the $X_{Pearson}$ feature, the model's performance decreases by 5.95%. The absence of global correlation weakens the GCN's ability to accurately model the relationships between a node and its neighbors. The global correlation matrix describes the degree of correlation between nodes and is calculated based on historical traffic features. When we remove the second round of stochastic filtering in the first stage (i.e., remove the $X_{Granger}$ feature), the lack of this second round of filtering leaves residual random noise in the $X_{SimCGAL}$ from the first stage, resulting in a 23.09% performance drop compared to the full MSA-GCN model. When the CGAL module in the first stage is removed (i.e., the $X_{SimCGAL}$ feature), the model experiences a significant performance decline. This is because

the remaining Granger features use an adjacency matrix containing many elements with a value of 1, which aligns with the conclusion from the previous discussion where we experimented with using only $X_{Granger}$. However, in this case, the performance is better than when using a single feature because the second-stage global correlation feature $X_{Pearson}$ and the third-stage real-world simulation are retained. In summary, this analysis of the phase-wise optimization structure and graph structure ablation experiments of the MSA-GCN model demonstrates that the multi-graph convolution and phase-wise optimization approach of the MSA-GCN is indeed effective.

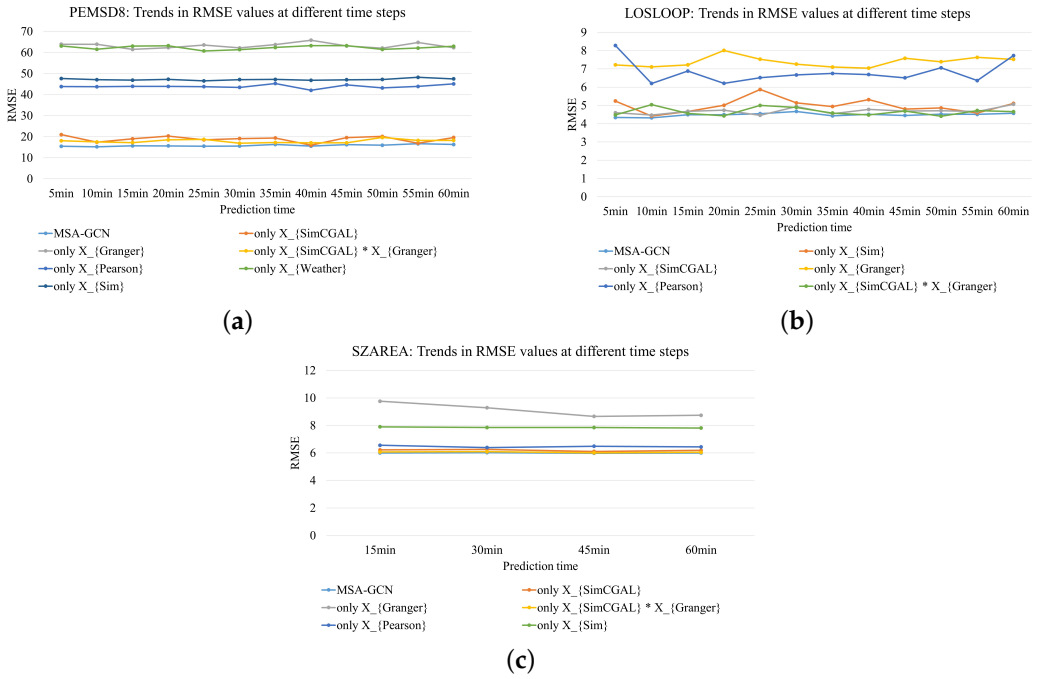


Figure 5. Ablation experiments: specific RMSE values for each time step. (**a**) PEMSD8, (**b**) LOSLOOP, (**c**) SZAREA.

