Received 12 February 2024, accepted 23 April 2024, date of publication 20 May 2024, date of current version 5 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3403516

SURVEY

Spatio-Temporal Feature Engineering for Deep Learning Models in Traffic Flow Forecasting

HONGFAN MU[1], NOURA ALJERI[1,2], (Senior Member, IEEE), AND AZZEDINE BOUKERCHE[1], (Fellow, IEEE)

[1]PARADISE Research Laboratory, EECS, University of Ottawa, Ottawa, ON K1N 6N5, Canada
[2]Computer Science Department, Kuwait University, Kuwait City 12037, Kuwait

Corresponding author: Noura Aljeri (aljeri@cs.ku.edu.kw)

This work was supported in part by NSERC, Canada Research Chairs Program, and NSERC-CREATE TRANSIT.

ABSTRACT In the past decade, modern transportation systems have employed various cutting-edge deep-learning approaches for traffic flow prediction. Due to its significant temporal correlations, researchers have mainly focused on extracting temporal features from traffic flow data. As a result, time-series models based on deep learning methods like Gated Recurrent Unit (GRU), Long-Term Short-Term Memory (LSTM), and Temporal Convolutional Networks (TCN) have been introduced as solutions for traffic flow prediction. However, the spatial features of the road network have also shown an impact on the prediction, leading to the application of deep learning methods on spatial dependency modeling for this problem. This paper defines the traffic flow forecasting problem, considering both time-series information with and without spatial information and the corresponding techniques of current solutions to depict spatio-temporal traffic dependency. We propose a new taxonomy of spatial and temporal dependencies in the fine-grained subcategory and the methods depicting them based on neural network-based models. Furthermore, we highlight the architecture of spatial and temporal ensembles in Spatio-temporal modelling based on the fine-grained categories obtained. We point out several open issues and future directions of traffic flow forecasting, such as graph reconstruction, temporal and spatial information data balance, and multi-model spatial and temporal correlations.

INDEX TERMS Deep learning, graph neural network, spatial-temporal dependence, traffic flow forecasting.

I. INTRODUCTION

Traffic management is a serious issue worldwide due to the increase of vehicles on the road every year [1]. An increasing number of road users can potentially bring continuous traffic congestion every minute, which is able to decrease the utilization of traffic efficiency. Unpredictable traffic conditions can reduce traffic management efficiency and deteriorate passengers' travel experience [1]. Hence, an efficient and accurate method for traffic management is required to solve these problems. In modern society, intelligent transportation systems (ITS) aim to establish reliable connections between roads, vehicles, and people, leading drivers to travel safely and freely on the road, which could contribute to improving traffic efficiency [2].

Traffic flow forecasting is imperative to intelligent transportation systems (ITS). That is because the accurate prediction of traffic conditions in the next time steps could lead to a successful ITS [3], [4]. Since state-of-art traffic flow forecasting methods can contribute to reducing the possibility of traffic accidents on the road, attaining the reasonable and significant utilization of traffic road networks leads to alleviating urban traffic congestion efficiently. Besides, the typical traffic applications in the traffic field, such as real-time traffic signal control [5], [6], traffic demand [7], route guidance [8], [9], [10], automatic navigation [11], [12], are all based on the guidance of an accurate traffic flow prediction, which shows the significance of the reliable and proper traffic flow prediction.

The associate editor coordinating the review of this manuscript and approving it for publication was Rajeeb Dey.

The traffic flow prediction problem estimates future traffic conditions based on historical traffic flow observations and current real-time information. At the beginning of this area, solutions mainly focused on extracting the temporal dependency from the observed data [4], [13], [14] since the flow fluctuation is directly affected by the recent time intervals. With further research, the topological structure of the road network has been found to contribute to the prediction. Therefore, researchers tried to devise a traffic forecasting solution based on the combination of spatial and temporal dependencies mined behind the traffic flow data.

Oswald et al. [15] pointed out that non-parametric methods, referring to the models without parametric presetted, can contribute better results than parametric models (machine learning) because of their performance in capturing the non-linearity feature of traffic flow data. Moreover, deep learning models can deal with massive quantities of data generated by advanced transport detection systems every minute, which outperform machine learning approaches that limit computation ability for solving ''big data'' problems. Due to the outstanding performances of feature learning and end-to-end modeling on mining information from big data, NN-based models have shown significant performance on temporal and spatial feature extraction, which implies its importance in solving traffic flow prediction problems.

Besides, the ensembles of spatial and temporal feature extraction should be emphasized with enough importance to traffic solutions. When we looked through all the traffic-related competitions in Kaggle, it was interesting to find that most of the winning solutions are considered ensemble learning. The answer might be the combination of each base learner has contributed its own different knowledge to the whole picture. Therefore, each feature the base learners learned will complement the others while the errors cancel out each other. Because of this, the inspiration of the Spatio-temporal ensembles in this survey can be, what is the practical and efficient architecture of spatial and temporal feature expression in traffic flow forecasting?

