mathematics

MDPI

Article

Adaptive Graph Convolutional Recurrent Network with Transformer and Whale Optimization Algorithm for Traffic Flow Prediction

Chen Zhang [1,2,3], Yue Wu [1,*], Ya Shen [1], Shengzhao Wang [1], Xuhui Zhu [3] and Wei Shen [1]

[1] Department of Artificial Intelligence and Big Data, Hefei University, Hefei 230601, China; zhangchen@hfuu.edu.cn (C.Z.); yashen_diligent@163.com (Y.S.); shengzhaowsz@163.com (S.W.); shenwei1998v@163.com (W.S.)
[2] Guochuang Software Co., Ltd., Hefei 230094, China
[3] Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei 230009, China; zhuxuhui@hfut.edu.cn
* Correspondence: wy_123go@163.com

Abstract: Accurate traffic flow prediction plays a crucial role in the development of intelligent traffic management. Despite numerous investigations into spatio-temporal methods, achieving high accuracy in traffic flow prediction remains challenging. This challenge arises from the complex dynamic spatio-temporal correlations within the traffic road network and the limitations imposed by the selection of hyperparameters based on experiments and manual experience, which can affect the performance of the network architecture. This paper introduces a novel transformer-based adaptive graph convolutional recurrent network. The proposed network automatically infers the interdependencies among different traffic sequences and incorporates the capability to capture global spatio-temporal correlations. This enables the dynamic capture of long-range temporal correlations. Furthermore, the whale optimization algorithm is employed to efficiently design an optimal network structure that aligns with the requirements of the traffic domain and maximizes the utilization of limited computational resources. This design approach significantly enhances the model's performance and improves the accuracy of traffic flow prediction. The experimental results on four real datasets demonstrate the efficacy of our approach. In PEMS03, it improves MAE by 2.6% and RMSE by 1.4%. In PEMS04, improvements are 1.6% in MAE and 1.4% in RMSE, with a similar MAPE score to the best baseline. For PEMS07, our approach shows a 4.1% improvement in MAE and 2.2% in RMSE. On PEMS08, it surpasses the current best baseline with a 3.4% improvement in MAE and 1.6% in RMSE. These results confirm the good performance of our model in traffic flow prediction across multiple datasets.

Keywords: spatio-temporal correlation; adaptive graph convolutional recurrent network; transformer; whale optimization algorithm; traffic prediction

MSC: 68U01

Citation: Zhang, C.; Wu, Y.; Shen, Y.; Wang, S.; Zhu, X.; Shen, W. Adaptive Graph Convolutional Recurrent Network with Transformer and Whale Optimization Algorithm for Traffic Flow Prediction. *Mathematics* **2024**, *12*, 1493. https://doi.org/10.3390/math12101493

Academic Editor: Andrea Scozzari

Received: 6 April 2024
Revised: 5 May 2024
Accepted: 8 May 2024
Published: 10 May 2024

1. Introduction

The transportation system is a crucial type of infrastructure in modern cities. As urbanization progresses, the growing urban population and the number of vehicles on the transportation network contribute to the increasing complexity of the traffic system. Consequently, there is an urgent need that lies in the development of Intelligent Transportation Systems (ITS). Early intervention based on traffic flow prediction is a crucial prerequisite for implementing ITS as it improves the efficiency of a transportation system, mitigates traffic-related problems, and facilitates the development of smart cities [1]. By analyzing the past data on traffic flow, it can help with accurate traffic flow prediction attempts that can

anticipate the future circumstances of traffic on road networks. The intelligent management of road networks in metropolitan areas is made possible by accurate and prompt traffic flow forecasts, which also decrease traffic congestion and improve traffic efficiency. To acquire the traffic flow status of a city, various sources of information can be utilized, including a transportation network, private vehicle movements, taxi tracks, and public transportation transaction records that are captured by sensors [2].

Nonetheless, achieving accurate traffic prediction has become progressively more challenging. On the one hand, traffic data are intrinsically a time series characterized by intricate temporal dependencies, exhibiting periodicity, volatility, uncertainty, and nonlinearity along the time dimension. Traffic data at a specific location exhibit nonlinear variations at distinct points in time, rendering the long-term prediction of traffic flow challenging. On the other hand, traffic data exhibit intricate dynamic spatial correlations, and the variability of traffic flows across different regional patterns is significant. Figure 1 illustrates an actual road network in a given region, thereby showing the dynamic correlation of traffic flows across geography. The complex spatial and temporal relationships in a road traffic network can have a significant impact on the accuracy of a traffic flow prediction system. In reality, traffic flow can show significant differences in different areas and time periods, with frequent congestion in office areas (such as from business districts) and residential areas (such as from dwelling districts) during morning and evening peak hours. In addition, traffic flows in commercial areas increase significantly during holidays. In addition, the intricate relationship between vehicles and roads in the spatial dimension complicates the accurate prediction of traffic flow. For example, disruptions in traffic flow due to temporary road closures for maintenance and unpredictable traffic accidents over a period of time can have an impact on distant roads. In addition, the complexity of roadway intersections poses additional challenges for the accurate prediction of traffic flow.

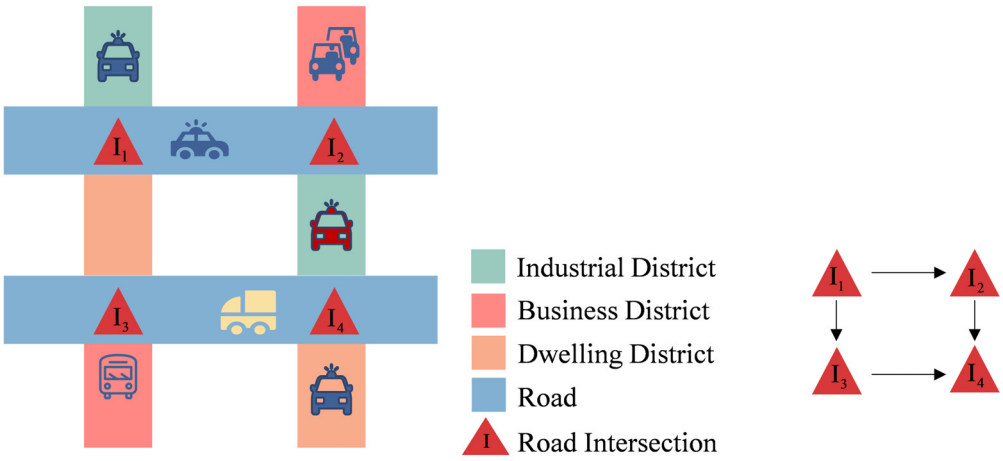

Figure 1. Road network of a certain zone.

To tackle the aforementioned challenges, researchers have extensively investigated the issue. The currently available methods may be roughly divided into four categories: conventional methods of statistical analysis, methods of machine learning, methods of deep learning, and graphical neural network methods. The predominant statistical methods employed include Autoregressive Integrated Moving Average (ARIMA) [3] and Vector Autoregression (VAR) [4], both of which rely on the assumption of ideal smoothness [5]. However, traffic road networks display dynamic and complicated behaviors, and statistically based approaches fall short in capturing nonlinear correlations, leading to large inaccuracies when forecasting enormous volumes of traffic data defined by complex spatio-temporal properties. However, machine learning techniques excel in capturing nonlinear relationships, leading to the application of classical methods such as Support Vector Regression (SVR) [6] and K-Nearest Neighbor [7] in traffic flow prediction. Nonetheless, these

models still exhibit limited performance when mining intricate spatio-temporal correlations. Deep learning methods effectively extract abstract features from raw data using multilayer neural networks. Deep learning methods excel in deriving traffic flow information from past data and producing precise forecasts, which is in contrast to machine learning methods that depend on features that are manually generated. Consequently, methods based on deep learning for predicting traffic flow have become more popular recently. Notably, Recurrent Neural Networks (RNNs), encompassing Long Short-Term Memory (LSTM) [8] and Gated Recurrent Unit (GRU) [9], effectively capture the temporal correlation within traffic data. Nevertheless, these RNN models ignore the spatial correlations present in spatio-temporal data and interpret traffic sequences from different roadways as separate data streams. While convolutional neural networks (CNNs) [10] excel in capturing spatial correlations within regular spatial grids, traffic road networks pose a challenge as they possess a topologically complex non-Euclidean data structure. On the other hand, graph neural networks (GNNs) [11] have the capability to directly model non-Euclidean data; as such, graph convolutional networks (GCNs) [12] and graph attention networks (GATs) [13] are employed to capture the spatial correlations present in traffic road networks.

Due to the inherent temporal and spatial correlations contained in traffic flow data, focusing exclusively on modeling time or space is inadequate for traffic flow prediction. Temporal modeling ignores the influence of geographical location on traffic flow and only concentrates on the temporal patterns of traffic flow. For instance, during specific morning peak hours, the traffic flow on residential roads might be influenced by nearby office clusters, an aspect that temporal modeling fails to capture. In contrast, spatial modeling solely concentrates on capturing the variation pattern of traffic flow in the spatial dimension while disregarding the influence of time on traffic flow. For example, within a commercial area, traffic flow exhibits fluctuations during particular time periods, such as holidays, resulting in a significant surge in traffic. This phenomenon cannot be sufficiently captured through spatial modeling alone. Consequently, it is essential to take into account both temporal and spatial correlations in order to improve the accuracy of traffic flow forecasts. Hence, several studies ([14–18]) in the field of spatio-temporal modeling commonly employ a combination of RNNs and GNNs to capture the intricate spatio-temporal relationships within traffic data. The network model's accuracy is increased by the attention mechanism, which enables the model to concentrate more on the data that are pertinent to the current issue. Therefore, the attention mechanism is widely employed in spatio-temporal methods for traffic flow prediction. By utilizing the attention mechanism, the model can prioritize the significant locations and time points associated with traffic flow changes, thereby furnishing additional information to elucidate the prediction results of the model. To effectively capture the spatio-temporal correlations in traffic data, several studies have included attention processes together with independent modules. This is demonstrated by models such as Spatio-Temporal Graph Convolutional Network (STGCN) [18] and Attention-Based Spatio-Temporal Graph Convolutional Network (ASTGCN) [19]. Initially, these models capture temporal correlations through a dedicated module, and this is followed by forwarding the extracted temporal features to a module responsible for capturing spatial correlations and incorporating an attention mechanism. However, this approach diminishes the captured spatio-temporal dependence. Consequently, certain models strive to devise novel graph structures to address the challenging problem of spatio-temporal correlation in traffic data. Spatio-Temporal Synchronous Graph Convolutional Network (STSGCN) [20] addresses the challenge of capturing spatio-temporal correlations synchronously by constructing local spatio-temporal maps. Spatial Temporal Graph Neural Network (STGNN) [21] integrates GRU and transformer [22] models to capture both local and global temporal dependencies, thereby demonstrating the effectiveness of the attention mechanism in capturing the long-term temporal relationships of Spatio-Temporal Fusion Graph Neural Networks (STFGNN) [23], which achieves the simultaneous capture of spatio-temporal correlations by generating spatio-temporal maps and fusing features. However, these models do not account for the dynamic spatio-temporal dependencies among the nodes in the traffic road

network. To capture dynamic spatio-temporal correlations, adaptive graph convolutional recurrent network (AGCRN) [24] automatically infers the interdependencies among different traffic sequences through two adaptive modules and a learnable node embedding matrix. However, it lacks the inclusion of attention mechanisms to capture both short-term and long-term temporal correlations. Graph Convolutional Dynamic Recurrent Network with Attention [25] integrates the attention mechanism into the graph convolution and dynamic GRU to capture the long-term temporal dependencies in traffic flow. However, its performance is highly dependent on the K-value in the k-hopGC module, which employs a k-hop neighbor matrix to extend the receptive field of the GCN; thus, it requires several manual tests to determine the optimal K-value. Multi-Attention Predictive Recurrent Neural Network [26] combines convolutional neural networks and predictive recursive neural networks to extract spatio-temporal information from traffic flow data. However, it still lacks sufficient focus on the correlation between global information and comprehensive features of traffic flow.

