An Attention-Driven Spatio-Temporal Deep Hybrid Neural Networks for Traffic Flow Prediction in Transportation Systems

Ahmad Ali, Inam Ullah, *Member, IEEE*, Shabir Ahmad, *Senior Member, IEEE*, Zongze Wu, *Member, IEEE*, Jianqiang Li, *Senior Member, IEEE*, and Xiaoshan Bai, *Member, IEEE*

***Abstract*— In the context of rapidly growing city road networks, understanding complex traffic patterns and implementing effective safety monitoring through advanced Transportation Cyber-Physical Systems (T-CPS) has become increasingly challenging. This involves understanding spatial relationships and non-linear temporal associations. Accurately predicting traffic in such scenarios, particularly for long-term sequences, is challenging due to the complexity of the data. Traditional ways of predicting traffic flow use a single fixed graph structure based on location. This structure does not consider possible correlations and cannot fully capture long-term temporal relationships among traffic flow data, thereby limiting the system ability to ensure safety and reliability. To address this challenge, we propose a novel traffic prediction framework called Attention-based Spatio-temporal Multi-scale Graph Convolutional Recurrent Network (ASTMGCNet). This study introduces a novel framework designed to improve prediction accuracy in dynamic urban traffic systems by effectively capturing complex spatio-temporal correlations through multi-scale feature extraction and attention mechanisms. ASTMGCNet records changing features of space and time by combining Gated Recurrent Units (GRU) and Graph Convolutional Networks (GCN). Its design incorporates multi-scale feature extraction and dual attention mechanisms, effectively capturing informative patterns at different levels of detail. This strategic design allows ASTMGCNet to effectively capture complex spatio-temporal correlations within traffic sequences, enhancing prediction accuracy. We have tested this method on two different real-world datasets and found that ASTMGCNet predicts significantly better than other methods, demonstrating its potential to advance traffic flow prediction and improve safety and reliability in T-CPS applications.**

***Index Terms*— Intelligent transportation systems, traffic prediction, transportation cyber-physical systems, attention mechanism, GCN, safety monitoring.**

I. Introduction

With the increasing number of cars on roads worldwide, managing traffic jams and ensuring effective safety monitoring through Transportation Cyber-Physical Systems (T-CPS) in cities has become a significant challenge. T-CPS relies heavily on safety and dependability. Our model effectively captures spatio-temporal interdependence, enabling traffic predictions under various scenarios. By predicting and preventing traffic accidents and sending early notifications to avoid accidents, the model can be connected with real-time monitoring systems to improve safety. This is crucial for Intelligent Transportation Systems (ITS) [1], [2] [3], which manage road traffic and help drivers accurately predict traffic, allowing people to commute more efficiently. However, there are some major challenges with traffic prediction. First, the intricate spatio-temporal correlations between the vast transportation network pose difficult problems for traffic prediction. Second, many factors, such as the weather, can disrupt traffic. These events create abrupt changes in traffic flow. Even though the ASTMGCNet model is good at capturing common spatio-temporal patterns, these erratic events are frequently underrepresented in the training set, thus it might have trouble handling them. For example, more people drive their cars when it rains instead of using public transportation. Also, unexpected events, such as accidents or major events, can suddenly change traffic flow. However, because we can collect data from different devices that monitor traffic, researchers can now use this information to understand and predict traffic better. This is possible because computers are now very powerful and can quickly process information. With this new approach, scientists can build intelligent models looking at real-world data to determine how traffic works. They can better understand how traffic behaves in various situations and make predictions that help manage traffic more effectively.

Traffic data exhibit strong dynamic correlations in both spatio-temporal dimensions [1]. Consequently, accurate

Received 26 August 2024; revised 24 November 2024 and 20 January 2025; accepted 7 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62373255 and Grant 62173234; in part by the Natural Science Foundation of Guangdong Province under Grant 2024A1515011204; in part by Shenzhen Natural Science Fund through the Stable Support Plan Program under Grant 20220809175803001; in part by the Open Fund of National Engineering Laboratory for Big Data System Computing Technology under Grant SZU-BDSC-OF2024-15; and in part by the Researchers Supporting Project, King Saud University, Riyadh, Saudi Arabia, under Grant RSP2024R32. The Associate Editor for this article was C. Chakraborty. *(Corresponding author: Xiaoshan Bai.)*

Ahmad Ali and Zongze Wu are with the College of Mechatronics and Control Engineering and the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.

Inam Ullah is with the Department of Computer Engineering, Gachon University, Seongnam 13120, Republic of Korea, and also with the Department of Artificial Intelligence, Tashkent State University of Economics, Tashkent 100066, Uzbekistan.

Shabir Ahmad is with the Center of Artificial Intelligence for Medical Instruments, South Korea.

Jianqiang Li is with the National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China.

Xiaoshan Bai is with the College of Mechatronics and Control Engineering, and the National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China (e-mail: baixiaoshan@szu.edu.cn).

Digital Object Identifier 10.1109/TITS.2025.3540852

traffic prediction conditions depend on efficiently capturing these complex and non-linear spatio-temporal relationships. To achieve the best predictive model articulation, it remains crucial to carefully integrate and model these changing spatio-temporal dependencies within the context of T-CPS, while incorporating safety monitoring measures. Energy efficiency and computational cost are essential for practical implementation, especially for smart city infrastructures. In this work, we present the computational efficiency of the model by giving the total training times and inference times on various hardware configurations. The energy usage for model training and inference is also tracked to guarantee peak performance. ASTMGCNet can be used for large-scale energy-efficient applications since techniques like hardware-efficient topologies and lightweight model versions have been investigated to reduce computational load.

Predetermined adjacency graphs have traditionally used one of two strategies: 1) distance functions, in which the structure of the graph is determined based on geographic distances among diverse nodes [4], [5]; or 2) similarity-based approaches, where node proximity is assessed by evaluating the resemblance of node attributes, such as Points of Interest (PoI) data as discussed in [6] and [7], or by comparing the flow sequences directly as shown in [8], offer an intuitive perspective. However, it is important to note that these approaches come with intrinsic limitations when uncovering concealed spatial dependencies. Predefined graph structures do not provide a comprehensive understanding of spatial relationships and do not align directly with traffic prediction, potentially leading to substantial biases. Furthermore, the generalizability of these models is limited to contexts with appropriate domain knowledge, thereby hindering their application to diverse domains. This, in turn, makes existing GCN-based models ineffective in scenarios beyond the boundaries of existing knowledge. T-CPS first and most severe issue is the difficulty in simultaneously extracting the joint influences of many spatio-temporal dependencies.

Unfortunately, the task of dynamically predicting spatio-temporal patterns within specific areas is a complex challenge that has received limited attention from researchers. Traffic flow is a spatio-temporal graph problem. Each area in Fig. 1 shows the total amount of vehicle inflow and outflow in that area during a given period. Traffic flow estimation utilizes historical data to deduce spatio-temporal patterns, providing insights into future traffic behavior. Regrettably, simultaneously predicting traffic in various areas remains a significant challenge with limited research. Traffic movement is primarily influenced by three key factors: place connectivity, time impact, and external factors. The inflow of region 2 ($r2$) is influenced by the outflow of neighboring regions, such as region 1 ($r1$) and region 3 ($r3$), as well as from more distant regions as shown in Fig. 1.

Similarly, the outflow of $r2$ can influence the inflow of nearby regions. Furthermore, it is worth noting that the inflow of $r2$ may also impact its outflow within the same region. Semantic and spatial are the two categories of spatio-temporal correlations. The semantic relationship illustrates how contextual traffic flow parameters are comparable, while

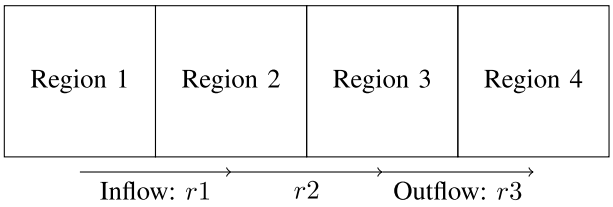


Fig. 1. Traffic flows in a specific region.

the spatial relationship shows how the region is connected. During the morning rush hour, workplaces serve as the primary traffic destination. Consequently, neighboring areas around work locations are likely to experience similar traffic patterns. At the moment, spatial neighbors have a greater impact than semantic neighbors. Few people go to work locations during evening rush hour, with the majority of people likely heading to business districts. Traffic patterns in comparable regions exhibit similar characteristics. Currently, semantic neighbors have a stronger influence than spatial neighbors. The second issue in T-CPS is the absence of a robust method for extracting multi-scale spatio-temporal information. Because the reference value at a particular time and the integrity of past data fluctuate with time, it cannot treat inputs equally. Accurate traffic prediction requires the effective use of both short- and long-term spatio-temporal correlations.

Traffic forecasting methods originally emerged from statistically driven univariate time series prediction techniques with limited consideration for T-CPS and safety monitoring. Among these techniques are methodologies like the Vector Autoregressive (VAR) [9], Seasonal Autoregressive Integrated Moving Average (SARIMA) [10], Historical Average (HA) [11], and Autoregressive Integrated Moving Average (ARIMA) [12]. Most of these techniques require the assumption of temporal stationarity inherent in each time series. However, it is worth noting that these methods rely heavily on predefined parameters that cannot be derived from the data itself, leading to inherent limitations in effectively capturing the complex spatio-temporal correlations present in the dataset.