Moreover, we found that a solid solution to traffic flow forecasting relies on the thorough consideration of both the understanding and detailed analysis of the real-life problem. For example, Guo et al. considered the proposed Spatial-temporal (ST) block to fuse three fine-grained temporal dependencies (hourly, daily, and weekly) for prediction to capture the detailed representations of the features [22]; Geng et al. modeled spatial dependency in three subcategories: neighboring information, region functional similarity, and geographically distant but reachable regions for richer spatial information [23]; Yang et al. developed the model with benefits of multi-scale spatial and temporal dependency to extract informative expression of the data [24].

To our knowledge, some related surveys have worked on how to model the temporal dependencies from traffic data, how to describe the spatial correlations for traffic networks; and how to mine the spatio-temporal relationships in traffic domain [16], [17], [18], [19], [20], [21], [22]. However, the perspectives of all the surveys were from technical application - providing the differences of deep neural network models and introducing the related applications. They mostly lacked the relationship between real-world needs and applied technologies.

Moreover, the researchers explored and compared different types of NN-based temporal methods used in traffic flow prediction [16], [17], [18], [19] but lacked the discussion of spatial dependencies extraction. Even though some scholars mentioned spatial correlations, they only provided related techniques for extracting spatial features instead of discussing the link between detailed spatial needs and why these spatial features can be captured by the model [20], [21], [22]. Luo and Zhou [25] presented a review of the NN-based ensembles, while did not show the overview of the techniques applied for the commonalities and differences of the Spatio-temporal dependencies. In Table 1, we showcase the comparison of different models in term of temporal correlations.

After researching the solutions to traffic flow forecasting, we emphasize the importance of understanding the problem and exploring the influential characteristics of traffic observers in this survey. We believe the key to obtaining the appropriate solution for traffic flow forecasting lies in deeply understanding the problem and choosing a proper model to illustrate the dependencies to the fullest, rather than solely considering applying advanced models as the solution. This article builds upon the research conducted in [26], offering a detailed examination of the spatial and temporal dependencies in traffic flow forecasting, the related NN-based models applied to depict the corresponding dependencies, and the Spatio-temporal ensembles on modeling instead of simply concerning the advanced NN-based models to solution. As for the dataset, most of the surveys have already mentioned the traffic dataset [16], [18], [19], [20] used for modelling traffic flow forecasting, we will not repeat to list the available dataset in this survey again, since they are already presented in aforementioned references.

Since the solution to traffic flow forecasting aims to mine as many as detailed features that can be predicted, we define a good solution to the problem by depending on 1. a deep understanding of the real scenarios and analyze the related dependency; 2. picking appropriate models to extract the corresponding dependency 3. propose a proper architecture to fuse spatial and temporal ensembles. To sum up, the main contributions of this paper would be:

- To our knowledge, we are the first to categorize and summarize the fine-grained factors affecting traffic flow forecasting, such as periodic temporal dependency, temporal dynamics, geographical scales, spatial heterogeneity, etc., instead of the general spatial and temporal factors provided by other related surveys.
- We summarize the NN-based models based on their ability to extract the categorized fine-grained dependency, explaining why the models work. We want to save researchers time in getting familiar with previous traffic flow forecasting solutions and inspiring their future

TABLE 1. Summary the main content of related surveys.

Ref.	Temporal models	Spatial models	Architecture	Others
[16]	CNN/RNN/LSTM/FNN/	-	-	
[17]	CNN/RNN/LSTM/GRU DBN/FNN	CNN	-	
[18]	CNN/DBN/FNN/RNN	GCN	AutoEncoder	
[19]	CNN/RNN/LSTM/GRU/DBN	-	AutoEncoder	
[20]	RNN/LSTM/GRU/TCN	GNN/GCN	Seq2Seq	Fine-grained dependency
[21]	-	GNN/GCN	-	
Ours	RNN/LSTM/GRU/TCN	GNN/GCN/GAT	AE/Seq2Seq/Transformer	Fine-grained dependency/ architecture of Spatio-temporal ensembles

approaches. This in-depth investigation of complex traffic domain dependencies has yet to be shown in previous surveys.

- we construct a comprehensive review of feature/ ensembles in Spatio-temporal architecture. After reading the survey, researchers may be inspired to propose new architectures of the Spatio-temporal dependency in their solution.
- we finally highlight the future directions that should gain further attention on the traffic flow forecasting. We point out the potential improvement in the reconstruction of graph structure and the application of the multi-source spatial and temporal dependency. We also discuss the diverse representation learning of the feature to express richer information and whether the weights should be considered in applying spatio-temporal dependency.

The rest of the paper is organized as follows: Section II introduces the definition development in the traffic flow prediction, from the sole temporal domain to the spatio-temporal domain; and the varying techniques considered during each stage. Section III elaborates fine-grained spatial and temporal dependency that current researchers considered that would affect the forecastings. We categorize it under temporal, spatial, and spatio-temporal dependency. We also state multiple NN-based techniques for modeling these specific fine-grained dependencies. Section IV provides the state-of-art approaches to modeling multiple temporal and spatial dependencies and new perspectives of the solutions. Section V shows the future direction of the solution to traffic flow forecasting. We demonstrate the potentiality of the reconstruction of the graph structure, the weights when considering the temporal, spatial, and spatio-temporal features in the model, getting the multi-source spatial and temporal correlations involved in the model, and more powerful representation learning on feature expression.