To address the above challenges, this paper introduces a novel deep learning framework, named Adaptive Graph Convolutional Recurrent Network with Transformer and Whale Optimization Algorithm (WOA-AGCRTN), which combines the transformer algorithm and WOA. The proposed network model has the capability to automatically infer the interdependencies among various traffic sequences while capturing both short-range and long-range spatio-temporal correlations within traffic road networks. This enhanced capability enables the model to effectively capture global spatio-temporal correlations. In summary, the key contributions of this paper can be summarized as follows:

- We propose an adaptive graph convolutional recurrent network with the transformer algorithm. This network infers the interdependencies between traffic sequences and integrates the transformer technique to capture both long and short-term temporal dependencies.
- We propose utilizing whale optimization algorithms to design an optimal network structure that aligns with the transportation domain, thereby aiming to significantly enhance the accuracy of traffic flow prediction.
- The feasibility and advantages of the proposed network model are validated using four real datasets. The results from experiments on these datasets affirm the effectiveness of our method. In PEMS03, our model reduces MAE by 2.6% and RMSE by 1.4%. In PEMS04, improvements are 1.6% in MAE and 1.4% in RMSE. In PEMS07, a 4.1% MAE improvement and 2.2% in RMSE is exhibited. Moreover, in PEMS08, our model surpasses the baseline with a 3.4% MAE improvement and 1.6% in RMSE.
- We effectively address the challenge of long-range time dependence and significantly improve the performance of the network model compared to several baseline methods, including the most recent state-of-the-art approaches.

The remaining parts of this article are structured as follows: Section 2 presents some work related to traffic prediction and the swarm intelligence optimization algorithm. Section 3 presents the methodology of the proposed WOA-AGCRTN, and Section 4 presents and analyzes the experiment results. Finally, Section 5 details the conclusions and the prospects of our research.

2. Related Work

This section provides an overview of the existing literature for traffic flow prediction, categorizing it into two main areas: spatio-temporal data prediction methods and graph convolutional neural network methods. And swarm intelligence optimization algorithm-aided methods are analyzed for traffic flow prediction.

2.1. Spatio-Temporal Prediction

Traffic flow prediction is a subset of the broader field of spatio-temporal data prediction, which has garnered significant research attention over the past few decades. Among the available research methods for traffic flow prediction, the earliest models can be clas-

sified as either statistical-based or neural network-based. Statistical models, including ARIMA and VAR, capture spatial correlations in a probabilistic manner and aid in analyzing uncertainties within traffic flows. However, since these models rely on linear time series methods and assume ideal smoothness [5], they often fail to accurately predict actual complex nonlinear traffic data, thus rendering them less effective. To capture the intricate nonlinear relationships in traffic data, researchers have employed SVR, KNN, and neural network models to traffic prediction for traffic prediction. However, their fully connected structures impose significant computational and storage costs.

Compared with traditional methods, deep learning is effective in automatically capturing features in representational learning. For example, while overcoming the gradient vanishing and explosion issues of conventional recurrent neural networks, LSTM enhances the modeling of long sequence correlations and applies LSTM to traffic flow prediction. The proposed GRU simplifies model construction, decreases the parameter count, enhances computational efficiency, and mitigates the risk of overfitting. However, these RNN models (including RNN, LSTM, and GRU) consider traffic sequences from several roadways as separate data streams, which can only extract temporal information but cannot capture spatial information from the traffic sequences. Deep Spatio-Temporal Residual Networks (ST-ResNets) [10], which leverage CNNs to extract spatial features and capture spatial correlations by treating the traffic road network as Euclidean data. However, the intricate and irregular topology of the traffic road network severely limits the spatial correlations captured by CNNs.

GNNs were subsequently proposed. GNNs are capable of modeling non-Euclidean data and have demonstrated strong performance in various forecasting tasks at the node, edge, and graph levels [2]. Recently, numerous baseline methods have utilized GNNs, underscoring their prominent position in deep learning-based traffic flow prediction research. For instance, Diffusion Convolutional Recurrent Neural Network (DCRNN) [14] employs diffusion processes on directed graphs to capture the spatial correlations of traffic road networks. By combining GRU with diffusion GCNs, DCRNN achieves enhanced performance in traffic prediction. Nevertheless, traffic road networks possess intricate and dynamic topological structures, thus rendering the fixed graph structure and static adjacency matrix utilized by DCRNNs and STGCNs to be inadequate in representing these dynamic changes, such as morning and evening peak periods or traffic accidents. To address this, Graph WaveNet [27] introduces an adaptive adjacency matrix that models dynamic spatial correlations through GCNs. However, the adaptive matrix employed in Graph WaveNet remains fixed after training, thus impeding its adaptability to diverse traffic road networks. ASTGCN captures temporal and spatial correlations independently through dedicated temporal and spatial attention mechanisms. AGCRN leverages two adaptive modules and a learnable node embedding matrix to infer the interdependencies among different traffic sequences. Dynamic Spatial–Temporal Aware Graph Neural Network (DSTAGNN) [28] introduces a novel spatio-temporal attention module to capture the dynamic spatio-temporal dependencies among nodes along with a gated convolution module for improving the capture of temporal dynamics in traffic data.

In contrast to the aforementioned approaches, some models aim to develop novel graph structures. STSGCN addresses the challenge of capturing synchronous spatio-temporal correlations through the construction of local spatio-temporal graphs. STFGNN achieves the concurrent capture of spatio-temporal correlations by generating spatio-temporal graphs and fusing features. Spatio-Temporal Graph Convolutional Ordinary Differential Equation (STGODE) [29] enhances GCNs by incorporating ordinary differential equations (ODEs) using a combination of semantic and static spatial adjacency matrices, which results in improved performance. However, these models neglect the dynamic spatio-temporal dependencies among nodes in the traffic road network.

2.2. Graph Convolution Networks

GCNs extend classical convolutional methods to graph structures, enabling the capture of information pertaining to the neighbors of nodes and edges. GCNs, which are widely employed in various tasks, are primarily categorized into two approaches: spectral GCNs. Bruna et al. [6] employed the Laplacian matrix in the spectral domain to enable graph expansions for convolutional operations. However, the computation involved is computationally expensive. Defferrard et al. [30] enhanced the traditional GCN by introducing ChebNet, which employs a Chebyshev polynomial expansion of the diagonal matrix based on eigenvalues. With the help of this approximation method, graph convolution becomes less computationally demanding. With the classical GCN, Kip and Welling [31] simplified ChebNet and utilized it in a CNN-like deep network architecture for effectively embedding graph structure and node attributes. Another approach is the spatial GCN, with which Micheli and Alessio [32] captured node representations by convolving information from the aggregated features of their neighbors, as demonstrated in methods like GraphSAGE [33]. Velickovic et al. proposed the GAT [13], which incorporates an attention mechanism into graph convolution to dynamically adjust the connections between neighboring nodes and measure node importance based on the magnitude of the weights.

2.3. Swarm Intelligence Optimization Algorithm

Although graph convolution networks have powerful automatic feature extraction capability for unstructured data, the design of network architecture models relies heavily on the researcher's prior knowledge and experience, and the optimal combination of hyperparameters selected based on experiments and manual experience may limit the emergence of new optimal network architectures to some extent. Therefore, we need to design the network model structure with the help of algorithms, and swarm intelligence algorithms are a type of swarm-based gradient-free optimization algorithms that have seen significant application recently for the construction of neural networks and the modification of hyperparameters. The swarm intelligence optimization algorithm is a kind of computational intelligence algorithm. This algorithm's fundamental idea is to mimic the natural behavior of animal species, including fish, birds, and wolves. It makes use of group collaboration and knowledge sharing to optimize a given issue through straightforward and constrained individual interactions. In 2016, Mirjalili et al. [34] introduced the whale optimization algorithm (WOA), a novel swarm intelligence algorithm that specifically mimics the hunting behavior of humpback whales when they are in close proximity to their prey. The primary objective of WOA is to seek the best possible solution for intricate optimization problems. Compared to other swarm intelligence optimization methods, including Particle Swarm Optimization [35], Gravitational Search Algorithm [36], Genetic Algorithm [37], Grey Wolf Optimizer [38], and Ant Colony Optimization [39], WOA is a recently developed and widely used optimization technique across multiple tested scenarios. The use of swarm intelligence algorithms to optimize the hyperparameters and structure of neural networks has received increasing academic attention. Consequently, numerous scholars have begun employing this algorithm in practical applications. For instance, WOA_BiLSTM_Attention [40] uses WOA to optimize the BiLSTM_Attention network model for traffic flow prediction to obtain its four optimal parameters, including the learning rate, the number of training sessions, and the number of nodes in both hidden layers. The accuracy of traffic flow prediction is thus improved. Pham Q V [41] et al. have utilized WOA to address the resource allocation problem in wireless networks. Pradeep Jangir [42] et al. employed WOA to train a multilayer perceptron for addressing the local optima problem, thereby achieving a high level of accuracy in optimizing the system. This paper is the first to optimize AGCRN using WOA to improve the accuracy of traffic flow predictions.

3. Methodology

3.1. Problem Statement and Preliminaries

Traffic flow prediction is a common time series prediction problem. In our study, we modeled the traffic road network as a graph denoted by $G = (V, E, A)$, where $V = \{v_1, \cdots\cdots, v_n\}$ represents the set of N traffic sensor nodes (observation points) and E represents the set of edges connecting these nodes. $A \in R^{N \times N}$ is the adjacency matrix of the traffic network graph G, if $v_i, v_j \in V$ and $(v_i, v_j) \in E$, which then makes $A_{ij} = 1$. The traffic flow prediction task is to use the observed historical traffic sequence data $X^{(t-M+1)}, X^{(t-M+2)}, \ldots, X^{(t)}$ to predict the future traffic sequence data $X^{(t+1)}, X^{(t+2)}, \ldots, X^{(t+T)}$, where $X^{(t+T)} \in R^{N \times C}$ denotes the observed value at the time step t, C denotes the number of feature channels, and T denotes the length of the time series to be predicted. Given the recorded data $X^{(t-M+1):(t)} \in R^{N \times M}$, then, by training the model $\mathcal{F}$ to predict the traffic volume on the road network G for the next T time steps, $X^{(t+1):(t+T)} \in R^{N \times T}$. The traffic flow prediction problem is formally defined as presented in Equation (1):

$$X^{(t+1):(t+T)} = \mathcal{F}\left[X^{(t-M+1):(t)}; G\right], \tag{1}$$

where $\mathcal{F}$ is a function of the learning historical traffic flow sequence information.