In order to solve the above challenges, we present a network framework capable of dynamically producing adaptable adjacency matrices. Unlike previous efforts that exclusively address unchanging features, our proposed network utilizes an innovative, flexible generation graph framework that accommodates the dynamic evolution of traffic flows. Our method does not use standard graph structures; instead, it uses a flexible parameter learning strategy for each node in the graph convolution process, considering both time and space. This integration involves the intelligent incorporation of spatio-temporal mechanisms, effectively capturing extensive long-term dependencies. At the same time, our approach incorporates a multi-scale extraction feature that partitions the input characteristics into four distinct parallel partitions of different dimensions. The main contributions of this paper are highlighted as follows:

- We propose a novel model called Attention-Based Spatio-Temporal Multi-Scale Graph Convolutional Recurrent Network (ASTMGCNet). This network

can carefully include data from changing graphs. This model combines spatio-temporal extraction feature mechanisms at several scales, which makes it better at processing information from a wide range of receptive regions at various hierarchical levels. Furthermore, we establish a traffic flow prediction architecture enriched with attention-based multi-scale modules.

- We present a novel module for multi-scale spatio-temporal extraction features. This module divides the input characteristics into four parallel partitions with different dimensions. This design makes it easier to store complex contextual information and enhances the system robustness to scale changes.
- We demonstrate the effectiveness of our proposed model through extensive experiments conducted on two different real-world traffic datasets. Our model is superior to the prevailing approaches in these empirical evaluations, demonstrating its effectiveness in traffic flow prediction.

The remainder of this paper is organized as follows: In Sec. II, we present the problem formulation. Sec. III outlines the methodology. Sec. IV discusses the experiments and empirical investigations conducted on two real-world traffic datasets. Sec. V provides a comprehensive review of related literature. Finally, Sec. VI presents the concluding remarks.

II. Problem Formulation

In this section, we formally present the traffic prediction problem and provide a mathematical definition of the traffic network concepts. A concise summary of the key notations used throughout this paper is provided in Table I.

The task of predicting traffic flow can thus be viewed as a complex time series multi-scale prediction problem, enhanced by incorporating auxiliary prior knowledge. In general, this involves predicting a weighted graph, typically denoted as $|G| = (S, E, A)$, where S signifies the nodes set indicating the sources of the traffic flow, $|S| = N$ denoting the number of nodes, while $E = (i, j, w_{ij})$ is a set of edges, the edge weight w_{ij} among nodes i and j representing distance, travel time, and $A \in R^{N\times N\times T}$ represents a spatial matrix that characterizes the inter-node similarities, encompassing factors like (e.g., the distance of the road network and POI similarity). This matrix plays a crucial role as foundational information input for the graph convolution process. More specifically, we consider a collection of traffic data consisting of N related univariate time series denoted as $Y = (Y_0, Y_1, \ldots, Y_t, \ldots)$ where each constituent series $Y_t = (y_{1,t}, y_{2,t}, \ldots, y_{i,t}, \ldots, y_{N,t})$ is encapsulated in a vector $S \in R^{N\times 1}$, that enumerates the compilation of N sources at the discrete time instance t. We aim to predict future values within these traffic patterns based on historical observations.

The objective of traffic forecasting is to learn a function, denoted as F, with the capability to predict a future signal tensor θ based on a set of historical signal tensors D and the underlying graph structure G. This function predicts subsequent data spanning time steps τ ahead, and k represents the time step.

TABLE I
Summary of Major Notations

Notations	Description
G	Weighted graph
E	Set of edges
S	Set of nodes
$Y : t$, b_t	Input and output at time t
$A \in R^{N\times N}$	Adjacency matrix
$b \in R^{F}$	Biases
W_n	Reservoir weights
N_n	Node matrix
Q	Query subspace
V	Value subspace
K	Key subspace
$Y(t)$	Predicted traffic flow

$$[Y_{t+1}, \ldots, Y_{t+\tau}] = F_\theta(Y_{t-T+1}, \ldots, Y_t; G) \quad (1)$$

$$Y_{t+k} = F_\theta(Y_{t-T+1}, \ldots, Y_t; G)_k \quad \text{for } k = 1, 2, \ldots, \tau \quad (2)$$

III. Methodology

The proposed ASTMGCNet model is shown in Fig. 2. In this section, we first present the structure of our proposed ASTMGCNet model. We then provide a detailed explanation of each component within this model.

A. Model Framework

The proposed ASTMGCNet model effectively captures intricate spatio-temporal interactions and offers dynamic, real-time predictions, greatly enhancing the safety and dependability of T-CPS. As shown in Fig. 2, our study proposed the ASTMGCNet model. This network combines the principles of the Dynamic Generation Graph Network (DGGN), multi-scale attention, and GRU. The intention is to capture spatio-temporal relationships between nodes within traffic flow sequences. The ASTMGCNet model replaces the conventional MLP layer within the GRU with the DGGN, enabling the detection of node-specific patterns. In addition, the DGGN module inherently identifies spatial dependencies, contributing to a comprehensive understanding of the underlying traffic dynamics. The bottom row of Fig. 2 illustrates the sequential data processing steps from the initial input to the final prediction, highlighting key operations such as loss calculation, activation, feature embedding, and applying a fully connected layer. Compared to current state-of-the-art traffic flow prediction models, the suggested approach, ASTMGCNet, makes significant novel additions. ASTMGCNet combines GCNs with GRUs and dual attention mechanisms to effectively capture both spatio-temporal dependencies in dynamic traffic data, in contrast to traditional models that frequently rely on fixed graph structures or limited temporal representations. The

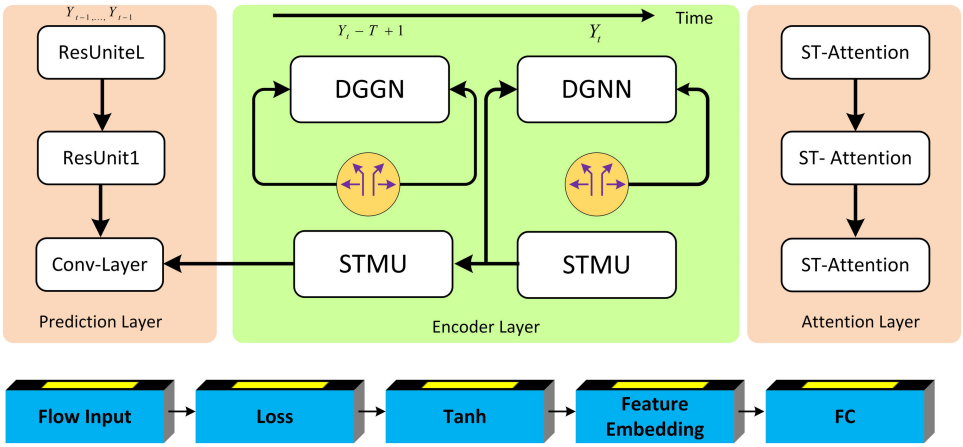


Fig. 2. Framework of the ASTMGCNet model, including prediction, attention, and encoder layers.

multi-scale extraction feature approach is the main innovation. It allows the model to learn patterns at different spatio-temporal resolutions, allowing it to be flexible enough to be applied to complicated and large-scale traffic scenarios. Furthermore, the model may dynamically modify the adjacency matrix in real-time to reflect changing interactions between traffic nodes thanks to dynamic graph formation. Additionally, ASTMGCNet incorporates a spatio-temporal attention mechanism in comparison to other models. The ASTMGCNet can constantly shift its focus on pertinent traffic conditions by learning which elements of the data are most influential at any particular time. This ensures that safety-critical events, including congestion builds or abrupt outages, are given more weight in the prediction process. Early alerts become more precise as a result, and real-time implementation of preventive safety measures is made possible. The ASTMGCNet formulation is as follows:

$$\begin{aligned}
\bar{A} &= \tanh(\mathrm{ReLU}(NN^D)) \\
s_t &= \sigma(\bar{A}[Y:,t,b_t-1]NW_s + Nb_s) \\
n_t &= \sigma(\bar{A}[Y:,t,b_t-1]NW_n + Nb_n) \\
\bar{b_t} &= \mathrm{softmax}(\sigma A[Y:,t,r\odot b_t-1]NW_{\bar{b}} + Nb_{\bar{b}}) \\
b_t &= STMU[s\odot b_t - 1 + (1-s)\odot \bar{b}]
\end{aligned} \tag{3}$$

where $Y:t$ and b_t represent the input and output at given time t. The symbol $\odot$ denotes the concatenation operation, while n and s symbolize the update and reset gates. Each node undergoes an initial random assignment using a learnable node embedding dictionary denoted by N. The parameters NN, W_s, W_n, W_b, b_s, b_n, and b_b are considered as trainable entities in the ASTMGCNet framework. Eq.3 employs notations that depict the temporal dependencies in the data. Specifically, the term $b_t - 1$ refers to the state at the previous time step. This notation is essential for incorporating historical information into the current state calculations, a fundamental aspect of time-series models and Recurrent Neural Networks (RNN). Like the GRU architecture, all of these parameters can be trained end-to-end within ASTMGCNet via back-propagation. By examining eq.3, it becomes clear that ASTMGCNet integrates all embedding matrices as a singular entity denoted by N. This approach differs from the conventional practice of using different node embedding matrices within different DGGNs. This combining of embeddings works as a strong regularization method, ensuring all GCN blocks have the same node embeddings. Consequently, it bolsters the interpretability of our model as a whole.