5.5.3. Model Performance Analysis

This section provides statistics on the training time and the number of parameters for each time step corresponding to the MSA-GCN model and its single-graph ablation experiments, as shown in Figure 6. On the PEMSD8 dataset, the MSA-GCN and the rest of the ablation experiments results could start to converge in the fifth epoch of loss, and the time consumed for each time step was 0.31 min. In contrast, the convergence speed of the MSA-GCN was a little slower than the rest of the kinds of convergence speed, but, taking into account of the time consumed and the number of parameter variations, it was in the acceptable range. On the LOSLOOP dataset, the MSA-GCN started to converge at the eighth epoch, the convergence speed of the main body and the structure of the ablation experiments were not very different, and its time-consuming and parameter count variations were also within the acceptable range. The SZAREA dataset as a whole started to converge at the 40th epoch, and the overall situation was similar to that of the LOSLOOP dataset. Of the several ablation experiments, the one that was closer to the performance of the MSA-GCN was the case of Only $X_{SimCGAL} \odot X_{Granger}$, i.e., the case with only two randomness filters. The training time consumed on the three datasets was only reduced by 3.22%, 1.29%, and 9.37% compared to that of the MSA-GCN, respectively, resulting in a 13.8%, 7.00%, and 1.16% performance improvement (RMSE metrics under the 15 min prediction benchmark) for the model.

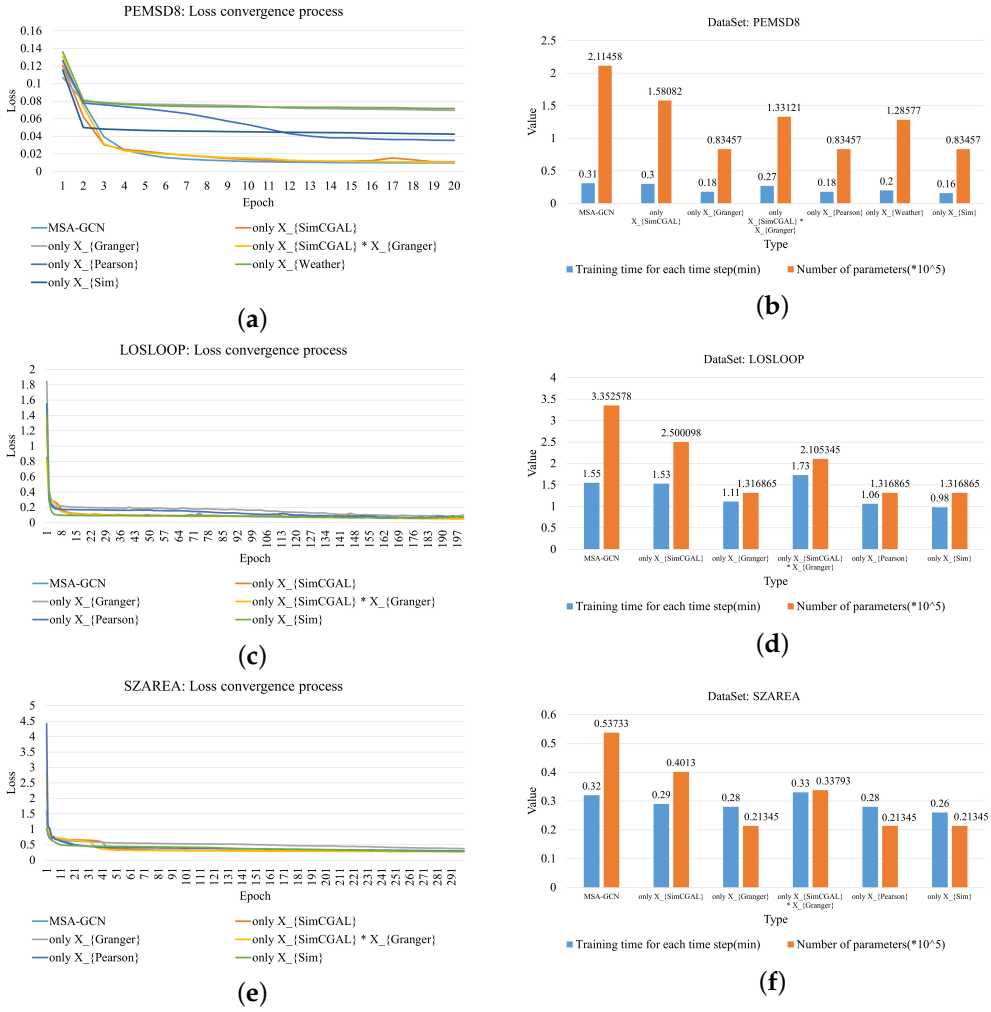


Figure 6. Convergence process of loss values for different datasets, training time consumed, number of parameters statistics. (**a**) PEMSD8—loss convergence process, (**b**) PEMSD8—training time and number of parameters, (**c**) LOSLOOP—loss convergence process, (**d**) LOSLOOP—training time and number of parameters, (**e**) SZAREA—loss convergence process, (**f**) SZAREA—training time and number of parameters.