II. PROBLEM DEFINITION AND THE PERSPECTIVES FOR SOLUTIONS

In this section, we define the problem based on the time-series issue and introduce spatial dependency into the modeling. In Tables 2 and 5, we demonstrate the corresponding techniques in temporal and spatial dependency modeling.

In Tables 3, 4, and 6, we present the hyperparameters that should be considered or fine-tuned when modeling. This is because we believe that the good performance of the models depends on the informative dependencies selected, the appropriate models chosen for feature extraction, and the suitable parameters that fully exploit the models' capabilities.

A. PROBLEMS DEFINED WITHOUT SPATIAL DEPENDENCY

The traffic flow forecasting problem can generally be defined as a time series problem since it aims to predict future traffic indicators over time steps. The historical traffic indicators can be denoted as $X = [x^{t-T+1}, \dots, x^t]$ over the past T time slots, and the future traffic predictors over the next T' time slices is represented as $Y = [y^{t+1}, \dots, y^{t+T'}]$. Given the relevant historically observed data until time slice t, the $t+T'$ th-step traffic flow forecasting problem can be stated as

$$\begin{gathered} Y = F(X) \\ [y^{t+1}, \dots, y^{t+T'}] = F([x^{t-T+1}, \dots, x^t]) \end{gathered} \tag{1}$$

where

$$T' = \begin{cases} 1, & one - step\ prediction \\ n, & multi - step\ prediction \end{cases}$$

Some statistical theories, including the history average ARMA model, Kalman filtering model, linear regression, and non-parametric regression [44], were the first to be introduced into traffic flow forecasting. These models are relatively simple due to the assumption that future conditions will exhibit the same patterns and characteristics as historical flow data [45].

However, as a nonlinear time-varying system, real-world traffic conditions are significantly affected by complex and irregular previous states on the road. Additionally, the variables in statistical approaches are manually selected, relying on domain knowledge. Moreover, because of outdated equipment used for collecting traffic data, the usable observations in the traffic domain are limited; only part of the traffic conditions can be reflected due to bias. Therefore, classic mathematical models are adequate for simple tasks and limited datasets. However, due to complicated real-world conditions and requirements for robustness and accuracy, a more reliable approach is needed.

TABLE 2. Summary of models that describe temporal correlations behind the traffic flow data.

Model Category	Gradient Vanishing	Long-distance information	Time cost	Reference
RNN	Yes	No	Gradient vanishing; Not able to capture the long distance time-series information	[27]
LSTM	No, the gated mechanism to avoid gradient vanishing	Yes, adding cell state to store the long distance information	Yes, the massive parameters	[28] [29] [30] [31]
GRU	No, gated mechanism to avoid gradient vanishing	YES	No, fewer parameters	[32] [33] [34] [35] [36]
CNN	No	Yes, by stacking layers	No, the parallel computing for the architecture	[37] [38] [39]
TCN	No, residual connections	Yes, dilated convolution	NA, the longer the sequence, the better the efficiency	[40] [41] [42] [43]

Machine learning techniques are regarded as a new solution for modeling time-series traffic data following classic statistical approaches. Compared with statistical methods, Yang et al. [46] demonstrated the impressive ability of data-driven strategies to model non-linear correlations. Data-driven approaches, such as K-nearest neighbor (KNN) [47], support vector machine (SVM) [15], [48], [49], random forest, and NN-based models, have shown potential in handling complex traffic conditions. Machine learning algorithms outperform in capturing non-linearity and high-dimensional changes due to their capacity to mine information and understand the underlying relationships between historical traffic data and future data. Additionally, machine learning models, such as SVM [50], achieve higher accuracy with limited datasets.

However, realistic problems must be considered when applying machine learning techniques to traffic flow prediction. First is feature quality. Since machine learning models require manual feature selection before use, inappropriate feature selection can lead to unpredicted errors. Second is time cost. Due to their structure, machine learning methods cannot utilize traffic data efficiently. Furthermore, with advanced intelligent transport equipment and various data sources, such as video image processors, radar sensors, GPS, cellular phones, social media, etc. Reference [45], dealing with "Big Data" efficiently remains a challenge for machine learning methods [2]. Hence, the introduction of NN-based models in traffic flow prediction is expected due to their powerful ability to derive time series patterns from large amounts of data.

1) RNN

Multiple NN-based networks, such as the classic recurrent neural network (RNN), LSTM, and GRU, have been adapted to the traffic prediction task because of their capability to model basic temporal dependencies in traffic flow data.