B. Modeling Dynamic Generation Graph Network Module

In T-CPS, urban traffic flow conditions frequently undergo rapid and significant fluctuations within time intervals. The topological structure of the input adjacency matrix can affect how well GCN extracts features. To address this issue and flexibly respond to input node information, we present a novel framework that combines GRU with GCN. This hybrid model improves prediction accuracy while minimizing parameter usage and has a simpler structure than RNNs. The model uses enhanced gated recurrent units to get multi-scale temporal correlations and stacked fusion graph convolutional modules to get multi-level spatial correlations. This enables the intricate spatio-temporal correlation between traffic flow data to be obtained. By incorporating graph generation for each node, the model captures dynamically the finer details of the road network topology, thus achieving higher granularity in its representation. The mathematical formulations for GCN and GRU are as follows:

$$Z(t) = \mathrm{GCN}(X, A, H(t-1)) \tag{4}$$

$$H(t), Y(t) = GRU(Z(t), H(t-1)) \tag{5}$$

where $Z(t)$ represents the model hidden state at time step t, $H(t-1)$ represents the model hidden state at the previous time step (t-1), X represents the input features, and $H(t)$ represents the updated hidden state at time step t.

Within graph convolutional operations, sharing parameters and biases (b) uniformly across all nodes is common practice. However, such parameter sharing can limit the model ability to extract complex features effectively. This observation is relevant to traffic scenarios where neighboring nodes may manifest distinct traffic patterns due to their unique attributes. Furthermore, we can observe different flow sequences, characterized by opposite patterns, between two non-overlapping

nodes. Given these considerations, the need to identify specific patterns for each node becomes apparent, necessitating the maintenance of a separate parameter space dedicated to each node. In [13], the graph convolution operation is amenable to approximation using the first-order Chebyshev polynomial expansion.

$$Z = b + Y\theta(1 + D^{\frac{-1}{2}} A D^{\frac{-1}{2}}) \tag{6}$$

$$G_{Conv} \approx T_1(L) \cdot Y \cdot \ominus \tag{7}$$

Consider a graph denoted by its adjacency matrix $A \in R^{N\times N}$ and the associated degree matrix D. Additionally, let $\theta \in R^{C\times F}$ and $b \in R^F$ symbolize the learning weights and biases, respectively. Here, Y denotes the input, while Z denotes the output derived from the GCN layer. In the context of isolating a single node i for analysis, the operation performed by the GCN can be construed as effecting a transformation of the features of the node from an initial state, represented as $Y^i \in R^{1\times C}$, to a result in the state denoted as $Z^i \in R^{1\times F}$. Note that G_{Conv} represents the graph convolution operation, $T_1(L)$ denotes the first order Chebyshev polynomial applied to the graph Laplacian matrix L, Y represents the input feature matrix, while $\ominus$ denotes the learnable weight parameters.

The DGGN introduces an automated mechanism that dynamically generates graphs and features in response to the input data encountered at every successive time step. This adaptive approach allows for the better capture of hidden dependencies between nodes. In the DGGN module, individual nodes are given a randomized initialization via a trainable node embedding dictionary denoted as $N \in R^{N\times D}$, where a dedicated row denotes every embedding node's profile in N. Here, D denotes the embedding's vertex dimensionality. The establishment of graph similarity relies on the node affinities, and spatial correlations among node pairs are inferred by the matrix product of N with its transpose, N^T. Throughout the training program, N is subject to automatic updates, enabling the discovery of concealed dependencies among various flow sequences. The result is an adaptive matrix tailored to graph convolutions. Importantly, the adaptive matrix undergoes a normalization process, enhancing its utility and significance.

Our model integrates the DGGN into the traditional GRU architecture. Unlike conventional models that use fixed graph structures, the DGGN dynamically adjusts the graph topology to better capture node-specific patterns and spatial dependencies in traffic data. This approach enhances the model understanding of complex, non-static relationships between traffic nodes. We introduce a multi-scale attention mechanism that simultaneously captures both short-term and long-term dependencies. This mechanism allows the model to focus on relevant spatio-temporal features at different resolutions, which is crucial for accurately predicting traffic flow across various timescales. In contrast to existing methods that use separate node embedding matrices for different graph convolutional networks, our approach consolidates all embedding matrices into a single entity. This consolidation is a robust regularization technique, promoting uniformity and consistency in node embeddings among all blocks of GCN. This uniformity improves the interpretability and generalization capabilities of our model.

$$D^{\frac{-1}{2}} A D^{\frac{-1}{2}} = tanh(ReLU(N \times N^T)) \tag{8}$$

The DGGN uses the model defined in eq.8 to generate the generalization of the term $D^{\frac{-1}{2}} A D^{\frac{-1}{2}}$ to the high-dimensional context of GCN. Hence, mitigating the need for repetitive computations throughout the iterative training process is expressed mathematically in the eq.11.

$$Z = b + Y \ominus (I + tanh(ReLU(N \times N^T))) \tag{9}$$

$$M = I + tanh(ReLU(N \times N^T)) \tag{10}$$

$$Z = b + Y \ominus M \tag{11}$$

Assigning different parameters to individual nodes leads to many parameters, which in turn leads to challenges in optimizing these parameters in subsequent training stages, culminating in overfitting concerns. To address this issue, the DGGN method strategically decomposes the parameters, denoted as $\ominus 1, 2, \ldots, N \in R^{N\times C\times F}$, assigned to each node. This decomposition effectively splits the initial parameters into two comparatively smaller parameter matrices, as given by:

$$\ominus_k = W_n \times N_n \quad \text{for } k = 1, 2, \ldots, N \tag{12}$$

The above eq.12 satisfies both a reservoir of weights represented by $W_n \in R^{d\times C\times F}$ and a matrix of node embeddings denoted as $N_n \in R^{N\times d}$, where d denotes the embedding dimension, which is notably less than N. By substituting the previously articulated parameter set $\ominus_k, \ldots, N \in R^{N\times C\times F}$ with the matrix multiplication product of these two parameter matrices, a noticeable reduction in the number of parameters is achieved. This methodology also extends to the treatment of bias terms. Finally, the advanced GCN integrated within the DGGN framework can be formulated as follows:

$$Z = b_g N_g + Y W_n N_n (I + tanh(ReLU(N \times N^T))) \tag{13}$$

Given an individual node, denoted as i, the procedure involves extracting parameters $\ominus_i$, unique to the i node, with a widely shared reservoir weight W_n, while exploiting the inherent node embeddings N_n^i. Conceptually, we can view this process as a mechanism that acquires node-specific patterns by identifying specific patterns inherent in a broader collection of prospective patterns derived from all the traffic sequences.

C. Modeling Dynamic Spatio-Temporal Multi-Scale Feature Unit

Adjacent road segments often exhibit traffic flow characteristics where given node's predicted state of traffic depends on its neighbors traffic dynamics. This outcome is predicted by a confluence of factors, including observed time intervals, immediate historical context, and abrupt variations such as traffic incidents and meteorological conditions. However, the degree of influence exerted by individual nodes is highly variable. We have designed a Spatial and Temporal Multi-Scale Unit (STMU) to address the challenge of graph convolution inability to allocate varying weights to individual nodes. We designed this module to encapsulate

both spatio-temporal correlations within traffic nodes, thereby improving the accuracy of node-level traffic flow predictions. The STMU module consists of three main parts: temporal attention, spatial attention, and multi-scale module. The temporal and spatial modules obtain the three-dimensional tensor and input, encapsulating F-dimensional features for N nodes over successive time instants $(Y : t, Y : t-1, \ldots, Y : t-T+1)$. The previously introduced DGGN performs this extraction. The mathematical formulation for long-term temporal correlations is as follows:

$$Y_{l:t}, Y_{l:t-1}, \ldots, Y_{l:t-T_l+1} \tag{14}$$

where l represents the long-term temporal scale, while Y_l denotes features at the long-term scale.

D. Modeling Attention Based Spatio-Temporal Module

Fig. 3 depicts the Temporal Transformer, denoted as TT, a novel construct introduced in this work to encapsulate persistent long-term temporal dependencies effectively. In contrast to alternative neural network frameworks, the TT demonstrates an enhanced ability to accommodate long-term temporal dependencies seamlessly. Similarly, the ST module of the Spatial Transformer follows a design same as the TT, highlighting its usefulness in handling spatial transformations. To initiate the procedure, the TT starts with a 1×1 convolutional layer to process the input features.

$$Y' = Conv_{1\times1}(Z) \tag{15}$$

$$Y' = W \cdot Z \tag{16}$$

where $Z \in R^{T\times N\times F}$, the operation of $Conv_{1\times1}$ produces vectors of d_g dimensions for each node over temporal intervals, while W is the weight matrix for the 1×1 convolution. This convolutional process is further characterized by using concurrent processing across nodes to capture temporal dependencies effectively. At the same time, we introduce the two-dimensional spatial feature tensor, denoted as $\bar{Y} \in R^{T\times d_g}$, which represents the features associated with each node within the graph G. The time series $\bar{Y} \in R^{T\times d_g}$, as input for the temporal attention unit, we employ a sliding window with a length of T and a channel size of d_g. By applying dynamic computations within high-dimensional latent subspaces, like query subspace $Q \in R^{T\times d_g}$, the key subspace $K \in R^{T\times d_A}$, and the value subspace $V \in R^{T\times d_g}$, temporal dependencies are extracted and delineated.

$$Q = \bar{Y} \cdot W_q \tag{17}$$

$$K = \bar{Y} \cdot W_k \tag{18}$$

$$V = \bar{Y} \cdot W_v \tag{19}$$

where $W_q \in R^{d_g\times d_A}$, $W_k \in R^{d_g\times d_A}$, and $W_v \in R^{d_g\times d_g}$ denotes learned linear mappings. Using data from the past T time steps, we predict future values for the next T time steps. For example, given data at times $(Y : t, Y : t-1, \ldots, Y : t-T-1)$ predicts $(Y : t, Y : t+1, Y : t+2, \ldots, Y : t+\tau)$. we use a scaled dot product function method to predict multiple future steps. This helps us understand how temporal patterns work in both directions in historical traffic data. To make our predictions even more accurate, we combine another set of information, represented as V, with time-related details, represented as W. This helps us understand how different factors are related over time. We also use a fully connected neural network.