To examine the prediction of traffic features of different nodes more intuitively, we visualized the real and predicted features of some nodes on the PEMSD8, LOSLOOP, and SZAREA datasets. As Figure 7 shows, we observed that the predicted features generated by the MSA-GCN and the real features matched well, which proves that the MSA-GCN, which captures and optimizes the potential spatio-temporal correlations through multiple stages, can learn the complex potential spatio-temporal correlations in the traffic features and make good predictions. Moreover, for the two different traffic features, flow and speed, both regional and road-level predictions performed well, where PEMSD8 and LOSLOOP are road-level features and SZAREA is a regional-level feature.

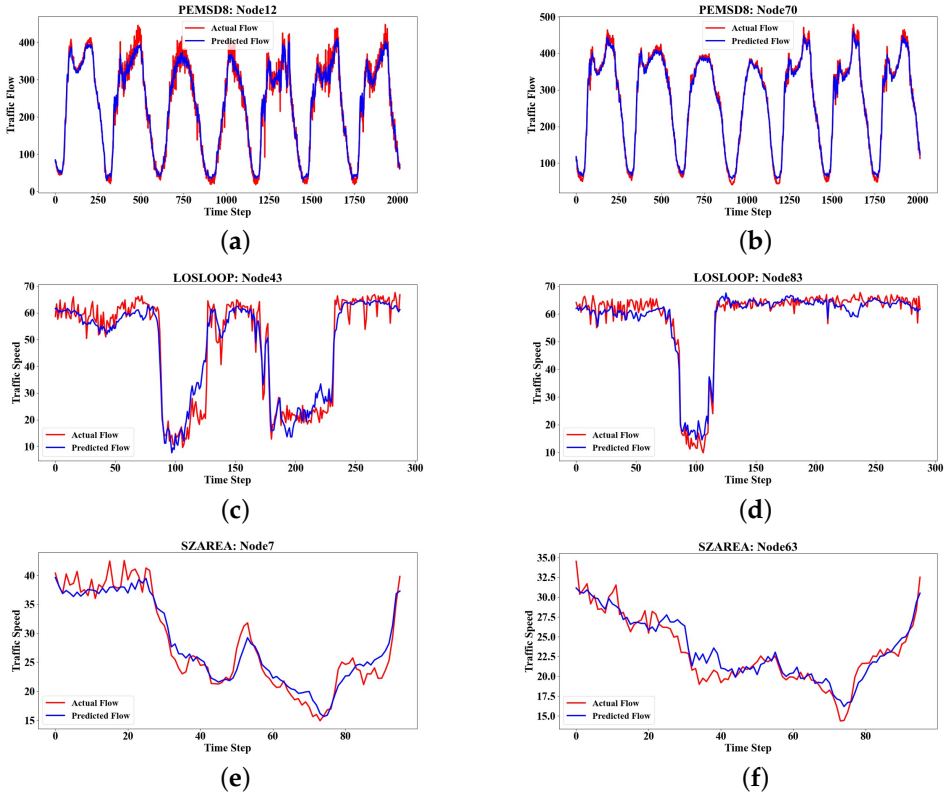


Figure 7. Fitted plots of predicted and true values of some nodes for different datasets. (**a**) Real and predicted features of PEMSD8 dataset node 12, (**b**) real and predicted features of PEMSD8 dataset node 70, (**c**) real and predicted features of LOSLOOP dataset node 43, (**d**) real and predicted features of LOSLOOP dataset node 83, (**e**) real and predicted features of SZAREA dataset node 7, (**f**) real and predicted features of SZAREA dataset node 63.