3.2. The Proposed Algorithm

This section presents the architecture of the WOA-AGCRTN model, as depicted in Figure 2. The model comprises four main components: (1) Adaptive graph convolutional recurrent network (AGCRN), which includes NAPL-DAGG-GCN and GRU modules. (2) The transformer layer that enhances long-range time correlations by performing correlation modeling in the time dimension. This contributes to the construction of adaptive graph convolutional recurrent network based on transformer (AGCRTN). (3) The transformer layer that captures time correlations across long ranges to enhance global time correlations. This layer is responsible for constructing the transformer-based adaptive graph convolutional recurrent network. We employ the whale optimization algorithm to optimize the network architecture and hyperparameters of AGCRTN, thereby aiming to enhance the model's performance. (4) WOA is employed to enhance the model's performance by optimizing both the network architecture and hyperparameters of AGCRTN (WOA-AGCRTN). The specific optimization parameters include the learning rate, learning rate decay, the number of layers in the implicit GRU layer, the number of neurons in the implicit layer, the number of layers in the transformer, and the number of heads in the multi-headed attention module. These parameters significantly influence the model's performance, and finding the optimal combination based on human experience is challenging. Consequently, the whale optimization algorithm is employed to globally optimize these parameters and determine the best parameter combination, thus leading to the reconstruction of the AGCRTN network model for training and prediction purposes. The model's specifics are elaborated in the subsequent subsections.

3.2.1. Adaptive Graph Convolutional Recurrent Network

AGCRN is enhanced by two adaptive modules for GCN: the Node Adaptive Parameter Learning (NAPL) module and the Data Adaptive Graph Generation (DAGG) module. To augment the conventional GCN, the NAPL module learns two smaller parameter matrices: (1) the node embedding matrix $E_G \in R^{N \times d}$, where d is the number of embedding dimensions; and (2) the weight pool $W_G \in R^{d \times C \times F}$, which has the bias $b_G \in R^{d \times F}$. Thus, the NAPL-augmented GCN is as follows Equation (2):

$$Z = (I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) X E_G W_G + E_G b_G. \tag{2}$$

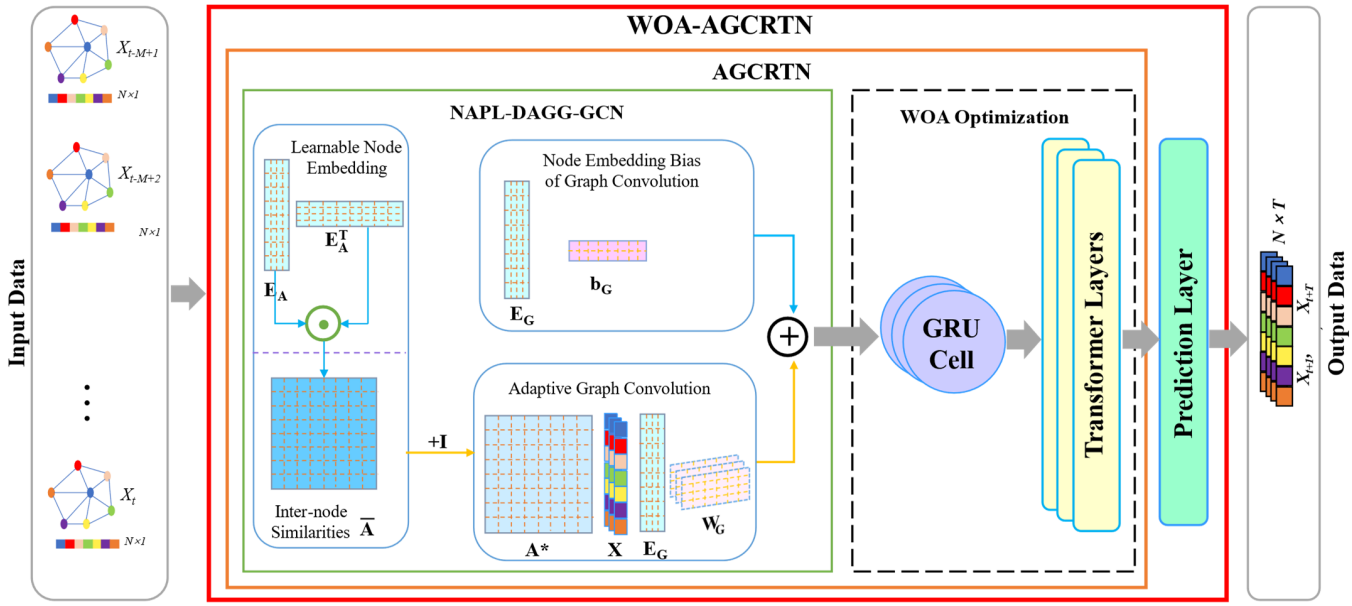

Figure 2. The overall architecture of WOA-AGCRTN.

To automatically infer the hidden interdependencies in the data, the DAGG module randomly initializes the learnable node embedding dictionary $E_A \in R^{N \times d_e}$ for all nodes, where each row of E_A denotes the embedding of a node and d_e denotes the node embedding dimension. The spatial dependencies between each pair of nodes are then inferred through a node similarity definition graph, which is as follows Equation (3):

$$D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = softmax(ReLU(E_A \cdot E_A^T)). \tag{3}$$

Finally, NAPL-DAGG-GCN can be represented as follows Equation (4):

$$Z = (I_N + softmax(ReLU(E_A \cdot E_A^T)))XE_G W_G + E_G b_G. \tag{4}$$

To capture local temporal dependencies sequentially, the AGCRN model utilizes the GRU mechanism for processing sequence information, thus preserving a hidden representation for each time step. This hidden representation serves the dual purpose of regulating information flow to subsequent time steps and serving as the output for the current time step. Notably, the MLP layer within the GRU is substituted with NAPL-DAGG-GCN to capture the temporal correlation of specific nodes in the traffic sequence. More precisely, when presented with an input X_t, the GRU operates using the equation below for each node v_i at time step t, which is as follows Equation (5):

$$\begin{aligned} \widetilde{A} &= softmax(ReLU(EE^T)) \\ z_t &= \sigma(\widetilde{A}\{X_t[i,:], h_{t-1}[i,:]\}EW_z + Eb_z) \\ r_t &= \sigma(\widetilde{A}\{X_t[i,:], h_{t-1}[i,:]\}EW_r + Eb_r) \\ \hat{h}_t[i,:] &= \tanh(\{X_t[i,:], r \odot h_{t-1}[i,:]\}E_{\hat{h}} + Eb_{\hat{h}}) \\ h_t[i,:] &= z \odot h_{t-1}[i,:] + (1-z) \odot \hat{h}_t[i,:] \end{aligned}, \tag{5}$$

where $\{\cdot\}$ represents the concatenation operation, $\odot$ represents the element-by-element multiplication, and $h_t[i,:]$ signifies the output of the current time step, which is further employed as the input for the subsequent time step. $E, W_z, W_r, b_z, b_r, b_{\hat{h}}$ are learnable parameters.

3.2.2. Adaptive Graph Convolutional Recurrent Network Based on Transformer

While GRU is beneficial for capturing local temporal information, traffic flow prediction problems involve more than just sequential correlation. Thus, it becomes crucial to capture global temporal information. In contrast to STFGNN and DSTAGNN, we incorporated the transformer algorithm after the GRU layer to directly capture global dependencies. Notably, the transformer algorithm fully replaces the convolutional structure to fulfill this task. The transformer technique is used to capture the global temporal dependencies in traffic flow because of its effectiveness in accurately describing long-distance and multi-scale characteristics, which improves model generalization.

In the temporal dimension of traffic data, the current traffic state in a spatial region is closely linked to the traffic states before and after the current time in the same region. To more effectively capture the long-term temporal dependence of traffic data patterns, we propose integrating a transformer layer. This layer enables modeling of the global contextual information within the sequence. The layer comprises a location embedding, a multi-headed attention layer, a feedforward network layer, and a normalization layer. The transformer layer is applied to each node independently, with the output sequence $(h_1[i,:], \cdots, h_t[i,:])$ from the GRU serving as the input for the transformer layer. The multi-headed attention mechanism in the transformer layer allows the model to capture long-term temporal patterns and prioritize important information, thus addressing the limitations of GRU in modeling time dependencies. This is illustrated in Figure 3.

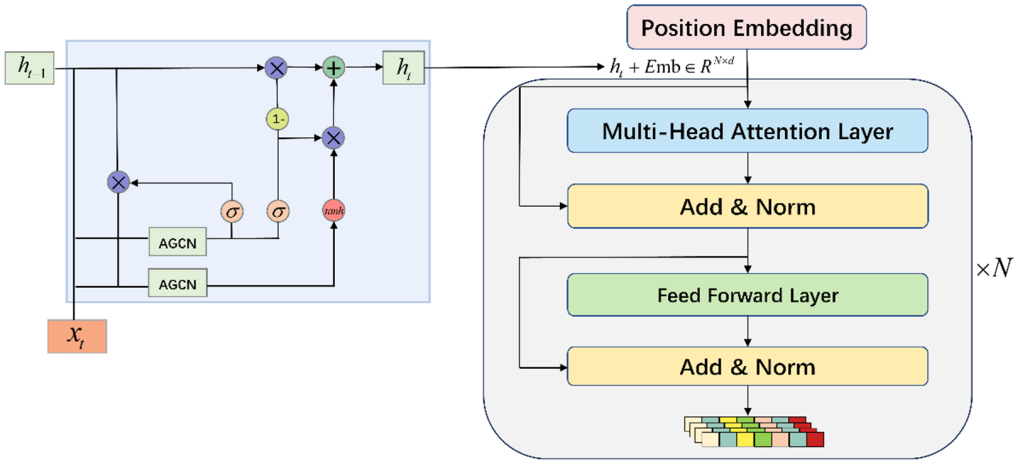

Figure 3. The transformer layer.

In the Transformer layer, the multi-headed attention layer employs a self-attentive mechanism to dynamically capture the spatial dependencies among sensors. The self-attentive mechanism is extensively employed in various domains, such as computer vision and natural language processing, to capture the relationships between individual elements. Similarly, GAT utilizes the self-attentive mechanism to calculate the weights between nodes. However, due to the requirement of a predefined and static graph topology, the nodes can only dynamically model each other. In real traffic road networks, there exist hidden spatial dependencies that are not entirely captured by the road topology. Therefore, the graph and edge weights need to be dynamically constructed.