In a recurrent neural network (RNN), the input of an RNN layer contains information (X) at the current time step t and a hidden state (h) that stores the historical information corresponding to the previous sequence in $t-1$ time steps. Hence, RNN is better at handling the explicit temporal order from consecutive traffic records. Assuming the historical indicators over T time slices in sequence as $X = [x^{t-T+1}, x^{t-T+2}, \ldots, x^{t}]$, we can have the hidden state for the next step as:

$$h^t = \sigma(U \cdot x^t + W \cdot h^{t-1} + b) \tag{2}$$

where U, W, and b are the learned weights and bias, and σ is the activation function. To obtain the output of the current time step, we have:

$$y^t = \sigma(V \cdot h^t + c) \tag{3}$$

where V is the weight to be learned and c is the bias. The structure of RNN is shown in Fig.1. The RNN network has shown better performance than regression models and other machine learning time series models in expressing the context of observations over time [28].

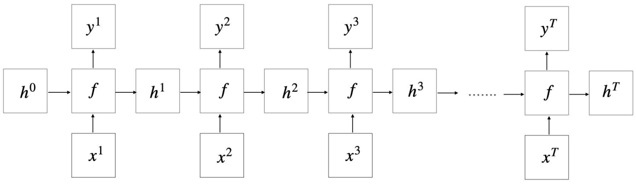

FIGURE 1. The structure of RNN model.

However, as we can see from Fig. 1, in RNN, a neuron can only accept information from the previous time step. In other words, the distance between the current state and the corresponding time step should not be too far. It is important to note that time-series forecasting in real-world traffic flow prediction focuses on long-term temporal dependency. For example, traffic flow consistently repeating patterns over time show periodicity, leading to the expectation that the model should learn information over a long distance under certain conditions. Moreover, the existing vanilla RNN faces the problem of vanishing gradient–the amount of extracted information may decrease exponentially as the distance of time steps increases due to errors in the obtained training parameters.

2) LSTM

The RNN variant, long short-term memory (LSTM) network, was proposed [51] to address the inherent deficiencies of RNN. First, it was developed to deal with the vanishing gradient and exploding gradient problems of the classic RNN during the back-propagation process. Additionally, while RNN only considers the state at the most recent moment, LSTM adds a filter function to the past state based on RNN, thus preserving the long-term sequential information that has more influence on the current moment instead of simply choosing the most recent state. Furthermore, LSTM employs the sigmoid function as a gate to filter the state or input to control the state of the transmitted information, enabling better performance. In essence, LSTM can store memory from a longer distance, retaining information over a long time and overlooking unimportant information. It performs better in processing time-series traffic data over long distances [52]. The basic architecture of LSTM is similar to RNN but introduces a cell state to store historical information and adapts a forget gate to control the past information from the previous unit, as seen in Fig. 2. The architecture of LSTM contains three gates to process the time-series data: forget gate, input gate, and output gate.

3) LSTM

The RNN variant, long short-term memory (LSTM) network, was proposed [51] to address the inherent deficiencies of RNN. First, it was developed to deal with the vanishing gradient and exploding gradient problems of the classic RNN during the back-propagation process. Additionally, while RNN only considers the state at the most recent moment, LSTM adds a filter function to the past state based on RNN, thus preserving the long-term sequential information that has more influence on the current moment instead of simply choosing the most recent state. Furthermore, LSTM employs the sigmoid function as a gate to filter the state or input to control the state of the transmitted information, enabling better performance. In essence, LSTM can store memory from a longer distance, retaining information over a long time and overlooking unimportant information. It performs better in processing time-series traffic data over long distances [52]. The basic architecture of LSTM is similar to RNN but introduces a cell state to store historical information and adapts a forget gate to control the past information from the previous unit, as seen in Fig. 2. The architecture of LSTM contains three gates to process the time-series data: forget gate, input gate, and output gate.

The output of the forget gate is f^t, which represents the probability of forgetting the previous hidden state:

$$f^t = \sigma(U_f \cdot x^t + W_f \cdot h^{t-1} + b_f),$$

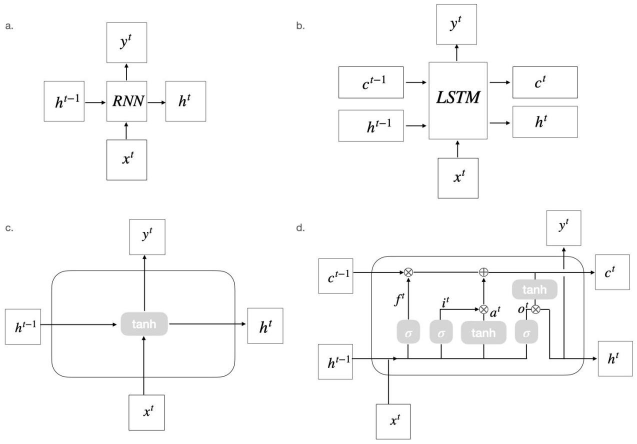

FIGURE 2. The structure comparison of RNN and LSTM model. Both of RNN and LSTM have hidden cells to retain the information from previous intervals where the LSTM introduces a cell state to remove or add information to the memory.

where the parameters are similar to those in RNN. The activation function σ is usually *softmax*, thus, leading to the output of the forget gate being between [0, 1]

The input gate has two components:

$$\begin{aligned} i^t &= \sigma(U_i \cdot x^t + W_i \cdot h^{t-1} + b_i) \\ a^t &= tanh(U_a \cdot x^t + W_a \cdot h^{t-1} + b_a), \end{aligned} \tag{4}$$

where U_i, W_i, b_i, U_a, W_a, b_a are the weights and bias.