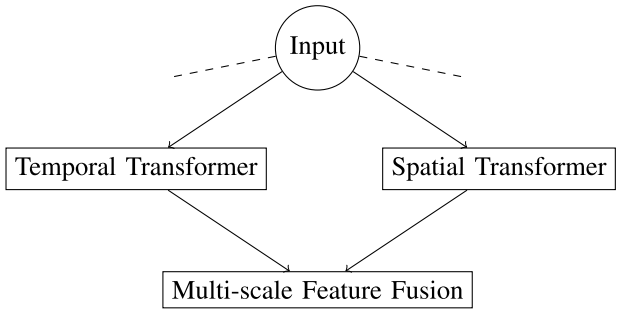


Fig. 3. STMU includes spatial and temporal transformers with multi-scale features.

$$W = tanh(\frac{Q \cdot K^T}{\sqrt{d_A}}) \tag{20}$$

$$S = M \times V \tag{21}$$

$$U = f(S) \tag{22}$$

where K^T and Q represent the matrix product and are normalized to achieve the attention distribution in each time step, while $\sqrt{d_A}$ denotes the weight scaled factor, and the function of tanh is employed to normalize the weight score.

To ensure robustness, we use a residual connection $W' = W + Z$ during the training phase. The resulting output of every node is denoted as $Y' = U + W'$, and this process ultimately leads to the parallelization of outputs across all nodes, yielding $Y \in R^{T\times d_g}$.

We intelligently capture bi-directional temporal dependencies over an extended period to expand the scope of time series forecasting for extended prediction horizons. We accomplish this within the structure of a sliding window, operating at each discrete temporal instance. The hierarchical architecture of the temporal transformer effectively captures complex dependencies across multiple layers, thereby enabling the prediction of long sequences without sacrificing computational efficiency by increasing the parameter T. RNN-based methods, on the other hand, have to deal with gradients that go away. In contrast, models using convolutional methods must clearly define an expanding convolutional layer concerning the parameter T.

E. Multi-Scale Feature Extraction Unit

In the domain of traffic flow prediction, the dynamics of nodes are subject not only to the influence of close neighbors but also to the states prevailing in distant geographical regions. The spatio-temporal multi-scale feature extraction module creates a multi-scale feature extraction process that prioritizes spatial characteristics to enhance the extraction of nodes' intrinsic spatial characteristics. Multi-scale convolutional kernels aid in this effort by extracting different spatial resolutions and depths. This approach can capture a wider range of informative positional cues embedded in

the input tensor. The resulting multi-scale feature representation lends to parallel processing strategies and promotes enhanced contextual insight. However, the number of parameters increases proportionally with the dimensional expansion of the multi-scale convolutional kernel. The group convolution technique is included in the plan to prevent increasing computing complexity.

$$H(t), Y(t) = \text{GRU}\left((\text{GCN}(X, A) \odot A1(t)) \odot A2(t), H(t-1)\right) \tag{23}$$

where $Y(t)$ represents the predicted traffic flow at time step t, while $H(t)$ represents the hidden state at time step t.

F. Training Process of ASTMGCNet

Alg. 1 describes the training procedure of ASTMGCNet. We use back-propagation to initialize and randomly optimize the trainable parameters based on ASTMGCNet. We minimize the loss function of our proposed model using the stochastic gradient method through back-propagation. To increase our method overall effectiveness, we implement the dropout strategy approach to increase the overall effectiveness of our method. The model preprocessed the input data (normalization and management of missing values), then initialized the parameters for the GCN, GRU, and attention mechanisms. Each training epoch involves batch processing of the input data. Attention mechanisms highlight important spatio-temporal aspects, and the representation is improved by multi-scale feature extraction.

Algorithm 1 The ASTMGCNet Algorithm

Input: Historical observations: $\{Y_t\}_{t=1,2,\ldots,T}$
Spatial graph: $Z = \{N_t\}_{t=1,2,\ldots,n}$
Pre-defined graph: $G = (S, E, A)$
Output: Learned ASTMGCNet Model

Initialize dataset $D \leftarrow \emptyset$;
for *each time interval t where* $2 \le t \le n$ **do**
  $Z_t \leftarrow b + Y_t \left(1 + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)$;
  $D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \leftarrow \tanh(\text{ReLU}(NN^{\top}))$;
  $Z \leftarrow b + Y_t \odot \left(I + \tanh(\text{ReLU}(NN^{\top}))\right)$;
  Append Y_t to D ;
end for
Initialize learnable parameters θ in ASTMGCNet ;
repeat
  Randomly select a batch $D_{\text{batch}} \subset D$;
  Update θ by minimizing the objective function on D_{batch} ;
until *until model convergence criteria are met*;
return *Trained ASTMGCNet model*

IV. Experiments and Performance Analysis

We have implemented and empirically evaluated the ASTMGCNet model using two large traffic datasets, BikeNYC and TaxiBJ. We first discussed the datasets we used, how we set some parameters, and how we measured the model performance.

A. Experiment Setup and Datasets

We utilize a Linux server equipped with various software, detailed later in Sec. IV-C, along with the following hardware specifications:

- 8 Intel(R) Xeon(R) E5-2680 v4 CPUs @ 3.80GHz.
- 4 NVIDIA P100 GPUs.
- 256GB RAM.
- CUDA version 8.0, cuDNN version 8.0.

In this experimental study, we focus on predicting traffic flows within two large datasets: BikeNYC and TaxkBJ. Table II explicitly provides detailed information about these datasets.

- **TaxiBJ**: We collected the taxi trajectory data for this dataset from the city of Beijing over 16 months: from July 1, 2013, to October 30, 2013, from March 1, 2014, to June 30, 2014, from March 1, 2015, to June 30, 2015, and from November 1, 2015, to April 10, 2016. The dataset contains over 35,000+ taxi trajectories, effectively capturing urban traffic flows in Beijing. We partition the data, using a subset for training and reserving the most recent four-week period for data testing.
- **BikeNYC**: We generated the BikeNYC dataset between February 5, 2014, and August 29, 2014. This dataset contains 6,900 traffic flow maps, with dimensions of 16×8 and a temporal resolution of one hour. It contains a wealth of information about riding bicycles, including trip mileage, start and finish times, and station identifiers for the origin and destination. For evaluation purposes, we refer to the last ten days of the dataset as the test set and the previous days as the training set.

B. Compared Methods

We conducted a comprehensive comparative assessment to measure the overall performance of our approach, comparing the ASTMGCNet to prominent, well-known models and state-of-the-art methods in the field.

Conventional Time-series Based Models:

- **HA [14]:** Predictions for both passenger flow and traffic congestion can be achieved by using average historical flows at the appropriate time intervals.
- **ARIMA [15]:** This method involves predicting and understanding various issues related to time series and then tailoring models to address them.

Deep Learning Based Models:

- **FC-LSTM [16]:** With a fixed number of recurrent layers in each segment of the encoder and decoder, the model effectively integrates the functionalities of both the codec and LSTM architectures. It is important to remember that each of these recurrent layers has a set quantity of LSTM units.
- **GRU-ED [17]:** A model using an encoder-decoder framework based on GRU was used to perform the machine translation task.
- **STGCN:** This model contains a distinct element called a spatio-temporal block. These blocks are

TABLE II
TRAFFIC DATASET INFORMATION

Datasets	TaxiBJ	BikeNYC
Urban	Beijing	New York
Format of Data	Taxi GPS	Rent Bike
Time span	July 1, 2013 – October 30, 2013 March 1, 2014 – June 30, 2014 March 1, 2015 – June 30, 2015 November 1, 2015 – April 10, 2016	February 5, 2014 – August 29, 2014
Period	30 mins	1 hour
Taxis and Bikes	35,000+ taxis	6,900+ bikes
Map size	32 × 32	16 × 8

stacked in multiple layers, denoted as k layers, within the model core.

- **ConvLSTM [18]:** Additional convolutional layers are introduced to enhance feature extraction. The model is trained using the Adam optimizer with a learning rate of 0.001
- **STDN [19]:** Uses attention mechanisms to emphasize important features. Utilizes an Adam optimizer with a learning rate of 0.001.
- **ST-ResNet [20]:** Multiple residual blocks are employed. When compared to other approaches, this one has proven to perform better.
- **MST3D [21]:** Employs 3D CNNs to capture spatio-temporal dependencies. Utilizes 3D convolutional layers.
- **STVANet [22]:** This model proposes a spatio-temporal visual attention neural network (STVANet), an innovative 2D CNN that incorporates a distinctive visual attention module. Incorporates LKA and SE mechanisms.
- **ASTMGCNet:** In this work, we combine GCN and GRU, this framework effectively extracting dynamic features of spatio-temporal from node properties.