First, we explain the principle of the single-headed attention mechanism and then elaborate on its extension to a multi-headed attention mechanism. In the Transformer, the multi-headed attention layer employs normalized Scaled Dot-Product Attention, thereby enabling elements at each position in the sequence to depend on all elements in the sequence. The attention function takes the input queries of dimension d_k, as well as the keys and values of dimension d_v for each position in the sequence. We compute the dot product between the queries and keys, divide the result by $\sqrt{d_k}$, and apply the Softmax function to obtain the attention scores for each position. The computed attention scores serve as

weights to represent the level of focus on specific information. The formula for calculating the attention score across all positions is as follows Equation (6):

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{6}$$

where $Q, K \in R^T \times d_k, V \in R^T \times d_v$ represent the queries, keys, and values of all nodes. In particular, the query at position i in the sequence corresponds to the i-th row of Q. Initially, we stack the GRU outputs $(h_1[1,:], \ldots, h_t[i,:])$ row by row in a sequential manner, thus resulting in the matrix $h^{v_i} \in R^{T \times d}$. The superscript v_i represents the corresponding node v_i in the traffic network. Next, we perform a linear projection of matrix h^{v_i} into the queries, keys, and values as follows Equation (7):

$$Q^{v_i} = h^{v_i} W^Q, K = h^{v_i} W^K, V = h^{v_i} W^V, \tag{7}$$

where $W^Q \in R^{d \times d_k}$, $W^K \in R^{d \times d_k}$, and $W^V \in R^{d \times d_v}$ are the projection matrices to be learned, and they are shared by all nodes in the traffic road network. Consequently, the function to compute the attention score can be expressed as follows Equation (8):

$$Attention(h^{v_i}) = softmax\left(\frac{(h^{v_i} W^Q)(h^{v_i} W^K)^T}{\sqrt{d_k}}\right) h^{v_i} W^V. \tag{8}$$

In this work, the reason for the choice of a multi-headed attention mechanism is that it provides a parallel mechanism to aggregate information from different subspaces focusing on different locations, thus enhancing the representation capability of the model. Consequently, it becomes possible to effectively concentrate on the long-term relevance of time series data. Within the multi-headed attention mechanism, the K-group projection matrix is utilized to project h^{v_i} into distinct K-groups, which comprise queries, keys, and values. The multi-headed attention score is then computed as a concatenation of the individual output scores from each single-headed attention function, which is expressed as follows Equations (9) and (10):

$$Multihead(h^{v_i}) = Concat(head_1, \ldots, , head_C) W^O, \tag{9}$$

$$head_i = Attetion_i(h^{v_i}) = softmax\left(\frac{(h^{v_i} W_i^Q)(h^{v_i} W_i^K)^T}{\sqrt{d_k}} h^{v_i} W_i{}^V\right). \tag{10}$$

The matrices $W_i{}^Q \in R^{d \times d_k}$, $W_i{}^K \in R^{d \times d_k}$, and $W_i{}^V \in R^{d \times d_v}$ correspond to the projection matrices of the i-th attention head, while $W_i{}^V \in R^{d \times d_v}$ (where h represents the number of heads in the multi-headed attention) denotes the linearly projected output combining their results.

After the feature sequence $(h_1[1,:], \ldots, h_t[i,:])$ output by the AGCRN module is sent to the transformer layer, we first add the position vector to the input data at each moment via considering that the attention mechanism in the transformer layer ignores the relative positions in the sequence during the dot product operation. This is shown as follows Equation (11):

$$h'_t[i,:] = h_t[i,:] + e_t, \tag{11}$$

where the Location Code Token A is defined as Equation (12):

$$e_t = \begin{cases} sin(t/10000^{2i/dmodel}), if\ t = 0, 2, 4, \ldots \\ cos(t/10000^{2i/dmodel}), otherwise \end{cases}. \tag{12}$$

Figure 2 illustrates the flow of information in the model. The output state from the multi-headed attention layer is sequentially processed through the normalization layer, feedforward network layer, and transformer layer, thereby resulting in the output denoted

as $h_{out}^{v_i} \in R^{T \times d}$. Subsequently, $h_{out}^{v_i}$ is utilized as the input for the prediction layer, which consists of a fully connected layer, to forecast the future traffic flow.

3.2.3. Whale Optimization Algorithm

Deep learning has achieved remarkable breakthroughs and advancements across various fields owing to its potent ability to automatically capture features from unstructured data. However, it has been demonstrated that the design of network architecture is crucial for achieving optimal performance. Nevertheless, designing the hyperparameters of complex network models heavily depends on researchers' prior knowledge and experience. However, due to the inherent limitations of human knowledge, it remains challenging to devise an optimal network architecture. Therefore, this work adopts the WOA approach to automatically design the most effective network structure while minimizing human intervention.

WOA is a swarm-based algorithm that obtains it inspiration from whale dietary habits. Due to its simplicity, versatility, and strong optimization capabilities, WOA has gained wide acceptance among scholars, and it has been used for neural network optimization. The two components of the WOA optimization process are illustrated as follows:

(1) Development stage

The development stage of the WOA includes both shrink-wrapping and bubble net attack mechanisms for position updates.

Shrinking envelope: The other individuals in the WOA move toward the whale that is currently in the best position, and they update their position by using Equations (13)–(16).

$$A = 2 \cdot a \cdot r_1 - a, \tag{13}$$

$$C = 2 \cdot r_2, \tag{14}$$

$$D = |C \cdot X^*(t) - X(t)|, \tag{15}$$

$$X(t+1) = X^*(t) - A \cdot D, \tag{16}$$

where t represents the current number of iterations, a linearly decreases from 2 to 0 with each iteration, and r_1 and r_2 are random vectors with values ranging from 0 to 1. $X^*(t)$ represents the leading whale in the WOA, denoting the individual with the highest fitness value at the given moment. D represents the distance between $X^*(t)$ and $X(t)$ after scaling by C. $X(t)$ represents the whale requiring position update, and $X(t+1)$ is its updated position.

Bubble net attack: In addition to shrink-wrapping, the WOA incorporates a bubble net-based attack for position updates, as described in the mathematical formulation presented in Equations (17) and (18).

$$D' = |X^*(t) - X(t)|, \tag{17}$$

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t), \tag{18}$$

where D' represents the distance between $X^*(t)$ and $X(t)$, b is a constant typically set to 1, and l is a randomly generated number in the range of 0 to 1. During the development phase, WOA utilizes two position update mechanisms: shrinkwrap and the bubble net attack. These mechanisms are randomly alternated, with each method chosen with a probability of 0.5. The formula for the WOA development phase is given by Equation (19), where p denotes a random number in the range of 0 to 1, which represents the probability of choosing the two position update mechanisms.

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D & if \quad p < 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) & if \; p \geq 0.5 \end{cases}. \tag{19}$$

(2) Exploration stage

In the exploration phase of WOA, the position update mechanism is the same as the shrinking envelope, as depicted in Equations (20) and (21). However, unlike the previous shrinkage envelope mechanism, it employs a randomly selected individual to guide its own position movement.

$$D = \left| C \cdot X^{rand}(t) - X(t) \right|, \tag{20}$$

$$X(t+1) = X^{rand}(t) - A \cdot D. \tag{21}$$

During the exploration phase of WOA, the $X^{rand}(t)$ is randomly chosen to guide the position update of $X(t)$. This mechanism enhances the algorithm's ability to escape local optima. The exploration phase of WOA is controlled by $|A|$. Equation (21) is triggered when $|A|$ exceeds 1. The fitness function in WOA represents the objective function of the problem to be optimized, and each whale within it represents a solution to that problem. Therefore, the fitness value of each whale in WOA is equal to the value calculated by the objective function of that problem. The specific optimization execution process of WOA is shown below.

Step 1: Initialize the important parameters of WOA as follows: population size P, current iteration number t, maximum number of iterations Max_t, the dimension of the objective function D, the lower bound of the value of the decision variables lb, and the upper bound ub.

Step 2: Initialize WOA's population and calculate the fitness value of each whale through the objective function to determine the whale with the best fitness value obtained thus far as $X^*(t)$.

Step 3: Update the following parameters: a, A, C, l, and p. These are based on computer-generated random numbers. Then, select Equations (16), (18), or (21) to update the positions of all whales based on the value of p and $|A|$.

Step 4: Restrict the updated positions of all whales to be within lb and ub, recalculate their fitness values, and update $X^*(t)$.

Step 5: Determine whether the current iteration reaches the maximum number of iterations, i.e., $t > Max\_t$. If not, proceed to the next iteration and repeat Steps 3 to 4; if yes, return the best whale obtained thus far as $X^*(t)$.

Step 6: Output $X^*(t)$ as the best solution optimized by WOA for this problem.

3.2.4. WOA-Optimized AGCRTN Method (WOA-AGCRTN)

The implementation of AGCRTN for traffic flow prediction considers spatial influence, long-term and short-range time dependence, and integrates multiple deep learning models. The increased complexity of the whole model and the enhanced tightness between parameters all increase the difficulty of parameter selection and model optimization. In contrast to traditional machine learning models, shallow models have a relatively simple structure and involve tuning fewer parameters. However, various hyperparameters still need to be taken into account. Unlike the traditional machine learning models, the shallow model structure is relatively simple, and the tuning of parameters is relatively single, so various hyperparameters need to be considered. The AGCRTN network model is influenced by the Lr-init and Lr-decay-rate, which significantly impact its training effectiveness. Additionally, Rnn-num-layers and Rnn-num-units play a decisive role in achieving an optimal model fit. Moreover, Transformer-num-layers and Transformer-num-heads are crucial for effectively capturing long-range time series. Finding the optimal combination of hyperparameters through experiments and manual exploration is a time-consuming process due to the challenge of exhaustively testing and evaluating all possible combinations. Therefore, the utilization of optimization algorithms is crucial in facilitating the selection of improved hyperparameters to enhance the performance of AGCRTN. In this study, the WOA is employed to identify the optimal combination of hyperparameters, thereby subsequently enabling the reconstruction of the AGCRTN network model for training and prediction. This approach significantly enhances the prediction accuracy of AGCRTN.

In this section, we describe the workflow of WOA-AGCRTN, the main workflow of which is shown in Figure 4. In the optimization process, each whale represents a set of hyperparameter combinations, so the lb and ub are the upper and lower bounds of the hyperparameters, the best whales are the best hyperparameter combinations found thus far, and the MAE on the validation set is used as the fitness value. We set the population size of WOA to 10 and the maximum number of iterations to 10. The primary objective of WOA-AGCRTN is to optimize the parameters of AGCRTN using the WOA, thereby enhancing its prediction accuracy. The specific steps for optimizing the AGCRTN model using WOA are as follows.

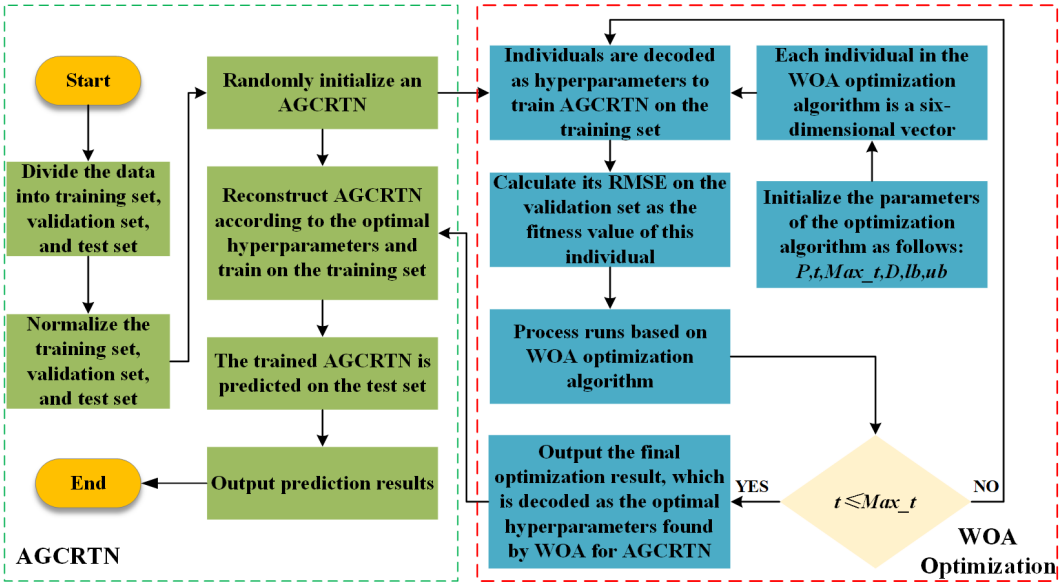

Figure 4. A flow chart of the WOA-optimized AGCRTN network model.