6. Conclusions

In this paper, the MSA-GCN: Multistage Spatio-Temporal Aggregation Graph Convolutional Networks for Traffic Flow Prediction method is proposed for traffic flow prediction. Specifically, in the data preparation stage, road base features are encoded using a GCN to fuse road distance or connectivity information and their traffic features. Then, different tasks are performed in each module and various spatio-temporal features are extracted in three stages. In the first stage, the features to be predicted and the randomness feature interference problem present in the DTW algorithm are filtered twice by the CGAL and CSC modules proposed in this paper, and similarity features are obtained. In the second stage, the potential spatio-temporal correlations are modeled. In the third stage, several weather features are extracted by the unsupervised pre-training Auto-LRU module proposed in this paper, and weather correlation features affecting people's travel are obtained. In the fourth stage, several previous potential spatio-temporal features are fused, time-dependent modeling is carried out using a BiGRU, and the prediction results are output through the fully connected layer. Our method achieved good prediction results compared to state-of-the-art baseline methods.

Based on the various experimental results, the MSA-GCN demonstrates promising application potential in the field of traffic flow prediction, offering a new approach for research on multi-graph convolution. This approach suggests that, when pursuing diversity in graph structures, a stepwise fusion method can help overcome certain inherent limitations in specific graph structures. Moreover, this model can be applied to stock prediction. Different stocks can be viewed as different nodes, and the relationships between stocks (such as the degree of industry correlation, similarity between stocks, and causal relationships) can be represented by edges. The GCN aggregates information from neighboring

nodes, revealing the mutual influence between different stocks. Additionally, stock price changes are typical time series data, and the use of a GRU, with its gating mechanism, can effectively address short-term memory issues, making it suitable for predicting future stock trends.

Although the proposed MSA-GCN model achieved good performance in experiments, there is still room for improvement. The examination and definition of randomness can be further enriched. Future work should consider more explicit traffic conditions, such as traffic accidents, weather features, and road conditions, as well as implicit traffic conditions, such as the degree of road congestion and the centrality of key traffic hubs. Therefore, discovering various types of graphs and optimizing these structures through multistage methods will be the direction of our future work.

Author Contributions: Conceptualization, Z.S. and J.F.; methodology, Z.S. and J.F.; software, J.F.; validation, Z.S., J.H., and J.F.; investigation, Z.S. and J.F.; formal analysis, J.F.; writing—original draft preparation, J.F.; writing—review and editing, Z.S., J.H., and C.G.; supervision, Z.S., J.H., and C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under grant 62176131.