The cell state is updated with the obtained output from the forget gate and input gate as:

$$C^t = C^{t-1} \odot f^t + i^t \odot a^t,$$

where $\odot$ is the hadamard product.

Hence, we can have the output of the current time step as:

$$\begin{aligned} o^t &= \sigma(U_o \cdot x^t + W_o \cdot h^{t-1} + b_o) \\ h^t &= o^t \odot tanh(C^t), \end{aligned} \tag{5}$$

4) GRU

Another variant of RNN is the gated recurrent unit (GRU), which is proposed with a simpler but more efficient structure than both LSTM and RNN. It has fewer parameters but shows the same impact as LSTM in temporal modeling [20]. GRU has a similar simple structure to RNN but with different operations inside the GRU unit: two gates–reset gate and update gate. The reset gate controls the importance of the hidden state (0/1), and the update gate decides if the previous cell state should be updated with the current hidden state. The difference between standard LSTM and GRU is not significant, so the choice between LSTM and GRU depends on the specific task.

$$\begin{aligned} r^t &= \sigma(U_r \cdot x^t + W_r \cdot h^{t-1} + b_r) \\ z^t &= \sigma(U_z \cdot x^t + W_z \cdot h^{t-1} + b_z) \\ o^t &= \sigma(U_o \cdot x^t + W_o \cdot h^{t-1} + b_o) \\ h^t &= z^t \odot o^t + (1 - z^t) \odot h^{t-1}, \end{aligned} \tag{6}$$

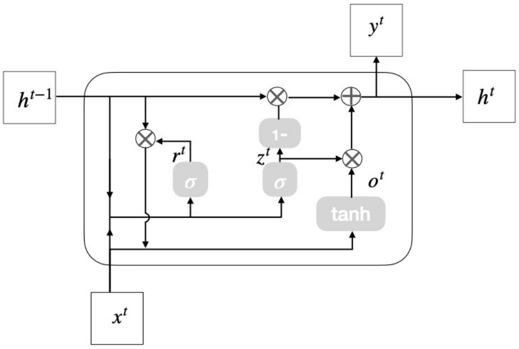

FIGURE 3. The cell structure of GRU.

5) CNN

In time series models, recurrent neural networks (RNNs) are a traditional approach that gathers global information sequentially through recursion without the use of parallel computation. Sequential data may also be modeled using convolutional neural networks (CNNs) with a two-dimensional "block" (mn matrix). Time series data can be considered a one-dimensional object (1n vector). A sufficiently large receptive field can be attained using CNN's multi-layer network structure, although this requires significant time due to the multiple layers.

6) TCN

However, due to the advantage of processing in a large-scale parallel structure and increasing the speed of network training, Duranton et al. [40] adapted the temporal convolutional networks (TCN) model to predict traffic flow. The TCN model is based on the CNN model and incorporates improvements in causal convolution, dilated convolution, and residual connections [53]. Results showed that TCN outperformed the latest LSTM and typical GRU models. Additionally, TCN avoids the issues of gradient dispersion and gradient explosion in RNN, making it more adaptable in processing various lengths of historical information [41], [42]. An example is shown in Fig. 4.

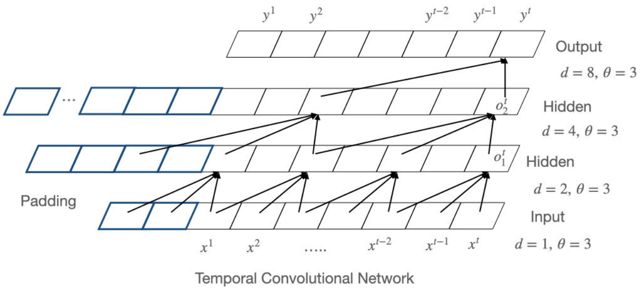

FIGURE 4. The structure of TCN.