C. Implementation Details and Evaluation Metrics

1) Data Preprocessing: In the case of the TaxiBJ dataset, the urban area was partitioned into discrete grid regions measuring 32 × 32 units, with each temporal segment corresponding to a duration of 30 minutes. In a parallel manner, for the BikeNYC dataset, we adopted a grid structure comprising dimensions of 16 × 8 to represent the city map, with each temporal interval set at one hour. We scaled the traffic flows between [-1, 1] using a method known as min-max normalization to make them comparable. By guaranteeing that every feature is on the same scale and preventing any one feature from controlling the learning process, scaling the data between [-1, 1] enhances model training. This prevents problems like gradient explosion and disappearing and speeds up the model convergence. After the prediction, we adjusted the values to match our research usual output and compared them with the actual values. The Table III briefly describes the hyperparameter settings.

TABLE III
HYPERPARAMETERS FOR ASTMGCNET IN TRAFFIC FLOW PREDICTION

Hyperparameter Description	Value or Range
Keras	2.2.4
Tensorflow	1.13.1
Python	3.6
Learning Rate	0.001
Optimization Algorithm	Adam
Number of Epochs	200
Dropout Rate	0.25
Batch Size	64
Early Stopping Patience	15
Temporal Convolution Layers	2
Kernel Size (Temporal Convolution)	5
Activation Function (Graph Convolution)	ReLU

2) Evaluation Metrics: We evaluate our model performance using two pivotal metrics: mean average percentage error (MAPE) and root mean square error (RMSE).

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2} \tag{24}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}_t - y_t|}{y_t} \tag{25}$$

where, y_t and $\hat{y}_t$ denote the actual and predicted flow maps, respectively. Moreover, the sample size n is employed to evaluate the accuracy and reliability of the prediction results.

D. Experiment Results and Analysis

Using the setup and settings mentioned above, we compared how well our ASTMGCNet model performed against other competing methods on two datasets: BikeNYC and TaxiBJ. The results are shown in Fig. 4 and Fig. 5. The ASTMGCNet stands out by achieving the lowest MAPE and RMSE values among all methods, demonstrating its superior performance. Specifically, for BikeNYC, the RMSE is about 4.06, and the MAPE is 18.86%. Similarly, for TaxiBJ, the MAPE is about 12.56%, and the RMSE is about 13.98, respectively. This indicates that our added attention and multi-scale feature framework effectively improve the proposed model performance. In addition to achieving superior accuracy metrics, ASTMGCNet consistently demonstrated stable performance across varying traffic conditions in both datasets. How the

model handles the TaxiBJ dataset, which has more intricate and dense traffic patterns, demonstrates how well it can adjust to various traffic dynamics. The durability of ASTMGCNet spatio-temporal attention mechanisms is demonstrated by the fact that it maintained its predicted precision despite the higher variability in this dataset. Additionally, the model multi-scale feature extraction demonstrated efficacy in identifying regional and worldwide traffic patterns, guaranteeing its continued high reactivity in the face of abrupt variations in flow. These results highlight the potential of ASTMGCNet for real-world applications where diverse and dynamic traffic situations need to be efficiently handled.

We compared our method average results to those of other methods on two datasets, TaxiBJ and BikeNYC. The results are shown in Table IV. We calculated the average performance for each model by running it ten times. The Table. IV clearly shows that our proposed ASTMGCNet model outperforms the other models regarding effectiveness and prediction accuracy. These findings emphasize the effectiveness of our model in capturing spatio-temporal dependencies for accurate traffic flow prediction. This robust result demonstrates even more the potential of ASTMGCNet for useful applications in real-world traffic prediction problems.

Our ASTMGCNet method significantly reduces prediction errors during model training when combined with the spatio-temporal GCN network. This demonstrates our method adaptability for urban traffic flow prediction. The ASTMGCNet model efficiency has consistently improved, particularly in long-term traffic flow prediction. In particular, traditional methods such as ARIMA and HA struggle to produce highly accurate forecasts, highlighting the limitations of approaches that ignore dynamic spatio-temporal dependencies and focus solely on historical statistical relationships. ARIMA focuses on modeling univariate time series and does not incorporate spatio-temporal correlations between different regions. This limitation prevents the model from exploiting important network structure and road system insights. Incorporating spatial correlations improves regression models such as FC-LSTM and GRU-ED, but they can still not capture the complex dynamic non-linear spatial and temporal dependencies. The FC-LSTM cannot typically explicitly capture the spatial dependencies in traffic data, which can be crucial for accurate traffic flow prediction in scenarios where road networks play a significant role. Furthermore, our model outperforms the ST-ResNet and MST3D models. The ST-ResNet residual structure is limited in capturing the highly non-linear relationships in complex traffic dynamics. Similarly, the MS3TD model has more hyperparameters that require tuning to achieve optimal performance. Insufficient tuning can lead to suboptimal results.

Current methods struggle to accurately predict traffic flow in a given area. Existing methods, including CNN, RNN, and LSTM, rely on manually extracting information from images, which reduces prediction accuracy. We need updated methods that automatically gather information using techniques such as GCN and attention modules. These approaches dynamically collect information from models, improving prediction accuracy while saving time and resources. If we look at Fig.4 and Fig. 5 and compare them with existing methods, it is clear that the existing models do not perform as well as our proposed method. Our proposed model achieves superior accuracy in terms of time efficiency, cost-effectiveness, and reliability.

TABLE IV
COMPARISONS OF MODELS ON TAXIBJ AND BIKENYC

Methods	BikeNYC		TaxiBJ	
	RMSE	MAPE	RMSE	MAPE
HA	15.80	38.99%	55.99	36.89%
ARIMA	10.10	27.99%	20.99	23.45%
GRU-ED	8.23	25.09%	20.18	19.35%
STGCN	7.32	25.39%	19.18	18.21%
FC-LSTM	6.49	22.98%	17.32	17.42%
ST-ResNet	6.34	21.31%	16.89	15.39%
STDN	6.20	20.65%	16.50	15.15%
MST3D	5.81	20.39%	15.90	14.41%
STVANet	5.34	19.86%	14.67	13.91%
Our(ASTMGCNet)	**4.06**	**18.86%**	**13.98**	**12.56%**

Our model combines the spatio-temporal GCN network to make training more accurate and reduce prediction errors. We tested our model against existing long-term traffic flow prediction methods and found it performed much better. The existing models used CNN, LSTM, or RNN separately with manual data extraction, which made training longer instead of more efficient. We need to dynamically extract features from the traffic flow data to improve prediction accuracy. Our model, which uses neural networks such as GCN and RNN with an attention mechanism, allows us to easily estimate the traffic flow in a city, including the number of people entering and leaving. This allows us to reduce training time, make predictions more accurate, and save time and cost.

E. Results of Different ASTMGCNet Variants

To show that the proposed ASTMGCNet method works, we did a full empirical evaluation that looked at all of its variations, accounted for different parameter settings, and kept the experimental setups consistent. Specifically, we investigated three critical aspects: (i) the influence of model depth and (ii) the consequences of changes in kernel and filter numbers. The following sections provide a detailed discussion of these research results. Finally, we analyze the computational complexity of our proposed algorithm, including considerations related to prediction and training times.

1) Impact of Depth Model: We conducted an extensive empirical experiment with nine variations of the ASTMGCNet model, each characterized by different depths. Our goal is to understand better the effectiveness of different depths within the ASTMGCNet model. The experimental results are shown in Fig.6. Focusing specifically on the TaxiBJ dataset, Fig.6 provides a detailed visualization of the impact of model depth. Our results show a compelling relationship between the number of residual units representing network depth and the RMSE metric. As we progressively increase the depth of the network by increasing the number of residual units, we observe an initial increase in RMSE, followed by a subsequent increase in RMSE. This trend indicates that deeper networks yield better results. Nevertheless, it is crucial to note that as the number

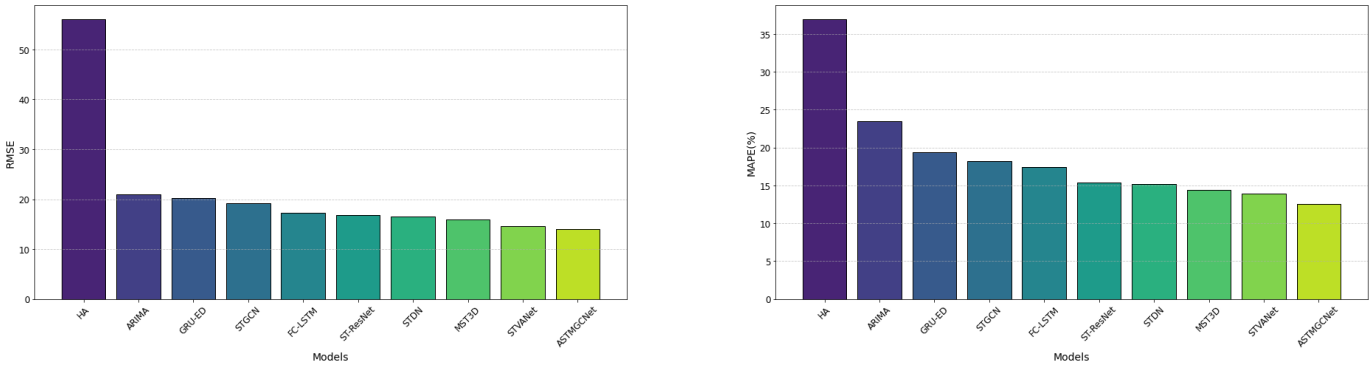


Fig. 4. TaxiBJ prediction results with ASTMGCNet, evaluated by (a) RMSE and (b) MAPE (%).