Data Availability Statement: The dataset used in this article is sourced from publicly available datasets and can be found at the following links: PEMSD8: https://github.com/guoshnBJTU/ASTGCN-2019-pytorch/tree/master. LOSLOOP: https://github.com/Rajnish-bhagat/Traffic-Prediction-TGCN/tree/main. SZAREA: https://github.com/Joker-L0912/Tms-GCN-Py/tree/main (accessed on 28 September 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jiang, W.; Luo, J. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* **2022**, *207*, 117921. [CrossRef]
2. Shaygan, M.; Meese, C.; Li, W.; Zhao, X.G.; Nejad, M. Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities. *Transp. Res. Part Emerg. Technol.* **2022**, *145*, 103921. [CrossRef]
3. Zhang, S.; Guo, Y.; Zhao, P.; Zheng, C.; Chen, X. A Graph-Based Temporal Attention Framework for Multi-Sensor Traffic Flow Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 7743–7758. [CrossRef]
4. Li, Z.; Xiong, G.; Tian, Y.; Lv, Y.; Chen, Y.; Hui, P.; Su, X. A Multi-Stream Feature Fusion Approach for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1456–1466. [CrossRef]
5. Lu, S.; Zhang, Q.; Chen, G.; Seng, D. A combined method for short-term traffic flow prediction based on recurrent neural network. *Alex. Eng. J.* **2021**, *60*, 87–94. [CrossRef]
6. Cools, M.; Moons, E.; Wets, G. Investigating the Variability in Daily Traffic Counts through use of ARIMAX and SARIMAX Models: Assessing the Effect of Holidays on Two Site Locations. *Transp. Res. Rec.* **2009**, *2136*, 57–66. [CrossRef]
7. Rubio, L.; Gutiérrez-Rodríguez, A.J.; Forero, M.G. EBITDA Index Prediction Using Exponential Smoothing and ARIMA Model. *Mathematics* **2021**, *9*, 2538. [CrossRef]
8. Almeida, A.; Brás, S.; Oliveira, I.; Sargento, S. Vehicular traffic flow prediction using deployed traffic counters in a city. *Future Gener. Comput. Syst.* **2022**, *128*, 429–442. [CrossRef]
9. Chen, L.; Zheng, L.; Yang, J.; Xia, D.; Liu, W. Short-term traffic flow prediction: From the perspective of traffic flow decomposition. *Neurocomputing* **2020**, *413*, 444–456. [CrossRef]
10. Griesbach, K.; Beggiato, M.; Hoffmann, K.H. Lane Change Prediction with an Echo State Network and Recurrent Neural Network in the Urban Area. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6473–6479. [CrossRef]
11. Lv, M.; Hong, Z.; Chen, L.; Chen, T.; Zhu, T.; Ji, S. Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3337–3348. [CrossRef]
12. Ma, C.; Zhao, Y.; Dai, G.; Xu, X.; Wong, S.C. A Novel STFSA-CNN-GRU Hybrid Model for Short-Term Traffic Speed Prediction. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3728–3737. [CrossRef]
13. Shu, W.; Cai, K.; Xiong, N.N. A Short-Term Traffic Flow Prediction Model Based on an Improved Gate Recurrent Unit Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16654–16665. [CrossRef]
14. Ma, C.; Dai, G.; Zhou, J. Short-Term Traffic Flow Prediction for Urban Road Sections Based on Time Series Analysis and LSTM-BILSTM Method. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5615–5624. [CrossRef]
15. Dai, F.; Huang, P.; Mo, Q.; Xu, X.; Bilal, M.; Song, H. ST-InNet: Deep Spatio-Temporal Inception Networks for Traffic Flow Prediction in Smart Cities. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 19782–19794. [CrossRef]