Step 1: The PEMS dataset is divided into a training set for training the model, a validation set for calculating the fitness value of each whale in WOA, and a test set for evaluating the prediction error of the model in the ratio of 6:2:2.

Step 2: Normalize the training set, validation set, and test set with the following formula:

$$x' = \frac{x - Min}{Max - Min}. \tag{22}$$

In this scenario, x represents the original feature value, x' denotes the normalized feature value, and Max and Min represent the maximum value and the minimum value of the feature, respectively.

Step 3: Configure the WOA's parameters as follows: the population size is set to P ($P = 10$); each individual has a dimension of 6, which matches the number of hyperparameters (Lr-init, Lr-decay-rate, Rnn-num-layers, Rnn-num-units, Transformer-num-layers, and Transformer-num-heads) to be optimized in this study (AGCRTN); and the maximum number of iterations is set to 10. The initial population of WOA is generated by randomly creating a matrix with P rows and 6 columns. Each row in the matrix represents an individual, and each individual is represented by a 6-dimensional vector.

Step 4: The fitness value of each individual in the optimization algorithm is determined by calculating the mean absolute error (MAE) between the true and predicted values on the validation set after decoding the individual as an AGCRTN hyperparameter. Therefore, the objective function is MAE, as shown in Equation (23), and its value on the validation set is used as the fitness value of each whale in WOA.

$$Fitness = MAE = \frac{1}{N}\sum_{I=1}^{N}\left|X_n^i - Y_n^i\right|, \tag{23}$$

where X_n^i denotes the n-th observation of the traffic flow on the validation set, which is the true value; Y_n^i denotes the predicted value of the model; and N denotes the total number of samples of data on the validation set. It should be noted that AGCRTN does not use the validation set during the training process. AGCRTN calculates its MAE on the validation set only after the training process is completely finished, and this MAE serves as the WOA's fitness value.

Step 5: WOA is executed according to the workflow described in Section 3.2.3.

Step 6: If the WOA reaches the termination condition, i.e., $t > Max\_t$, the final optimization result is output; otherwise, go back to Step 5.

Step 7: The optimal hyperparameters of AGCRTN are determined by using the values of each dimension from the final optimization result obtained in Step 6. Subsequently, AGCRTN is reconstructed based on these optimal hyperparameters identified by WOA.

Step 8: The reconstructed AGCRTN is trained using the data from the training set, and its performance is evaluated by making predictions on the test set.

4. Experiment

4.1. Datasets

We evaluated the performance of the WOA-AGCRTN model using four real traffic datasets, namely PEMS03, PEMS04, PEMS07, and PEMS08, which were released in California, USA. These datasets were collected by the Caltrans Performance Measurement System (PEMS) [43] at a frequency of every 30 s, and it captured the real-time traffic flow on California highways. The collected data were aggregated into 5 min windows, which we considered as time steps. Each hour consists of 12 time steps. Our objective was to utilize the historical data from one hour to predict the data for the next hour. Detailed information about the datasets is provided in Table 1.

Table 1. Description of the dataset.

Datasets	Nodes	Edges	Time Steps	Time Range	Missing Rate
PEMS03	358	547	26,208	1/9/2018–11/30/2018	0.672%
PEMS04	307	340	16,992	1/1/2018–2/28/2018	3.182%
PEMS07	883	866	28,224	5/1/2017–8/31/2017	0.452%
PEMS08	170	295	17,856	7/1/2016–8/31/2016	0.696%

4.2. Baseline Methods

We compared WOA-AGCRTN with 13 baseline methods, which are as follows.

(1) **Vector Autoregression (VAR)** [4]: VAR is a temporal model that captures the temporal correlation of traffic series.

(2) **Support Vector Regression (SVR)** [6]: SVR is a machine learning method that employs support vector machines for traffic sequence regression. Unlike the traditional statistical methods discussed earlier, SVR addresses the limitations associated with stability assumptions and incorporates the capability to model spatial and long-term time dependence.

(3) **Long Short-Term Memory (LSTM)** [8]: LSTM is a neural network model that can efficiently capture the temporal correlation of time series.

(4) **Temporal Convolutional Network (TCN)** [44]: TCN utilizes stacked causal convolutions to effectively capture the temporal correlations in time series data.

(5) **Diffusion Convolution Recurrent Neural Network (DCRNN)** [14]: DCRNN integrates the encoder–decoder architecture of diffusion GCNs and GRUs to predict traffic sequences.

(6) **Spatio-Temporal Graph Convolution Network (STGCN)** [18]: STGCN combines GCNs and temporal convolutions to capture spatial and temporal correlations, respectively.

(7) **Attention-Based Spatio-Temporal Graph Convolutional Network (ASTGCN)** [19]: ASTGCN introduces a spatio-temporal attention mechanism to capture the spatio-temporal correlations of sequences.

(8) **Spatio-Temporal Synchronous Graph Convolutional Network (STSGCN)** [20]: STSGCN constructs multiple local spatio-temporal graphs to capture complex local spatio-temporal correlations effectively. Additionally, modules with varying time periods are designed to capture the heterogeneity within the local spatio-temporal graphs.

(9) **Spatio-Temporal Fusion Graph Neural Network (STFGNN)** [23]: STFGNN incorporates a new fusion operation to learn hidden dependencies from spatial and temporal graphs.

(10) **Spatio-Temporal Graph ODE Networks (STGODE)** [32]: STGODE for Traffic Flow Forecasting utilizes a continuous graph neural network for multivariate time series traffic forecasting.

(11) **Time Zigzags at Graph Convolutional Networks (Z-GCNETs)** [45]: Z-GCNETs introduces the concept of zigzag persistence to time-aware graph convolutional networks for time series forecasting.

(12) **Adaptive Graph Convolutional Recurrent Network (AGCRN)** [24]: AGCRN leverages the learnable node embeddings in graph convolution and integrates GRUs for traffic prediction.

(13) **Dynamic Spatio-Temporal Aware Graph Neural Network (DSTAGNN)** [31]: DSTAGNN designs a new temporal attention module and gated convolution module to capture the temporal dynamic information in traffic data.

4.3. Experiment Settings

The datasets utilized in the experiments were split into training, validation, and test sets following a ratio of 6:2:2. The historical data from the past 1 h, consisting of 12 consecutive time steps, were employed to forecast the traffic flow for the subsequent 1 h, comprising 12 consecutive time steps. The experiments with WOA-AGCRTN were repeated 10 times for each dataset, and the final results were obtained by averaging the outcomes across the 10 experiments. The model experiments were conducted in a computer environment equipped with an 11th Gen Intel(R) Core (TM) i7-11700 @ 2.50 GHz CPU and NVIDIA GeForce RTX3080Ti graphics card. We used WOA to search for the best combination of parameters in the structure of the AGCRTN network model.

- In the first stage, we determine the parameter combinations to search for the optimal number of layers and neural units in the GRU hidden layers, the number of transformer layers, and the number of heads in the multi-headed attention mechanism, as well as the parameters of learning rate and learning rate decay within the structure of the AGCRTN network model. After conducting several experiments, we determined the range of search parameter combinations for the PEMS04 and PEMS08 datasets as [1, 20, 1, 1, 0.2, 0.002] to [2, 90, 6, 8, 0.6, 0.006]. Due to the relatively large size of PEMS07 and equipment limitations, the range for the PEMS07 dataset was set as [1, 20, 1, 1, 0.2, 0.002] to [1, 60, 2, 4, 0.6, 0.006]. For the PEMS03 dataset, the range of search parameter combinations was determined as [1, 20, 1, 1, 0.2, 0.002] to [2, 78, 6, 8, 0.6, 0.006]. The number of epochs for all four datasets was set to 200, and 15 early stopping mechanisms were employed for certain parameters. The batch size was set to 64 for PEMS03, PEMS04, and PEMS08, and to 32 for PEMS07. The embedding matrix dimension was set to 10 for the PEMS03, PEMS04, and PEMS07 datasets, thereby following the parameters of the AGCRN model; meanwhile, for the PEMS08 dataset, it was set to 2. WOA-AGCRTN was trained using the Adam optimizer with a decaying learning rate, and the L1 loss function was employed.
- In the second stage, the WOA is employed to search for the optimal combination of parameters within the network structure search space, with the L1 loss on the validation set serving as the fitness function. The performance of the obtained optimal network structure was assessed based on three evaluation metrics: the mean absolute error (MAE), the root mean square error (RMSE), and the mean absolute percentage error (MAPE). Assume that $X_n^i \in R^{N\times 1}$ is the real traffic flow data for all nodes at time

step i, $Y_n^i \in R^{N\times 1}$ is the predicted value, and N is the number of samples observed. These indicators are defined as Equations (24)–(26):

$$MAE = \frac{1}{N}\sum_{I=1}^{N}\left|X_n^i - Y_n^i\right|, \tag{24}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_n^i - Y_n^i\right)^2}, \tag{25}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{X_n^i - Y_n^i}{X_n^i}\right|. \tag{26}$$

The MAE, RMSE, and MAPE measures indicate better prediction performance with smaller values.

4.4. Experiment Results and Analysis

Table 2 presents the specific parameter values of WOA-AGCRTN across four datasets. It is evident that the optimal parameters for AGCRTN vary among these datasets, thereby underscoring the necessity of employing WOA to optimize AGCRTN. The prediction results of WOA-AGCRTN and the 13 comparative models for four real traffic flow datasets, which were forecasted one hour ahead (12-time steps), are presented in Table 3. The findings show that our proposed WOA-AGCRTN method surpassed all other models across all four datasets in terms of all measures, thus indicating its superior prediction performance. Specifically, the WOA-AGCRTN method exhibited significant improvements in its results when applied to the PEMS03, PEMS04, PEMS07, and PEMS08 datasets, as shown in Table 3. Compared to DSTAGNN, WOA-AGCRTN exhibited a similar performance in all metrics except for the MAPE index. In PEMS03, WOA-AGCRTN outperformed the current best baseline approach by 2.6% and 1.4% in MAE and RMSE, respectively. Similarly, in PEMS04, WOA-AGCRTN achieved improvements of 1.6% and 1.4% in MAE and RMSE, respectively, compared to the current best baseline approach. Furthermore, in PEMS07, WOA-AGCRTN surpassed the current best baseline approach by 4.1% and 2.2% in MAE and RMSE, respectively. For PEMS08, WOA-AGCRTN outperformed the current best baseline approach by 3.4% and 1.6% in MAE and RMSE, respectively. WOA-AGCRTN leveraged GRU and transformer architectures to effectively capture the temporal dependencies at both long and short ranges. Finding the ideal set of network structure parameters using WOA as a neural network architecture search approach eliminates the need for experience-dependent manual adjustment.