Assume we have the time series data as $X = [x^{t-T+1}, x^{t-T+2}, \ldots, x^t]$ to make the prediction in $Y = [y^{t-T+1}, y^{t-T+2}, \ldots, y^t]$, with the filter as $F_f = (f_1, f_2, f_3)$ since $\theta = 3$. Hence, the hidden state in the first hidden layer at time step t with causal convolution is

$$\begin{aligned} o_1^t &= F_f \odot X = \sum_{k=1}^{K} f_k x_{t-K+k} \\ &= f_1 x_{t-2} + f_2 x_{t-1} + f_3 x_t. \end{aligned} \tag{7}$$

Since the size of the convolution kernel determines the amount of information extracted, TCN introduces dilated convolution to the model to increase the receptive field and avoid losing information. The dilated convolution is achieved by inserting holes into the "block" to expand the receptive field. It introduces a dilation rate, which refers to the number of kernel intervals (the dilation rate in standard CNN is 1). The advantage of dilated convolution is that it expands the receptive field without losing information, as it avoids the pooling operation used in CNN. For example, if the dilation rate in the first hidden layer is 2, the dilated convolution in the second hidden layer is:

$$\begin{aligned} o_2^t &= F_f \odot o_1^t = \sum_{k=1}^{K} f_k o_{1_{t-(K-k)d}} \\ &= f_1 o_{1_{t-2d}} + f_2 o_{1_{t-d}} + f_3 o_{1_t}, \end{aligned} \tag{8}$$

The receptive field of which is $(K - 1)d + 1$. However, the convolution operation solely focuses on capturing local information, requiring the stacking of layers to achieve a larger receptive field and capture richer global information. Residual connections are introduced into the model to solve the degradation problem. The degradation problem occurs when deeper networks, which enlarge the receptive field, cause the accuracy of the training set to stabilize or even drop. This makes the mathematical solution space more complicated, leading to the stochastic gradient descent method failing to achieve global optimization and getting stuck in local optima. The deeper the network, the more abstract the features and the more semantic information obtained. The residual connections of TCN avoid problems that exist in deeper layers, such as gradient dispersion, gradient explosion, and performance degradation [54].

Besides the aforementioned models, we also want to discuss several model architectures for time series forecasting, such as Seq2Seq and Transformer.

7) SEQ2SEQ

Seq2Seq is a typical architecture under the Encoder-Decoder structure, as seen in Fig. 5. While RNN requires the input and output to have the same length, Seq2Seq provides a solution for unequal lengths of input and output. c is the latent vector that contains the transformed information of the input and can be converted to the output sequence, which is:

$$\begin{aligned} h_t &= f(x_t, h_{t-1}) \\ c &= q(h_1, h_2, \ldots, h_t) \\ h'_n &= g(c, h'_1, h'_2, \ldots, h'_{n-1}) \\ y_n &= G(h'_n) \end{aligned} \tag{9}$$

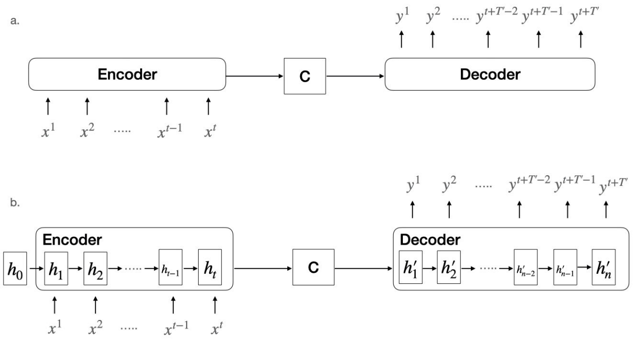
a.
Encoder
C
Decoder
x^1 x^2 ..... x^{t-1} x^t
y^1 y^2 ..... $y^{t+T'-2}$ $y^{t+T'-1}$ $y^{t+T'}$
b.
Encoder
h_0 h_1 h_2 h_{t-1} h_t
C
Decoder
h'_1 h'_2 h'_{n-2} h'_{n-1} h'_n

FIGURE 5. The structure of Seq2Seq.

There are also drawbacks of Seq2Seq when applying it to time-series data:1. Because the length of context x is pre-defined and fixed, information can get lost during compression; 2. The impact of each input to the target y_i is treated as the same, while in reality, it is different. For example, the conditions in past time steps affect the next steps differently, with the influence of the nearer ones outweighing that of the distant ones. Bahdanau et al. [55] employed the attention mechanism to address the different importance of the target in Seq2Seq. The attention mechanism learns the attention coefficients of each input from the sequence and then merges them according to their importance.

Compared with the original Seq2Seq model, every h is calculated based on the same context c, attention mechanism generates a different context c_n at each time step to solve this problem (Fig. 6).

$$c_n = \sum_{i=1}^{m} \alpha_{ni} h_i, \tag{10}$$

where α_{ni} is used to measure the influence of the h'_n in the decoder to the hidden state h_i in encoder at the time step i. The weight $\alpha_n i$ is:

$$\alpha_{ni} = \frac{exp(score(h'_n, h_i))}{\sum_{k=1}^{m} exp(score(h'_n, h_k))}, \tag{11}$$

where $score(h'_n, h_i)$ is to calculate the similarity of the hidden state h'_n in the decoder and h_i in the encoder. Finally, we can have the output of the Seq2Seq model with attention mechanism in:

$$\begin{aligned} h'_n &= g(c_n, h'_1, h'_2, \ldots, h'_{n-1}) \\ y_n &= G(h'_n) \end{aligned} \tag{12}$$

However, Seq2Seq only pays attention to the relationship between the input and the output, ignoring the position information. This means that the positional information in the time series data cannot be captured by Seq2Seq.