16. Zhao, L.; Chen, M.; Du, Y.; Yang, H.; Wang, C. Spatial-temporal graph convolutional gated recurrent network for traffic forecasting. *arXiv* **2022**, arXiv:2210.02737. [CrossRef]
17. Do, L.N.; Vu, H.L.; Vo, B.Q.; Liu, Z.; Phung, D. An effective spatial-temporal attention based neural network for traffic flow prediction. *Transp. Res. Part Emerg. Technol.* **2019**, *108*, 12–28. [CrossRef]
18. Sun, G.; Zhang, F.; Gao, B.; Zhou, Y.; Xiang, Y.; Pan, S. Photonics-based 3D radar imaging with CNN-assisted fast and noise-resistant image construction. *Opt. Express* **2021**, *29*, 19352–19361. [CrossRef]
19. Li, Y.; Liu, D.; Li, H.; Li, L.; Li, Z.; Wu, F. Learning a Convolutional Neural Network for Image Compact-Resolution. *IEEE Trans. Image Process.* **2019**, *28*, 1092–1107. [CrossRef]
20. Zheng, Z.; Yang, Y.; Liu, J.; Dai, H.N.; Zhang, Y. Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3927–3939. [CrossRef]
21. Ke, X.; Shi, L.; Guo, W.; Chen, D. Multi-Dimensional Traffic Congestion Detection Based on Fusion of Visual Features and Convolutional Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2157–2170. [CrossRef]
22. Cheng, Z.; Lu, J.; Zhou, H.; Zhang, Y.; Zhang, L. Short-Term Traffic Flow Prediction: An Integrated Method of Econometrics and Hybrid Deep Learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5231–5244. [CrossRef]
23. Zheng, H.; Lin, F.; Feng, X.; Chen, Y. A Hybrid Deep Learning Model with Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 6910–6920. [CrossRef]
24. Chen, L.; Shi, P.; Li, G.; Qi, T. Traffic flow prediction using multi-view graph convolution and masked attention mechanism. *Comput. Commun.* **2022**, *194*, 446–457. [CrossRef]
25. Chen, Z.; Zhao, B.; Wang, Y.; Duan, Z.; Zhao, X. Multitask Learning and GCN-Based Taxi Demand Prediction for a Traffic Road Network. *Sensors* **2020**, *20*, 3776. [CrossRef]
26. Wang, J.; Chen, Q.; Gong, H. STMAG: A spatial-temporal mixed attention graph-based convolution model for multi-data flow safety prediction. *Inf. Sci.* **2020**, *525*, 16–36. [CrossRef]
27. Zhang, J.; Chen, F.; Guo, Y.; Li, X. Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit. *Iet Intell. Transp. Syst.* **2020**, *14*, 1210–1217. [CrossRef]
28. Jin, G.; Cui, Y.; Zeng, L.; Tang, H.; Feng, Y.; Huang, J. Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. *Transp. Res. Part Emerg. Technol.* **2020**, *117*, 102665. [CrossRef]
29. Huang, D.; Deng, Z.; Wan, S.; Mi, B.; Liu, Y. Identification and Prediction of Urban Traffic Congestion via Cyber-Physical Link Optimization. *IEEE Access* **2018**, *6*, 63268–63278. [CrossRef]
30. Yang, S.; Li, H.; Luo, Y.; Li, J.; Song, Y.; Zhou, T. Spatiotemporal Adaptive Fusion Graph Network for Short-Term Traffic Flow Forecasting. *Mathematics* **2022**, *10*, 1594. [CrossRef]
31. Zhang, B.; Hirayama, K.; Ren, H.; Wang, D.; Li, H. Ship Anomalous Behavior Detection Using Clustering and Deep Recurrent Neural Network. *J. Mar. Sci. Eng.* **2023**, *11*, 763. [CrossRef]
32. Lv, W.; Lv, Y.; Ouyang, Q.; Ren, Y. A Bus Passenger Flow Prediction Model Fused with Point-of-Interest Data Based on Extreme Gradient Boosting. *Appl. Sci.* **2022**, *12*, 940. [CrossRef]
33. Koesdwiady, A.; Soua, R.; Karray, F. Improving Traffic Flow Prediction with Weather Information in Connected Cars: A Deep Learning Approach. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9508–9517. [CrossRef]
34. Dell'Acqua, P.; Bellotti, F.; Berta, R.; De Gloria, A. Time-Aware Multivariate Nearest Neighbor Regression Methods for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3393–3402. [CrossRef]
35. Tan, M.C.; Wong, S.C.; Xu, J.M.; Guan, Z.R.; Zhang, P. An Aggregation Approach to Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 60–69. [CrossRef]
36. Alghamdi, T.; Elgazzar, K.; Bayoumi, M.; Sharaf, T.; Shah, S. Forecasting Traffic Congestion Using ARIMA Modeling. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 1227–1232. [CrossRef]
37. Smith, B.L.; Williams, B.M.; Keith Oswald, R. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transp. Res. Part Emerg. Technol.* **2002**, *10*, 303–321. [CrossRef]
38. Feng, X.; Ling, X.; Zheng, H.; Chen, Z.; Xu, Y. Adaptive Multi-Kernel SVM with Spatial–Temporal Correlation for Short-Term Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2001–2013. [CrossRef]
39. Yan, X.; Liu, Y.; Mao, Z.; Li, Z.H.; Tan, H.Z. SVM-based Elevator Traffic Flow Prediction. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; Volume 2, pp. 8814–8818. [CrossRef]
40. Sun, Y.; Leng, B.; Guan, W. A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. *Neurocomputing* **2015**, *166*, 109–121. [CrossRef]
41. Cui, F. Study of Traffic Flow Prediction Based on BP Neural Network. In Proceedings of the 2010 2nd International Workshop on Intelligent Systems and Applications, Wuhan, China, 22–23 May 2010; pp. 1–4. [CrossRef]
42. Guo, X.; Deng, F. Short-Term Prediction of Intelligent Traffic Flow Based on BP Neural Network and ARIMA Model. In Proceedings of the 2010 International Conference on E-Product E-Service and E-Entertainment, Henan, China, 7–9 November 2010; pp. 1–4. [CrossRef]
43. Song, X.; Guo, Y.; Li, N.; Zhang, L. Online Traffic Flow Prediction for Edge Computing-Enhanced Autonomous and Connected Vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 2101–2111. [CrossRef]