PEMS07 requires excessive processing resources due to its large size, while PEMS03 suffers from a significant number of missing values, thus impacting result accuracy. To investigate these issues, we focused on analyzing PEMS04 and PEMS08. The variance in MAE, MAPE, and RMSE for the models in Table 2 across PEMS04 and PEMS08 is depicted in Figure 5. The difficulty of prediction increases with forecast time steps are reflected in the rising MAE, MAPE, and RMSE scores. WOA-AGCRTN outperformed DSTAGNN, particularly in terms of leveraging transformer's multi-headed attention mechanism. Although WOA-AGCRTN's performance closely matched DSTAGNN's MAPE on the PEMSE08 dataset, it notably exceled at time steps 10, 11, and 12. Additionally, WOA-AGCRTN's prediction performance surpassed DSTAGNN, thus indicating its ability to effectively model global temporal relationships across sequences by leveraging GRUs and the transformer algorithm to capture the correlations between long-term and short-term dimensions. Thus, WOA-AGCRTN demonstrates a superior capability in long-term prediction.

Table 2. The parameters of WOA-AGCRTN on four traffic datasets.

	PEMS03	**PEMS04**	**PEMS07**	**PEMS08**
Parameters	Values	Values	Values	Values
Lr-init	0.0060	0.0021	0.0020	0.0060
Lr-decay-rate	0.2417	0.3315	0.6000	0.5689
Rnn-num-layers	1	1	1	2
Rnn-num-untis	30	65	41	69
Transformer-num-layers	3	6	2	2
Transformer-num-heads	4	4	4	6

Table 3. Comparison of WOA-AGCRTN against the baselines on the four traffic datasets.

Methods	**PEMS03**			**PEMS04**			**PEMS07**			**PEMS08**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
VAR	23.65	38.26	24.51%	24.54	38.61	17.24%	50.22	75.63	32.22%	19.19	29.81	27.88%
SVR	21.97	35.29	21.51%	28.7	44.56	19.20%	32.49	50.22	14.26%	23.25	36.16	14.64%
LSTM	21.33	35.11	23.33%	26.77	40.65	18.23%	29.98	45.94	13.20%	23.09	35.17	14.99%
TCN	19.32	33.55	19.93%	23.22	37.26	15.59%	32.72	42.23	14.26%	22.72	35.79	14.03%
DCRNN	17.99	30.31	18.34%	21.22	33.44	14.17%	25.22	38.61	11.82%	16.82	26.36	10.92%
STGCN	17.55	30.42	17.34%	21.16	34.89	13.83%	25.33	39.34	11.21%	17.50	27.09	11.29%
ASTGCN	17.34	29.66	17.24%	22.93	35.22	16.56%	24.05	37.97	10.92%	18.25	28.06	11.64%
STSGCN	17.48	29.21	16.78%	21.19	33.65	13.90%	24.26	39.03	10.21%	17.13	26.80	10.96%
AGCRN	* 15.98	* 28.25	* 15.23%	* 19.88	* 32.27	* 13.03%	* 22.26	* 36.47	* 9.16%	* 15.97	* 25.25	* 10.13%
STFGNN	16.77	28.34	16.30%	19.83	31.88	13.02%	22.07	35.80	9.21%	16.64	26.22	10.60%
STGODE	16.50	27.84	16.69%	20.84	32.82	13.77%	22.59	37.54	10.14%	16.81	25.97	10.62%
Z-GCNETs	* 16.64	* 28.15	* 16.39%	* 19.67	* 31.86	* 12.91%	* 21.79	* 35.15	* 9.27%	* 16.03	* 25.28	* 10.39%
DSTAGNN	* 15.57	* 27.21	* 14.68%	* 19.44	* 31.83	* 12.82%	* 21.46	* 34.82	* 9.12%	* 15.81	* 25.08	* 9.98%
WOA-AGCRTN	**15.17**	**26.83**	**14.48%**	**19.13**	**31.37**	**12.77%**	**20.57**	**34.06**	**8.74%**	**15.27**	**24.67**	**9.96%**

* denotes re-implementation or re-training. The bold in the table represents the optimal solution.

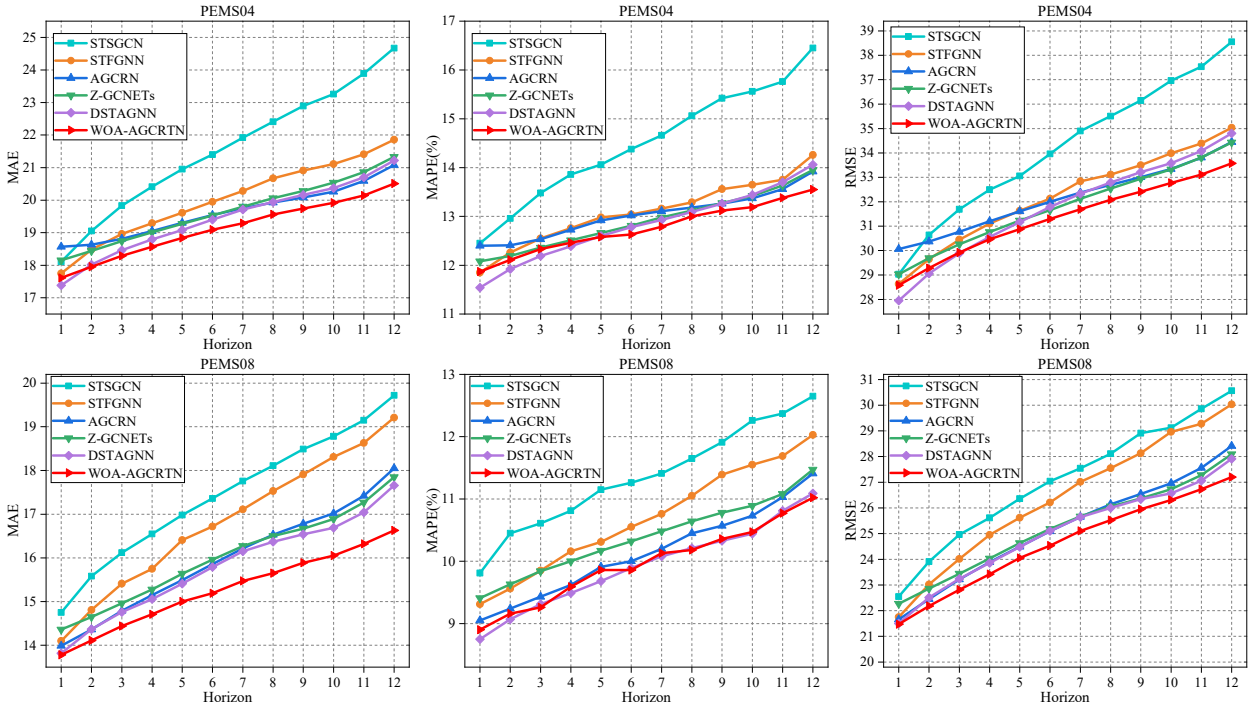

Figure 5. Comparison of the prediction performance at the 12 time steps on the PEMS04 and PEMS08 datasets.

To visually assess the prediction performance of our method, we created a plot that compares the predicted values of our approach (WOA-AGCRTN) with the ground truth for multiple stations/nodes throughout a given day, as shown in Figure 6. The WOA-AGCRTN effectively captures the real-world traffic variations across different time periods. This high accuracy is attributed to our precise modeling of long-range spatio-temporal dependencies and the optimal design of the network structure. This demonstrates that our method achieves accurate predictions even in challenging traffic scenarios, such as during peak hours or when significant fluctuations occur. Specifically, our method effectively captures changes in traffic patterns, as shown in Figure 6a,b for the PEMS04 dataset, where traffic trends between 4–6 and 16–20, and 4–6 and 18–20 are also observed. The predicted values in the PEMS08 dataset closely align with the actual changes in traffic trends.

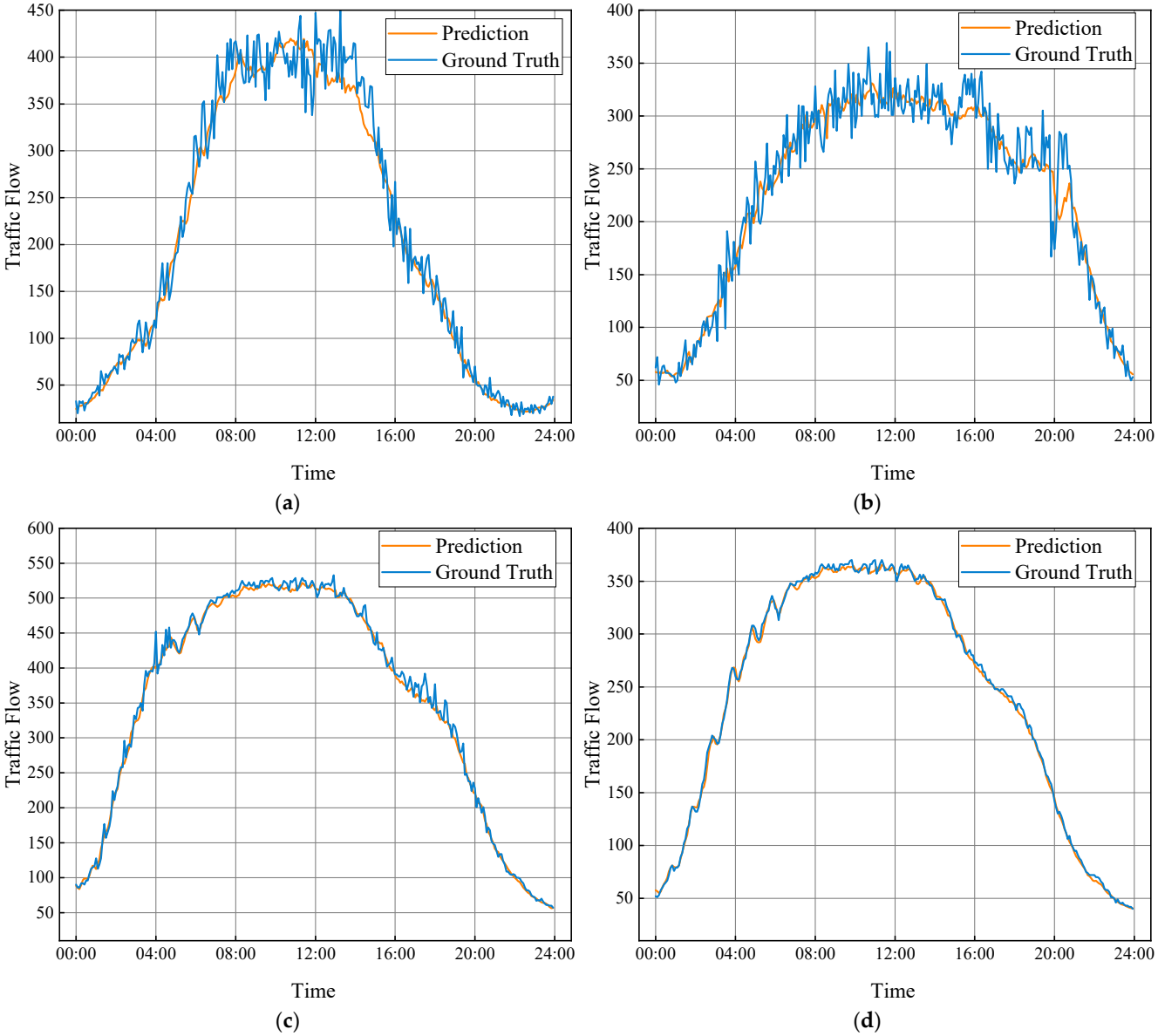

Figure 6. Traffic flow prediction at different time points. (**a**) Node 25 in PEMS04; (**b**) Node 125 in PEMS04; (**c**) Node 28 in PEMS08; and (**d**) Node 135 in PEMS08.