8) TRANSFORMER

Google proposed an architecture named "Transformer" [56], which offers a different approach to time-series modeling based on the Encoder-Decoder structure. The Transformer relies on the attention mechanism to obtain global information in one step. Previously, researchers captured global features by applying CNN in multiple layers. The Transformer incorporates position embedding and a self-attention mechanism to obtain hidden states instead of the recursion structure used in RNN-related models. The application of the attention mechanism provides the Transformer with powerful capabilities for time series forecasting [57], thanks to simultaneous improvements in modeling long-term and short-term temporal correlations in time series data based on the multi-head attention structure.

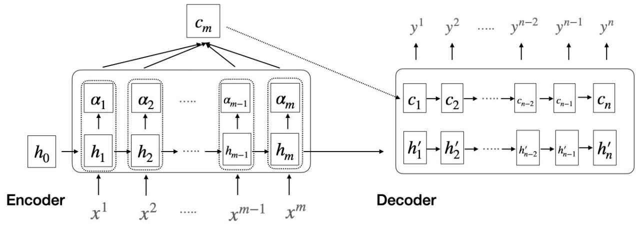
c_m
y^1 y^2 ..... y^{n-2} y^{n-1} y^n
α_1 α_2 ..... α_{m-1} α_m
h_0 h_1 h_2 h_{m-1} h_m
c_1 c_2 c_{n-2} c_{n-1} c_n
h'_1 h'_2 h'_{n-2} h'_{n-1} h'_n
Encoder
x^1 x^2 ..... x^{m-1} x^m
Decoder

FIGURE 6. The structure of Seq2Seq with attention mechanism.

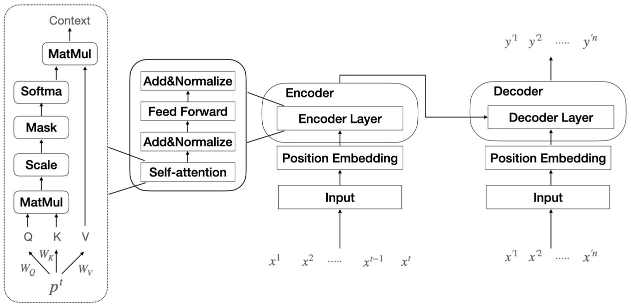

FIGURE 7. The structure of Transformer.

The attention mechanism behind the transformer is similar to that in Seq2Seq. We assume the time-series input $X = [x^{t-T+1}, \ldots, x^t]$ over the past T time slots to predict $Y = [y^{t+1}, \ldots, y^{t+T'}]$. The input layer converts the data to the required vector, and the position-embedding layer then encodes the sequential information of it with the sin-cos functions as $[p^{t-T+1}, \ldots, p^t]$. In the architecture shown as Fig. 6, the self-attention mechanism mentions the key(K), value(V), and query(Q) that are transformed by the matrices W_k, W_v, W_q which are calculated with the vector V. The idea of K, Q, V is from the retrieval system as querying the target Q from the pairs (K, V). Hence, the attention weights:

$$\begin{aligned} \alpha_{ni} &= softmax(\frac{QK^T}{\sqrt{d_k}}) \\ &= \sum_{i=1}^{t} exp(\frac{< q_n, k_i >}{\sqrt{d_k}}) \end{aligned} \tag{13}$$

The calculation of the alignment score of q_n and all the k can be done by obtaining the similarity of the record at time step n with all the records from the selected time period.

TABLE 3. Summary of the hyperparameters that should be chosen/fine-tuned in temporal modeling.

Models	Common Hyperparameters	Specific Hyperparameters
RNN	Number of Layers/Learning Rate/Batch Size/ Activation Function/Dropout Rate/Input Size	Number of Hidden Units
LSTM		Hidden Units/Recurrent Activation Function/Bidirectional
GRU		Hidden Units/Recurrent Activation Function/Bidirectional
CNN		Number of Filters/Kernel Size/Pooling Strategy
TCN		Dilation Rates/Number of Channels(Stacks/Layers per Stack)/Kernel Size

TABLE 4. Summary of the hyperparameters that should be considered in architecture.

Architectures	Hyperparameters
Seq2Seq	Number of Encoder and Decoder Layers/Hidden Units/Activation Function/Bidirectional
Transformer	Number of Transformer Layers/Attention Heads/Hidden Units/Positional Encoding Type

The $\sqrt{d_k}$ term is used to keep the gradient stable. Using the probabilities obtained by the softmax function to calculate the weighted average of current V at all times, the output is the vector that contains the global information at all time steps according to the self-Attention mechanism. Thus, the results of the self-attention layer are $s_n = \sum_{i=1}^{t} \alpha_{ni} v_i$. The multi-head attention mechanism helps the network capture richer features/information under different subspaces, similar to applying the multiple convolutional kernels in CNN. It works by repeating the self-attention in h times, as follows:

$$Multihead(Q, K, V) = Contact(head_1, head_2, \ldots, head_h), \tag{14}$$

where each head is the output of the self-attention.