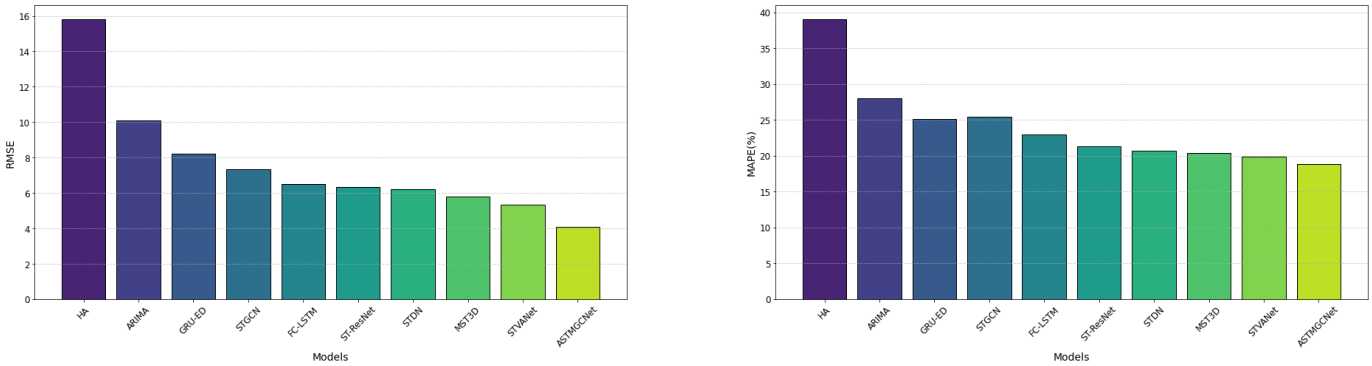


Fig. 5. BikeNYC prediction results with ASTMGCNet, evaluated by (a) RMSE and (b) MAPE (%).

of residual units exceeds a certain threshold, such as ≥ 12, the training process becomes more complex, and the risk of overfitting increases. This observation highlights the need for further research to understand the results obtained fully.

2) Impact of Filter Numbers and Kernel Sizes: We have extensively explored different dataset training configurations to optimize kernel size selection. Achieving precise prediction accuracy depends on carefully selecting the kernel size. As shown in Fig.7 (b), we investigated kernel sizes profound impact on the ASTMGCNet methodology performance. To carefully observe and illustrate the effects of kernel size, we systematically adjusted the kernel dimensions, ranging from 2×2 to 5×5. Fig.7 (b) shows a noteworthy trend: an increase in kernel size corresponds to a decrease in the RMSE metric. This compelling observation underscores the potential of optimal kernel size to improve the faithful representation of spatial correlations. In addition, as shown in Fig.7 (a), using larger filter sizes consistently yields significantly better results than smaller ones. This pattern is consistent with our observations regarding kernel size and number of filters.

Fig. 8 illustrates that the traffic state matrix varies across different time segments. Initially, the feature matrix before training appears random and unstructured. However, after training, the feature matrix displays distinct temporal patterns similar to the traffic state matrix. This demonstrates that the model successfully learns and abstracts temporal features for prediction.

F. Model Efficiency

Subsequently, considering the response time, we thoroughly investigate our proposed model performance in predicting outcomes and the training phase. We have summarized the results in the Table. V. One thing that immediately stands out is that our ASTMGCNet model performs better than the STDN and MST3D models in both the training and prediction phases. It is worth noting that the STDN method takes the longest time for both training and prediction. This is because STDN uses certain computations called local CNNs to make predictions, and it also has to go through the entire region using a sliding window. For example, if we work with a dataset like BikeNYC, predicting values for the entire region requires a certain computation 16×8 times. In the BikeNYC dataset, our ASTMGCNet model significantly outperforms the ST-ResNet approach. Our ASTMGCNet method primarily achieves this by combining two techniques, GCN and GRU, grounded in graph convolutional networks. We see a similar trend on the TaxiBJ dataset. Our ASTMGCNet model outperforms both the MST3D and STDN baselines in terms of both prediction speed and training speed. These results demonstrate the advantages of our model.

The space and time complexities of Alg. 1 are closely related to key parameters, mainly the amount of available historical information and the number of features. In particular, the complexities depend on two crucial factors: the training time, denoted by m, which is the time required to form the data set D, and the learning time, denoted by n, which characterizes the time required to acquire knowledge within the model while keeping the computational overhead for θ constant. It is important to understand that the algorithm performance shows different behaviors under different circumstances. In the best-case scenario, the algorithm exhibits a time complexity denoted by $\omega(m+n)$. The computational cost escalates linearly

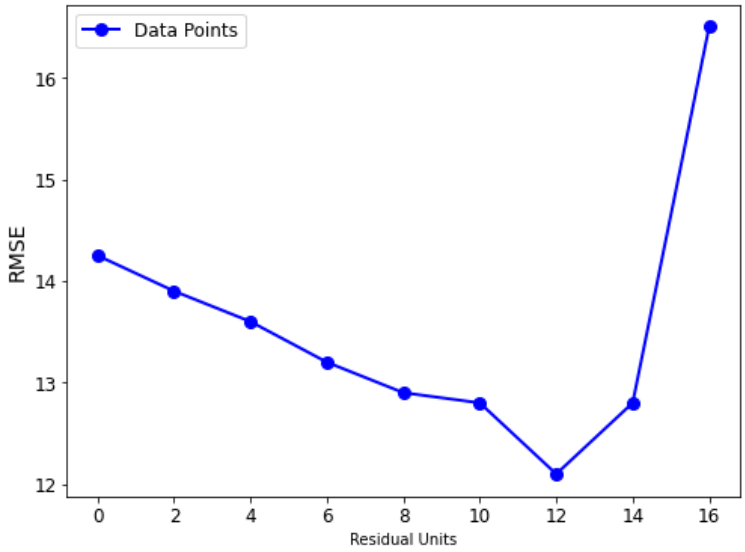


Fig. 6. The influence of model depth on RMSE metric through variational analysis.

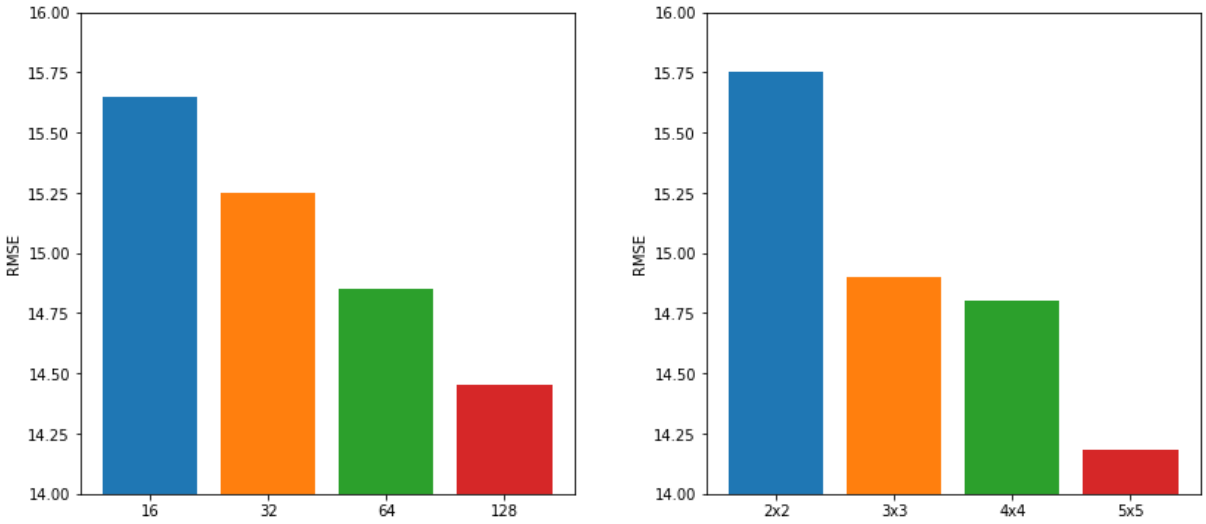


Fig. 7. Prediction Results of various (a) filter numbers and (b) kernel sizes.

TABLE V
TRAINING AND PREDICTION TIMES FOR BIKENYC AND TAXIBJ

Methods	BikeNYC		TaxiBJ	
	Training Time (s/epoch)	Prediction Time (s)	Training Time (s/epoch)	Prediction Time (s)
STDN	18,973	88.7%	369,601	207.4%
MST3D	126	0.23%	5,902	2.74%
Our (ASTMGCNet)	117	0.12%	4,849	2.69

with training and learning times in this case. This scenario represents an optimal match of algorithm efficiency with available computational resources, culminating in a favorable performance profile. Conversely, the worst-case scenario reveals a potentially significant time complexity that can rise to $O((mn)(m+n))$ and further simplifies to $O(m^2n+n^2m)$. This particular scenario depends on the number of observations and the dimensionality of the features.

V. RELATED WORKS

In this section, we emphasize how we use deep learning to understand the spatio-temporal correlation for traffic prediction.

A. Traffic Flows Prediction

In the field of traffic prediction, the incorporation of predictive spatial information is emerging as a key factor. Modern methods using convolutional neural networks (CNNs) [23], [24], [25] for traffic flow mapping often rely on grid-based map segmentation. However, the inherent regularity of grid-based data limits their ability to represent complex spatial information effectively. In light of this, they introduce the paradigm of Graph Neural Networks (GNNs) [26], an adept choice for accommodating non-Euclidean data and enabling the construction of spatial topological structures. In [27], (DCRNN) innovatively presented a diffusion convolutional RNN based on distance graphs. Their approach incorporates bi-directional random walks on the graph to capture spatial dependencies. In [28] and [29], proposed a spatio-temporal GCN network, where the urban road network is considered an adjacency matrix instead of relying on a grid representation.