4.5. Ablation Study

To more accurately evaluate the transformer algorithm and WOA's performance, we utilized the PEMS04 and PEMS08 datasets, which consist of freeway traffic flow data from

the Los Angeles area. These datasets offer a long-time span and high temporal resolution, thus providing ample data features for training and evaluating traffic flow prediction algorithms. Consequently, we conducted a comprehensive ablation study specifically on the PEMS04 and PEMS08 datasets. To achieve this, we devised four variations of WOA-AGCRTN and baseline AGCRN for the ablation experiments, as described below.

- AGCRN: This replaces the traditional GCN with NAPL, DAGG, and then integrates the NAPL-DAGG-GCN module with GRU to capture the temporal and spatial correlations.
- AGCRTN: Compared to WOA-AGCTRN, the WOA is not present, and specific parameters are set according to experience.
- WAGCRN: Compared to WOA-AGCRTN, the transformer layer is removed and WOA is used to only optimize the Rnn-num-units, Rnn-num-layers, Lr-init, and Lr-decay-rate mechanisms of the AGCRN module.
- WAGCRN-T: Compared to WOA-AGCRTN, WOA's optimization of the transformer module is removed, i.e., WOA only optimizes the Rnn-num-units, Rnn-num-layers, Lr-init, and Lr-decay-rate mechanisms of the AGCRN module.
- AGCRN-WT: On the basis of WOA-AGCRTN, the optimization of the AGCRN module by WOA is removed, and WOA only optimizes the Transformer-num-layers Transformer-num-layers, Lr-init, and Lr-decay-rate mechanisms of the transformer module.
- WOA-AGCRTN: The WOA-AGCRTN model employs the transformer algorithm and WOA to capture the global dependencies, thus effectively optimizing the model parameters to attain the optimal combination for the network model's performance.

The results of the ablation experiments conducted on the PEMS04 and PEMS08 datasets for the four variants are presented in Figure 7. The following observations can be made.

- The AGCRTN model demonstrated a superior overall performance compared to AGCRN, thereby highlighting the effectiveness of the transformer algorithm in capturing global temporal dependencies. In contrast, GRU was primarily used to capture the short-term temporal dependencies. However, by integrating GRU with the transformer algorithm to model both long- and short-range temporal dimensions, the prediction performance was further improved.
- The WAGCRN model consistently outperformed AGCRN, thus highlighting the necessity of employing WOA for neural network architecture searches.
- The WAGCRN-T model exhibited a superior performance compared to AGCRTN and WAGCRN. By leveraging the search capability of WOA to enhance the parameter training process of AGCRTN and integrating it with the transformer module, the indispensability of both WOA and the transformer modules in the overall model was demonstrated.
- The AGCRN-WT model demonstrated an overall superior performance compared to AGCRTN and WAGCRN. By leveraging the search capability of WOA to enhance the parameter training process of the transformer module, the indispensability of both WOA and the transformer modules for the entire model was demonstrated.
- The overall performance of the WOA-AGCRTN model was optimized compared to WAGCRN-T and AGCRN-WT. This optimization was achieved by utilizing the search capability of WOA to enhance the parameter training process of both AGCRN and the transformer algorithm, thus resulting in an improved prediction accuracy. These results demonstrate the synergistic nature of GRU and the transformer algorithm, and they also underscore the significance of WOA in optimizing the parameters of both GRU and the transformer algorithm for model performance.

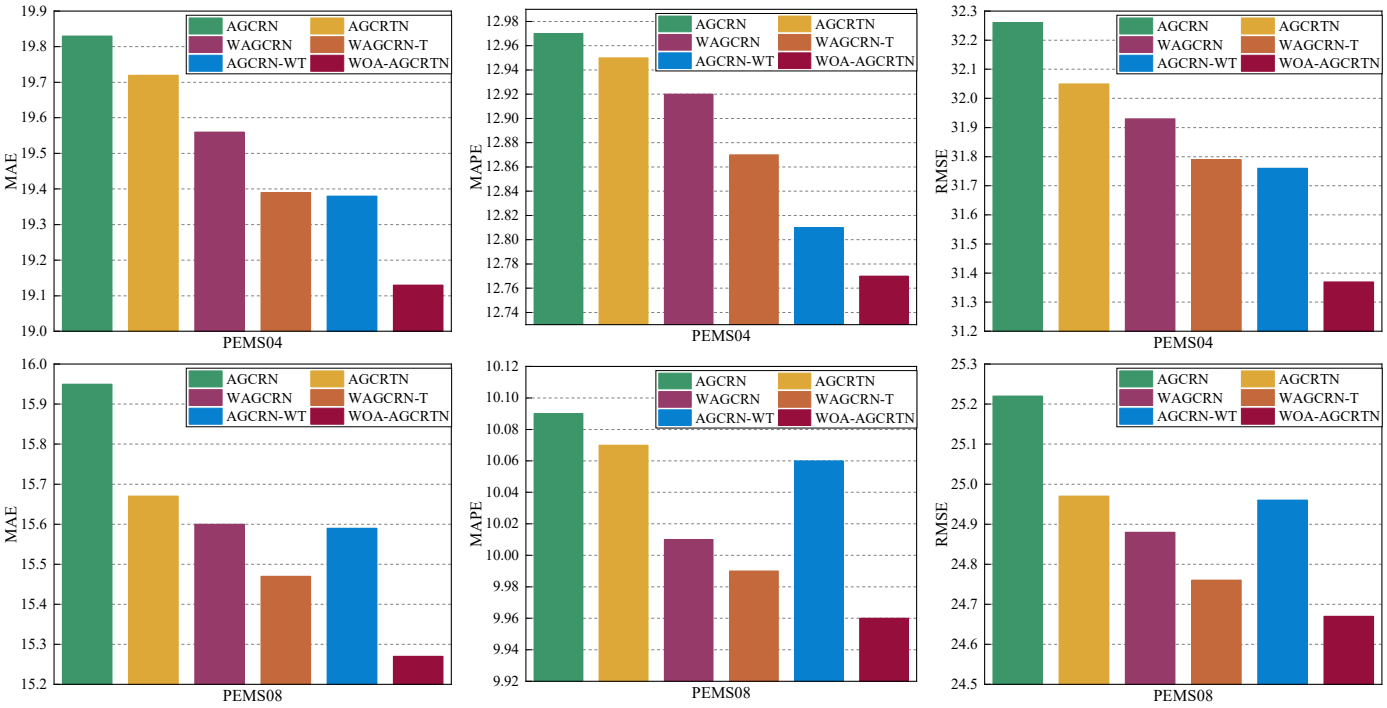

Figure 7. Ablation study on the PEMS04 and PEMS08 datasets.

Overall, our WOA and transformer modules can be deployed independently or in combination, and they consistently enhance predictive performance.

On PEMS04 and PEMS08, for these three variants of WOA-AGCRTN when using the WOA search for optimal model performance, we set the same range of search for the same parameters, and the parameters that made the model performance optimal after 10 whale populations and 10 iterations of search are shown in Tables 4 and 5. The optimized parameter settings are as follows: (8 significant digits are used for experiments, while 4 significant digits are provided for readability in papers). For the PEMS04 dataset, the WOA-AGCRTN model was configured with 1 GRU layer, 65 neural units, a learning rate of 0.0021, and a learning decay rate of 0.3315, as well as 6 layers in the Transformer and 4 in the Transformer. These parameter settings aim to minimize the loss value of WOA-AGCRTN on the validation set. For the PEMS08 dataset, the WOA-AGCRTN model was configured with 2 GRU layers, 69 neural units, a learning rate of 0.0060, and a learning decay rate of 0.5689, as well as 2 layers and 6 attention heads in the transformer algorithm. These parameter settings aimed to minimize the loss value of WOA-AGCRTN on the validation set. The parameter combinations after a WOA search vary across different datasets, as evident from Tables 4 and 5. This suggests that parameters should not be set universally but rather analyzed and tailored to individual datasets to search for optimal combinations. Such an approach can significantly enhance prediction accuracy. The analysis of Tables 4 and 5 reveal that the Rnn-num-layers, Rnn-num-units, Transformer-num-layers, and Transformer-num-heads mechanisms were not maximized. This observation further highlights the effectiveness of the WOA optimization algorithm in conducting a hyperparameter search. In conclusion, our WOA-AGCRTN model achieves the lowest loss value on the validation set, thus indicating the highest average prediction accuracy at 12 time steps.

Table 4. The parameters of the ablation experiment modules on the PEMS04 dataset.

		WAGCRN	WAGCRN-T	AGCRN-WT	**WOA-AGCRTN**
	Parameters	Values	Values	Values	Values
PEMS04	Lr-init	0.0049	0.0036	0.0027	0.0021
	Lr-decay-rate	0.5976	0.4026	0.2204	0.3315
	Rnn-num-layers	2	1	-	1
	Rnn-num-untis	55	75	-	65
	Transformer-num-layers	-	-	4	6
	Transformer-num-heads	-	-	2	4
	Best loss	19.55	19.10	19.17	18.93

Table 5. The parameters of the ablation experiment modules on the PEMS08 dataset.

		WAGCRN	WAGCRN-T	AGCRN-WT	**WOA-AGCRTN**
	Parameters	Values	Values	Values	Values
PEMS08	Lr-init	0.0027	0.0039	0.0026	0.0060
	Lr-decay-rate	0.3004	0.2659	0.3895	0.5689
	Rnn-num-layers	1	2	-	2
	Rnn-num-untis	85	76	-	69
	Transformer-num-layers	-	-	5	2
	Transformer-num-heads	-	-	3	6
	Best loss	16.01	15.83	15.96	15.73

4.6. WOA Optimizes the Iterations of the Different Modules

The optimization process of each module in AGCTRN was visualized using WOA on the PEMS04 and PEMS08 datasets, as presented in Figure 8. The number of whale populations was set to 10 on both datasets and the number of iterations was 10. The fitness value corresponds to the minimal loss value on the validation set in each iteration, where the L1 loss function (MAE) is utilized.