44. Yu, F.; Wei, D.; Zhang, S.; Shao, Y. 3D CNN-based Accurate Prediction for Large-scale Traffic Flow. In Proceedings of the 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 5–7 September 2019; pp. 99–103. [CrossRef]
45. Zhang, K.; Liu, Z.; Zheng, L. Short-Term Prediction of Passenger Demand in Multi-Zone Level: Temporal Convolutional Neural Network with Multi-Task Learning. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1480–1490. [CrossRef]
46. Chiang, W.L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; Hsieh, C.J. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 4–8 August 2019; KDD '19, pp. 257–266. [CrossRef]
47. Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; Cui, P. Structural Deep Clustering Network. In Proceedings of the Web Conference 2020, New York, NY, USA, 20–24 April 2020; WWW '20, pp. 1400–1410. [CrossRef]
48. Xia, W.; Wang, Q.; Gao, Q.; Zhang, X.; Gao, X. Self-Supervised Graph Convolutional Network for Multi-View Clustering. *IEEE Trans. Multimed.* **2022**, *24*, 3182–3192. [CrossRef]
49. Qi, C.; Zhang, J.; Jia, H.; Mao, Q.; Wang, L.; Song, H. Deep face clustering using residual graph convolutional network. *Knowl.-Based Syst.* **2021**, *211*, 106561. [CrossRef]
50. Du, G.; Zhou, L.; Li, Z.; Wang, L.; Lü, K. Neighbor-aware deep multi-view clustering via graph convolutional network. *Inf. Fusion* **2023**, *93*, 330–343. [CrossRef]
51. Li, C.; Tang, G.; Xue, X.; Saeed, A.; Hu, X. Short-Term Wind Speed Interval Prediction Based on Ensemble GRU Model. *IEEE Trans. Sustain. Energy* **2020**, *11*, 1370–1380. [CrossRef]
52. Sun, P.; Boukerche, A.; Tao, Y. SSGRU: A novel hybrid stacked GRU-based traffic volume prediction approach in a road network. *Comput. Commun.* **2020**, *160*, 502–511. [CrossRef]
53. Han, P.; Wang, W.; Shi, Q.; Yang, J. Real-time Short- Term Trajectory Prediction Based on GRU Neural Network. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019; pp. 1–8. [CrossRef]
54. Abdullah, S.M.; Periyasamy, M.; Kamaludeen, N.A.; Towfek, S.K.; Marappan, R.; Kidambi Raju, S.; Alharbi, A.H.; Khafaga, D.S. Optimizing Traffic Flow in Smart Cities: Soft GRU-Based Recurrent Neural Networks for Enhanced Congestion Prediction Using Deep Learning. *Sustainability* **2023**, *15*, 5949. [CrossRef]
55. Wang, Y.; Zheng, J.; Du, Y.; Huang, C.; Li, P. Traffic-GGNN: Predicting Traffic Flow via Attentional Spatial-Temporal Gated Graph Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18423–18432. [CrossRef]
56. Chen, Y.; Shu, T.; Zhou, X.; Zheng, X.; Kawai, A.; Fueda, K.; Yan, Z.; Liang, W.; Wang, K.I.K. Graph Attention Network with Spatial-Temporal Clustering for Traffic Flow Forecasting in Intelligent Transportation System. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 8727–8737. [CrossRef]
57. Wang, Y.; Jing, C.; Xu, S.; Guo, T. Attention based spatiotemporal graph attention networks for traffic flow forecasting. *Inf. Sci.* **2022**, *607*, 869–883. [CrossRef]
58. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3848–3858. [CrossRef]
59. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 914–921. [CrossRef]
60. Feng, S.; Huang, J.; Shen, Q.; Shi, Q.; Shi, Z. A Hybrid Model Integrating Local and Global Spatial Correlation for Traffic Prediction. *IEEE Access* **2022**, *10*, 2170–2181. [CrossRef]
61. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In Proceedings of theAAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 922–929. [CrossRef]
62. Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Li, Z. Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 29–31 January 2019; Volume 33, pp. 5668–5675. [CrossRef]
63. Yang, H.; Zhang, X.; Li, Z.; Cui, J. Region-Level Traffic Prediction Based on Temporal Multi-Spatial Dependence Graph Convolutional Network from GPS Data. *Remote Sens.* **2022**, *14*, 303. [CrossRef]
64. Cui, Z.; Henrickson, K.; Ke, R.; Wang, Y. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4883–4894. [CrossRef]
65. Yu, L.; Du, B.; Hu, X.; Sun, L.; Han, L.; Lv, W. Deep spatio-temporal graph convolutional network for traffic accident prediction. *Neurocomputing* **2021**, *423*, 135–147. [CrossRef]
66. Roy, A.; Roy, K.K.; Ali, A.A.; Amin, M.A.; Rahman, A.K.M.M. Unified Spatio-Temporal Modeling for Traffic Forecasting using Graph Neural Network. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8. [CrossRef]
67. Zhu, J.; Han, X.; Deng, H.; Tao, C.; Zhao, L.; Wang, P.; Lin, T.; Li, H. KST-GCN: A Knowledge-Driven Spatial-Temporal Graph Convolutional Network for Traffic Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15055–15065. [CrossRef]
68. Bao, Y.; Huang, J.; Shen, Q.; Cao, Y.; Ding, W.; Shi, Z.; Shi, Q. Spatial–Temporal Complex Graph Convolution Network for Traffic Flow Prediction. *Eng. Appl. Artif. Intell.* **2023**, *121*, 106044. [CrossRef]