However, the establishment of spatial topology in these methods through predefined adjacency matrices remains static

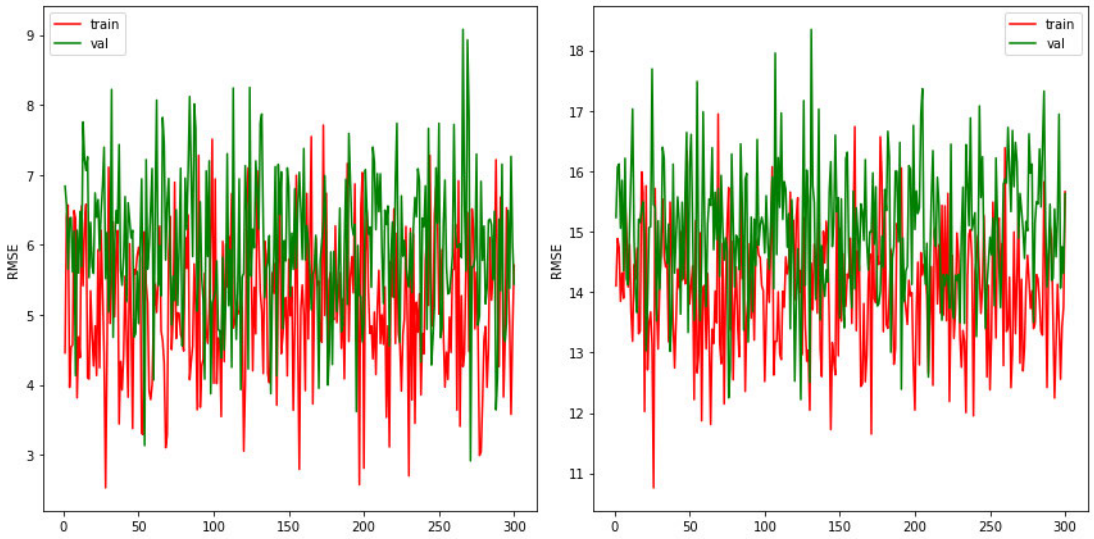


Fig. 8. Visualization of the traffic state matrix and corresponding feature matrices before and after training.

TABLE VI
COMPARISON OF MODELS BASED ON SPATIO-TEMPORAL DATA HANDLING AND UNIQUE CAPABILITIES OF ASTMGCNET

Model	**Spatial**	**Temporal**	**Both Spatio-temporal**	**Unique Capabilities of Models**	**Key Contributions**
HA	No	Yes	No	Limited to historical data, less effective in capturing rapid changes in traffic patterns	Relies on historical averages, without spatial dependency considerations.
ARIMA	No	Yes	No	Ineffective for capturing complex spatio-temporal relationships	Time-series model effective for univariate data lacks spatial considerations.
Att-DHSTNet	Yes	Yes	Yes	Lacks multi-attention mechanism, requires extensive hyperparameter tuning	Employs spatio-temporal attention to improve feature extraction and prediction accuracy.
AAtt-DHSTNet	Yes	Yes	Yes	Limited handling of multi-attention mechanisms, suboptimal for edge computing resource allocation	Utilizes attention in spatio-temporal graph convolution to capture fine-grained features.
ASTMGCNet	Yes	Yes	Yes	Optimized multi-attention mechanism for dynamic heterogeneous spatio-temporal feature extraction, efficient for edge computing	Advanced multi-attention spatio-temporal graph network, excels in prediction accuracy and dynamic feature adaptation.

and limited in nature, with inherent limitations in encapsulating the complex characteristics of complex road networks. Existing methods that rely on predefined graph structures face significant challenges. First, predefined graph constructs struggle with sparsity issues. Sparsity, which is particularly evident in large graphs, manifests itself in the adjacency matrix, which leads to computationally inefficient operations and may fail to encapsulate the totality of spatial dependencies comprehensively. This, in turn, can potentially undermine the model's accuracy [30]. A secondary concern revolves around the inflexible nature of predefined adjacency matrices, which makes them ill-suited to accommodate dynamic graphs or graphs characterized by recurrent structural changes [31].

Deep neural network models have recently been in the spotlight due to rapid improvements in deep learning [32]. These models can capture the dynamic properties of traffic data and produce state-of-the-art findings. The models can be categorized into two groups according to whether or not they take spatial dependence into account. Certain methods consider temporal dependence; for example, feed forward neural networks were employed by [33] to anticipate traffic flow. Similarly, [34] presented a model that combines a regression model with a deep belief network (DBN), showing increased prediction accuracy by collecting random features from various traffic datasets. The importance of traffic congestion was highlighted by [35], who provided a common definition and employed a locality constraint distance metric learning technique for congestion detection. To better capture the qualities of traffic congestion [36] later developed a robust hierarchical deep learning algorithm for semantic feature extraction. Furthermore, models with strong temporal dependency learning capabilities, such as RNN,

LSTM, and GRU have demonstrated improved prediction performance [37].

In conjunction with the previous discourse on the construction of spatial topology, the strategic modeling of spatial and temporal dependencies emerges as a key factor in traffic prediction. Capturing temporal dependencies requires recourse to RNN and their diversification, which are expertly tailored to sequential data with intrinsic capabilities to capture temporal correlations of a temporal nature [38], [39]. The recurrent nature of RNN operations provides increased model flexibility. The advent of the GRU [40] provides the RNN paradigm with enhanced modeling capabilities, particularly well suited for long sequence correlations, and effectively circumvents the uncertainty of gradient vanishing GRU, a subset of the RNN, improves on the limitations inherent in its ancestor by boasting the ability to preserve long-term memory and avoiding the gradient vanishing issue inherent in back-propagation.

In contrast to the more complicated LSTM [41], GRU advocates a simpler and more trainable architecture. Gaining prominence for their effectiveness and versatility, attention mechanisms [42], [43] have found widespread application in various domains. These mechanisms automatically focus on central information extracted from historical input data. Graph Attention Networks (GAN) [4], [44] have made significant progress in traffic prediction by constructing spatial models. STSGCN [45] introduces a novel concept by devising a synchronized GCN model for spatial and temporal aspects. These modules adeptly capture correlations across space and time. In a complementary approach, EPSANet [46] leverages multi-scale information pooling to improve the capacity of space and time features. This technique has effectively navigates the complexities inherent in multi-scale feature tensors, facilitating the extraction of spatial insights across various scales. We advised the ASTMGCNet model, which combines GCN with the attention unit, to collectively incorporate spatio-temporal traffic flow features, drawing inspiration from previous research as discussed above.

Firstly, not all cities or areas can access the vast volumes of historical traffic data the model uses to identify spatio-temporal trends. Second, even though the attention mechanism and multi-scale feature extraction of the model enable it to adjust to dynamic changes in traffic patterns, the model might not be able to handle severe situations such as unexpected road closures, accidents, or meteorological circumstances resulting in abrupt traffic flow changes. Furthermore, the model complexity raises its computational cost. It necessitates a large amount of memory and processing capacity during training, which may restrict its useful application in real-time traffic management systems. In conclusion, although the model has been evaluated on two datasets, TaxiBJ and BikeNYC, its applicability to additional datasets and cities with different traffic patterns has not yet been completely determined.

B. Attention Mechanism

The attention mechanism has become an important tool in various tasks such as language translation and image recognition, helping to decide which parts are most important to focus on [47]. This mechanism uses a process where a question and a set of key-value pairs create an output. The output is a mixture of the values, with the importance of each value determined using a compatibility function between the question and its key. Attention mechanisms are generally used in conjunction with convolutional or recurrent networks. For instance, in [48], the attention mechanism is used in the decoder to find a logical match between a source sentence and its target translation. In [49], the model relies on attention to see connections between input and output. Recently, attention mechanisms have been used in graph neural networks. In [50], they introduced gated attention networks for graph learning. And [43] used spatial-temporal attention to understand how traffic data changes.

VI. Conclusion and Future Work

In this paper, we propose a new approach to traffic flow prediction using a deep learning model called Attention-based Spatio-temporal Multi-Scale Graph Convolutional Recurrent Network (ASTMGCNet). We extended the typical graph convolutional networks in T-CPS to handle very complicated traffic scenarios by adding a spatio-temporal multi-scale feature extraction module and a dynamic graph creation component. These components help the model learn specific patterns for each point and understand relationships at different levels of detail in the data. To validate the effectiveness of our ASTMGCNet model, we conducted experiments using baseline methods on two large-scale traffic flow datasets. The experimental outcomes affirm the ASTMGCNet model performance and the proposed STMU module.

In the future, we aim to increase the model simplicity to facilitate its seamless application to related tasks. We aim to strengthen the model performance and expand its applicability through this approach. In addition, we plan to improve our model by integrating external factors, such as unexpected events and weather conditions, to increase its predictive accuracy. We anticipate these improvements will enhance the model practicality and application. We plan to implement optimizations like distributed computing and parallel processing to ensure scalability, allowing the model to handle larger datasets efficiently. Additionally, scalability tests can be incorporated to evaluate network performance with higher node counts. Techniques like model compression and lightweight architecture could be explored for real-time applications to ensure low-latency predictions in real-world scenarios.