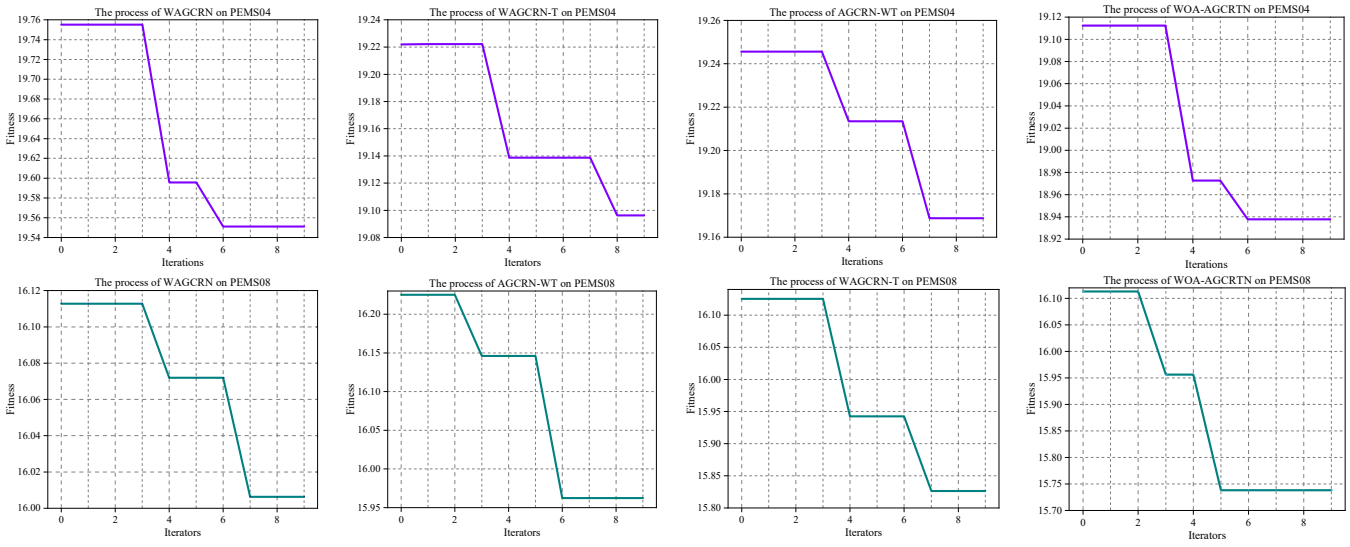

Figure 8. The performance of WOA in independently optimizing the different modules of AGCTRN on the PEMS04 and PEMS08 datasets.

WOA performs an optimization search based on the loss value of the validation set at each iteration of the AGCRTN. A smaller loss value indicates a greater prediction accuracy, thus demonstrating that the network model improved during the optimization process. Several experimental modules for both datasets were optimized, as depicted in Figure 8, and it demonstrated a faster discovery of the optimal parameter combination, thus leading to superior model performance. The majority of optimal parameter combinations were discovered by the sixth iteration. Our proposed model, WOA-AGCRTN, exhibited the lowest fitness value on both datasets, thus indicating the best overall performance among the network models.

5. Conclusions

This paper introduces a novel deep learning framework for traffic flow prediction. Firstly, we propose an AGCRN that integrates the transformer algorithm to effectively capture long-range temporal correlations. Secondly, the whale optimization algorithm (WOA) is employed to automatically design a WOA-AGCRTN network structure that achieves optimal performance with limited computational resources. Our network framework's efficacy and superiority for resolving traffic flow prediction problems are demonstrated by the results from the experiments on four real datasets. Specifically, in the PEMS03 dataset, it achieved a 2.6% improvement in MAE and a 1.4% improvement in RMSE. In the PEMS04 dataset, the improvements were 1.6% in MAE and 1.4% in RMSE, while the MAPE remained essentially the same as the best baseline. For the PEMS07 dataset, our approach demonstrated a 4.1% improvement in MAE and a 2.2% improvement in RMSE. Moreover, on the PEMS08 dataset, it surpassed the current best baseline approach by achieving a 3.4% improvement in MAE and a 1.6% improvement in RMSE. The experimental results demonstrate that the WOA-AGCRTN model achieved a good performance in traffic flow prediction across four public datasets.

However, using optimization methods to optimize the structure and hyperparameters of neural networks is a very computationally resource intensive task. It is often difficult to optimize a set of structures and hyperparameters for neural networks that perform well on larger datasets. In addition, for any neural network model, the optimal structure and hyperparameters will vary from dataset to dataset. We do not have enough computational resources to search for the optimal parameters and structure of the model for a new dataset. Therefore, we often need to re-optimize the model whenever we train on a new dataset.

Finally, traffic networks exhibit distinct structural and dynamical properties that differentiate them from other types of networks, such as social and biological networks. Therefore, in future research, the plan is to conduct a detailed analysis of the characteristics and dynamic correlations of various networks. We aim to leverage the strengths of the WOA-AGCRTN model in temporal and spatial domains and to explore its applicability in other tasks related to network structure analysis and time prediction.

Author Contributions: Conceptualization, Y.W. and C.Z.; Methodology, Y.W. and C.Z.; Software, Y.W.; Formal analysis, Y.W.; Investigation, S.W.; Data curation, W.S.; Writing—original draft, Y.W.; Writing—review & editing, Y.W.; Supervision, C.Z., Y.S. and X.Z.; Funding acquisition, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for the Central Universities of China under grant PA2023IISLO092, which was funded in part by the Program for Scientific Research Innovation Team in Colleges and Universities of Anhui Province under grant 2022AHO10095. It was also supported by the Universities Natural Science Research Project of Anhui Provincial under grant KJ2021ZDO118 and in part by the Anhui Provincial University Outstanding Talent Cultivation Project under grant gxgnfx2020117.

Data Availability Statement: Data available in a publicly accessible repository.

Conflicts of Interest: Chen Zhang was employed by Guochuang Software Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. Guochuang Software Co., Ltd.

had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

$G = (V, E, A)$	Graph with a set of vertices (V), edges (E), and an adjacency matrix (A)
N	Traffic sensor nodes (observation points)
$X^{(t+T)}$	The observed value at the time step t
$C(3.1)$	The number of feature channels
$\mathcal{F}$	A function of learning historical traffic flow sequence information
E_G	The node-embedding matrix
W_G	The weight pool
E_A	The learnable node embedding
$h_t[i,:]$	The output of GRU
Q, K, V	The queries, keys, and values of all nodes
d_k,d_v	Queries of dimension, and the keys and values of dimension
W^Q,W^K,W^V	The projection matrices to be learned
$h_{out}^{v_i}$	The input for the prediction layer
$A, C(3.2.3)$	The ratio in WOA
a	linearly decreases from 2 to 0
r_1, r_2	Random vector
D	The distance between $X^*(t)$ and $X(t)$ after scaling by C
$X^*(t)$	The leading whale in the WOA
$X(t)$	The whale requiring position update
$X(t+1)$	The whale's updated position
D'	The distance between $X^*(t)$ and $X(t)$
b	A constant typically set to 1
l	A randomly generated number in the range of 0 to 1
$X^{rand}(t)$	Randomly chosen to guide the position update of $X(t)$

References

1. Jiang, W.; Luo, J. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* **2022**, *207*, 117921. [CrossRef]
2. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]
3. Shahriari, S.; Ghasri, M.; Sisson, S.A.; Rashidi, T. Ensemble of ARIMA: Combining parametric and bootstrapping technique for traffic flow prediction. *Transp. A Transp. Sci.* **2020**, *16*, 1552–1573. [CrossRef]
4. Lu, Z.; Zhou, C.; Wu, J.; Jiang, H.; Cui, S. Integrating granger causality and vector auto-regression for traffic prediction of large-scale WLANs. *KSII Trans. Internet Inf. Syst. (TIIS)* **2016**, *10*, 136–151.
5. Wan Ahmad, W.K.A.; Ahmad, S. Arima model and exponential smoothing method: A comparison. *AIP Conf. Proc.* **2013**, *1522*, 1312–1321.
6. Jeong, Y.-S.; Byon, Y.-J.; Castro-Neto, M.M.; Easa, S.M. Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1700–1707. [CrossRef]
7. Van Lint, J.W.C.; Van Hinsbergen, C. Short-term traffic and travel time prediction models. *Artif. Intell. Appl. Crit. Transp. Issues* **2012**, *22*, 22–41.
8. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]
9. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 324–328.
10. Zhang, J.; Zheng, Y.; Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
11. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–August 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 2, pp. 729–734.
12. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2013**, arXiv:1312.6203.
13. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.

14. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
15. Chen, W.; Chen, L.; Xie, Y.; Cao, W.; Gao, Y.; Feng, X. Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3529–3536.
16. Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1720–1730.
17. 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.* **2019**, *21*, 4883–4894. [CrossRef]
18. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
19. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January 2019; Volume 33, pp. 922–929.
20. 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.
21. Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; Jia, C.; Yu, J. Traffic flow prediction via spatial temporal graph neural network. In Proceedings of the Web Conference 2020, New York, NY, USA, 20–24 April 2020; pp. 1082–1092.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems 30, Proceedings of the NIPS 2017, Long Beach, CA, USA, 4–9 December 2017*; MIT Press: Cambridge, MA, USA, 2017.
23. Li, M.; Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2021; Volume 35, pp. 4189–4196.
24. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems 33, Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada 6–12 December 2020*; MIT Press: Cambridge, MA, USA, 2020; pp. 17804–17815.
25. Wu, J.; Fu, J.; Ji, H.; Liu, L. Graph convolutional dynamic recurrent network with attention for traffic forecasting. *Appl. Intell.* **2023**, *53*, 22002–22016. [CrossRef]
26. Huang, X.; Jiang, Y.; Tang, J. MAPredRNN: Multi-attention predictive RNN for traffic flow prediction by dynamic spatio-temporal data fusion. *Appl. Intell.* **2023**, *53*, 19372–19383. [CrossRef]
27. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
28. Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; Li, P. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; PMLR: Baltimore, MD, USA, 2022; pp. 11906–11917.
29. Fang, Z.; Long, Q.; Song, G.; Xie, K. Spatial-temporal graph ode networks for traffic flow forecasting. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 364–373.
30. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29, Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016*; MIT Press: Cambridge, MA, USA, 2016.
31. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
32. Micheli, A. Neural network for graphs: A contextual constructive approach. *IEEE Trans. Neural Netw.* **2009**, *20*, 498–511. [CrossRef] [PubMed]
33. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30, Proceedings of the NIPS 2017, Long Beach, CA, USA, 4–9 December 2017*; MIT Press: Cambridge, MA, USA, 2017.
34. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
35. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948.
36. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
37. Mirjalili, S.; Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks: Theory and Applications*; Springer: Cham, Switzerland, 2019; pp. 43–55.
38. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
39. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
40. Xu, X.; Liu, C.; Zhao, Y.; Lv, X. Short-term traffic flow prediction based on whale optimization algorithm optimized BiLSTM_Attention. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6782. [CrossRef]
41. Pham, Q.-V.; Mirjalili, S.; Kumar, N.; Alazab, M.; Hwang, W.-J. Whale optimization algorithm with applications to resource allocation in wireless networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4285–4297. [CrossRef]

42. Bhesdadiya, R.; Jangir, P.; Jangir, N.; Trivedi, I.N.; Ladumor, D. Training multi-layer perceptron in neural network using whale optimization algorithm. *Indian J. Sci. Technol.* **2016**, *9*, 28–36.
43. Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; Jia, Z. Freeway performance measurement system: Mining loop detector data. *Transp. Res. Rec.* **2001**, *1748*, 96–102. [CrossRef]
44. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks for action segmentation and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 156–165.
45. Chen, Y.; Segovia, I.; Gel, Y.R. Z-GCNETs: Time zigzags at graph convolutional networks for time series forecasting. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR: Baltimore, MD, USA, 2021; pp. 1684–1694.