69. Cao, S.; Wu, L.; Zhang, R.; Wu, D.; Cui, J.; Chang, Y. A Spatiotemporal Multiscale Graph Convolutional Network for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8705–8718. [CrossRef]
70. Ma, Y.; Lou, H.; Yan, M.; Sun, F.; Li, G. Spatio-temporal fusion graph convolutional network for traffic flow forecasting. *Inf. Fusion* **2024**, *104*, 102196. [CrossRef]
71. Shen, X.; Chen, J.; Zhu, S.; Yan, R. A decentralized federated learning-based spatial–temporal model for freight traffic speed forecasting. *Expert Syst. Appl.* **2024**, *238*, 122302. [CrossRef]
72. Qian, W.; Nielsen, T.D.; Zhao, Y.; Larsen, K.G.; Yu, J.J. Uncertainty-Aware Temporal Graph Convolutional Network for Traffic Speed Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8578–8590. [CrossRef]
73. Liu, T.; Jiang, A.; Zhou, J.; Li, M.; Kwan, H.K. GraphSAGE-Based Dynamic Spatial–Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11210–11224. [CrossRef]
74. Xu, Y.; Lu, Y.; Ji, C.; Zhang, Q. Adaptive Graph Fusion Convolutional Recurrent Network for Traffic Forecasting. *IEEE Internet Things J.* **2023**, *10*, 11465–11475. [CrossRef]
75. Wang, Y.; Jing, C.; Huang, W.; Jin, S.; Lv, X. Adaptive Spatiotemporal InceptionNet for Traffic Flow Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3882–3907. [CrossRef]
76. Weng, W.; Fan, J.; Wu, H.; Hu, Y.; Tian, H.; Zhu, F.; Wu, J. A Decomposition Dynamic graph convolutional recurrent network for traffic forecasting. *Pattern Recognit.* **2023**, *142*, 109670. [CrossRef]
77. Xu, Y.; Han, L.; Zhu, T.; Sun, L.; Du, B.; Lv, W. Generic Dynamic Graph Convolutional Network for traffic flow forecasting. *Inf. Fusion* **2023**, *100*, 101946. [CrossRef]
78. Tian, R.; Wang, C.; Hu, J.; Ma, Z. Multi-scale spatial-temporal aware transformer for traffic prediction. *Inf. Sci.* **2023**, *648*, 119557. [CrossRef]
79. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.