References

[1] A. Ali, Y. Zhu, and M. Zakarya, "A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing," *Multimedia Tools Appl.*, vol. 80, no. 20, pp. 31401–31433, Aug. 2021.

[2] C. Creß, Z. Bing, and A. C. Knoll, "Intelligent transportation systems using roadside infrastructure: A literature survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6309–6327, Jul. 2024.

[3] D. S. Sarwatt, Y. Lin, J. Ding, Y. Sun, and H. Ning, "Metaverse for intelligent transportation systems (ITS): A comprehensive review of technologies, applications, implications, challenges and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6290–6308, Jul. 2024.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[5] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[6] L. Bai, L. Yao, S. S. Kanhere, Z. Yang, J. Chu, and X. Wang, "Passenger demand forecasting with multi-task convolutional recurrent neural networks," in *Proc. 23rd Pacific–Asia Conf.*, Macau, Macau. Berlin, Germany: Springer-Verlag, Aug. 2019, pp. 29–42.

[7] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 3656–3663.

[8] L. Bai, L. Yao, X. Wang, C. Li, and X. Zhang, "Deep spatial–temporal sequence modeling for multi-step passenger demand prediction," *Future Gener. Comput. Syst.*, vol. 121, pp. 25–34, Aug. 2021.

[9] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2020.

[10] M. X. Hoang, Y. Zheng, and A. K. Singh, "FCCF: Forecasting citywide crowd flows based on big data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, pp. 1–10.

[11] M. J. Cushing and D. I. Rosenbaum, "Historical averages, units roots and future net discount rates: A comprehensive estimator," *J. Forensic Econ.*, vol. 19, no. 2, pp. 139–159, Mar. 2006.

[12] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transp. Res. Rec.*, vol. 1644, no. 1, pp. 132–141, 1998.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[14] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," in *Modeling Financial Time Series With S-PLUS®*. New York, NY, USA: Springer, 2006, pp. 385–429.

[15] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, 2003.

[16] Y. Tong et al., "The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1653–1662.

[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[18] S. Xingjian, C. Zhourong, W. Hao, Y. Dit-Yan, W. Wai-Kin, and W. Wang-Chun, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 802–810.

[19] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial–temporal similarity: A deep learning framework for traffic prediction," 2018, *arXiv:1803.01254*.

[20] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[21] C. Chen et al., "Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 893–898.

[22] H. Yang, J. Jiang, Z. Zhao, R. Pan, and S. Tao, "STVANet: A spatio-temporal visual attention framework with large kernel attention mechanism for citywide traffic dynamics prediction," *Exp. Syst. Appl.*, vol. 254, Nov. 2024, Art. no. 124466.

[23] J. Zhu et al., "AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," 2020, *arXiv:2011.11004*.

[24] S. Akbar, Q. Zou, A. Raza, and F. K. Alarfaj, "IAFPs-Mv-BiTCN: Predicting antifungal peptides using self-attention transformer embedding and transform evolutionary based multi-view features with bidirectional temporal convolutional networks," *Artif. Intell. Med.*, vol. 151, May 2024, Art. no. 102860.

[25] A. Raza, J. Uddin, A. Almuhaimeed, S. Akbar, Q. Zou, and A. Ahmad, "AIPs-SnTCN: Predicting anti-inflammatory peptides using fastText and transformer encoder-based hybrid word embedding with self-normalized temporal convolutional networks," *J. Chem. Inf. Model.*, vol. 63, no. 21, pp. 6537–6554, Nov. 2023.

[26] Y. Li, Z. Hao, and H. Lei, "A review of convolutional neural networks," *Comput. Appl.*, vol. 36, no. 9, pp. 2508–2515, 2016.

[27] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[28] Y. Han, S. Wang, Y. Ren, C. Wang, P. Gao, and G. Chen, "Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 6, p. 243, May 2019.

[29] S. Sacone, "Platoon control in traffic networks: New challenges and opportunities," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 4, pp. 149–183, Apr. 2024.

[30] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17804–17815.

[31] F. Li et al., "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 1, pp. 1–21, Feb. 2023.

[32] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[33] D. Park and L. R. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 14, no. 5, pp. 357–367, Sep. 1999.

[34] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[35] Q. Wang, J. Wan, and Y. Yuan, "Locality constraint distance metric learning for traffic congestion detection," *Pattern Recognit.*, vol. 75, pp. 272–281, Mar. 2018.

[36] Q. Wang, J. Wan, and X. Li, "Robust hierarchical deep learning for vehicular management," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4148–4156, May 2019.

[37] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks," *Transp. Res. Rec.*, vol. 1811, no. 1, pp. 30–39, Jan. 2002.

[38] S. Rahmani, A. Baghbani, N. Bouguila, and Z. Patterson, "Graph neural networks for intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8846–8885, Aug. 2023.

[39] X. Zhang et al., "Traffic flow forecasting with spatial–temporal graph diffusion network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 17, pp. 15008–15015.

[40] R. Al-Huthaifi, T. Li, Z. Al-Huda, and C. Li, "FedAGAT: Real-time traffic flow prediction based on federated community and adaptive graph attention network," *Inf. Sci.*, vol. 667, May 2024, Art. no. 120482.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "GeoMAN: Multi-level attention networks for geo-sensory time series prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3428–3434.

[43] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[44] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Aug. 2019.

[45] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 914–921.

[46] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, "EPSANet: An efficient pyramid squeeze attention block on convolutional neural network," in *Proc. Asian Conf. Comput. Vis.*, 2022, pp. 1161–1177.

[47] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

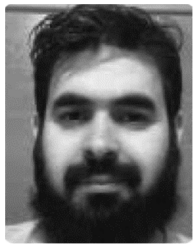
[48] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[49] K. Kim, S. Jin, S. Ko, and J. Choo, "STGRAT: A spatio-temporal graph attention network for traffic forecasting," in *Proc. Int. Conf. Inf. Knowledge Manage.*, 2020, pp. 1–8.

[50] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," 2018, *arXiv:1803.07294*.

Ahmad Ali received the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University, China, where he focused on advanced machine learning and intelligent systems. He is currently a Post-Doctoral Researcher at Shenzhen University, China. He has published extensively in high-impact journals and is actively involved in the academic community. Notably, according to the Web of Science, he has reviewed over 1,200 research articles, reflecting his deep expertise and commitment to advancing his field. His research interests center on spatio-temporal data analysis and graph neural networks, with applications in intelligent transportation systems and the IoT.

Inam Ullah (Member, IEEE) received the B.Sc. degree in electrical engineering (telecommunication) from the Department of Electrical Engineering, University of Science and Technology Bannu (USTB), Khyber Pakhtunkhwa, Pakistan, in 2016, and the master's and Ph.D. degrees in information and communication engineering from the College of Internet of Things (IoT) Engineering, Hohai University (HHU), Changzhou Campus, China, in 2018 and 2022, respectively. He completed his Postdoc with Brain Korea 2021 (BK21) at Chungbuk Information Technology Education and Research Center, Chungbuk National University, Cheongju, South Korea, in March 2023. He is currently an Assistant Professor with the Department of Computer Engineering, Gachon University, South Korea. He has authored more than 130 peer-reviewed articles on various research topics and five books as an editor. His research interests include robotics, the Internet of Things (IoT), wireless sensor networks (WSNs), underwater communication and localization, underwater sensor networks (USNs), artificial intelligence (AI), big data, and deep learning.



Shabir Ahmad (Senior Member, IEEE) received the B.S. degree in computer system engineering from the University of Engineering and Technology, Peshawar, Pakistan, the M.S. degree in computer software engineering from the National University of Science and Technology, Islamabad, Pakistan, in 2013, and the Ph.D. degree from the Department of Computer Engineering, Jeju National University, Republic of Korea. He was with the Software Engineering Department, University of Engineering and Technology, Peshawar, as a Faculty Member, from 2016 to 2017. Since 2020, he has been a Research Professor with the Department of Computer Engineering, Gachon University. His research interests include the Internet of Things applications, cyber-physical systems, intelligent systems, deep learning, and reinforcement learning, to name.

Zongze Wu (Member, IEEE) received the Ph.D. degree in pattern reorganization and intelligence system from Xi'an Jiaotong University, Xi'an, China, in 2005. He is currently a Professor with the College of Mechatronics and Control Engineering, Shenzhen University, and also with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China. He is the author and coauthor of 70 SCI-indexed papers, which include IEEE TAC, Automatica, IEEE TCYB, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, IEEE TIP, IEEE TCSVT, *Pattern Recognition*, and *Signal Processing*.

Jianqiang Li (Senior Member, IEEE) received the B.S. and Ph.D. degrees from South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively. He is currently the Executive Director of the National Engineering Laboratory for Big Data System Computing Technology of China and also the Vice Dean of the College of Computer Science and Software Engineering, Shenzhen University. He served on the editorial board of seven journals and has been selected for the list of the world's top scientists' lifelong influences by Stanford University. He led three projects for the National Natural Science Foundation and five projects for the Natural Science Foundation of Guangdong, China. His major research interests include robotics, hybrid systems, the Internet of Things, and embedded systems.

Xiaoshan Bai (Member, IEEE) received the Ph.D. degree in systems engineering from the University of Groningen, Groningen, The Netherlands, in 2018. From January to July 2015, he was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. From 2018 to 2019, he was a Lecturer with the Faculty of Science and Engineering, University of Groningen. From 2019 to 2020, he was a Post-Doctoral Fellow with the Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands. He is currently a Research Professor with the College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen, China. His main research interests include multivehicle/robot task assignment, path planning, logistic scheduling, and heuristic algorithms.