Wang et al. [58] used the extracted spatial and temporal correlations in the transformer architecture to obtain the long-term temporal features of the traffic flow data. A residual connection and layer normalization are also applied in the sub-layer to enhance the features.

B. PROBLEMS DEFINED WITH SPATIAL DEPENDENCY

With further research, the spatial information on the traffic road network leads to the success of the work in traffic flow prediction [13]. Hence, how to mine the non-linear spatial and temporal relationships behind the traffic data is a challenge. Since the deep learning methods contributes to mining the non-linear information efficiently, many deep-learning-based solutions are applied in the traffic domain, such as traffic flow forecasting, accident detection, and abnormal detection, to extract the spatial dependency. Given the previous definition, which only considers the temporal information in traffic data, the solution here that considers spatial context can be represented as S, at time slice t as:

$$Y = f(X, S) = f([x^{t-T+1:t}], S).$$

1) CNN

Before, to depict the topology of the traffic network, the road segments were regarded as the 2D structured data consisting of latitude and longitude, which was inspired by applying CNN to exploit the Euclidean traffic flow data. Zheng et al. explored the traffic flow graph structure by dividing the route into more fine-grained sections and adapting the CNN on the pre-processing matrix [63]. CNN has the sub-sampling (max-pooling) layers specially designed to reduce the traffic map into the sub-grid structure. In addition, it can learn the characteristics of the relations within the grid parts from the locality, and then combine the extracted local context into a high-level representation. CNN receives success in capturing local spatial dependency.

Du et al. [64] discussed the impact of the locality on the traffic flow forecasting problem and proposed a hybrid model combining LSTM and CNN. It considers the performance of CNN to handle the local information in the sub-grid structure. This model employed the LSTM to capture the long-term temporal information, and CNN for the local dependency. The extracted information was fused before feeding into the model. Henceforth, the results showed that the consideration of the locality feature could contribute to the results in satisfactory accuracy and effectiveness even under complex non-linear urban traffic conditions.

In general, Convolutional Neural Network (CNN) models the locality by decomposing the traffic network into grids [59] but ignoring the topological structure graph-wide of the transportation network because of the complicated spatial correlations in reality. In other words, Euclidean structure cannot fully describe spatial correspondences.

For example, two road sections are very close in Euclidean space, but the relations of a pair of road links may be in opposite directions due to the topology. So, it is hard to model completely different traffic flow patterns under CNN. This means that the spatial structure in traffic is non-Euclidean and directional which requires the models to be able to capture the non-Euclidean topological information. To discover and make use of the spatial dependence of traffic flow data for better prediction, a more appropriate method that can express the traffic network mathematically into a graph is required,

TABLE 5. Summary of spatial models.

Spatial / Temporal	Model Category		Data structure	Aggregater	Reference
Spatial	CNN		Grid	Convolutional Aggregator	[66]
	GNN		Graph	Massage passing	[61]
	GCN	Non-spectral	Graph	Convolutional Aggregator	[75]
		Spectral	Graph	Spectral filters	[37] [32]
	GAT		Graph	Attention Aggregator	[63] [24] [38]

TABLE 6. Summary of the hyperparameters that should be chosen/fine-tuned in spatial modeling.

Models	Common Hyperparameters	Specific Hyperparameters
GNN	Learning Rate/Optimizer/Number of Layers/Epochs/ Batch Size/Dropout Rate/Activation Function/Hidden Units/ Graph Connectivity/Node Feature Input/Graph Density	Message Aggregation
GCN		Number of Filters
GAT		Number of Attention Heads/Attention Heads/Attention Dropout

thus leading to the application of graph-based deep learning methods in traffic flow prediction.

Compared to Euclidean-structured data such as images, graph-structured data presents more complexity in modeling. Firstly, the size of non-Euclidean graph data varies with each input during every time slice, posing challenges to the application of CNNs for graph data. For example, in image processing, input images have a fixed size, making it straightforward to manually predefine the size for each input, as the neighbors of a target node are at a fixed distance. In essence, once the center node is identified, the neighboring nodes are also determined. However, with graph-structured data, defining a fixed input size is challenging due to the variability in the number of neighbors each node has and their multiple distances. Additionally, graph-structured data lack a strict order, unlike grid-structured data, which can be sequentially processed in CNNs. Mathematically, the feature matrix dimensions of each block in a graph differ from those in Euclidean-structured data, such as grid-structured image data. As a result, unified operators used in CNNs cannot directly perform operations like convolution and pooling on graph-structured data. This limitation has spurred discussions on handling characteristics like sparse connectivity, weight sharing, and feature extraction in graph-structured data processing.

Since the traffic network is usually graph-structured, the transition between the traffic states of the target node in the whole network can be defined as a graph Markov process. In order to handle missing values in the obtained raw data while anticipating short-term traffic, Williams et al. introduced the graph Markov network (GMN), particularly in the context of edge computing and online learning [65].

At the end of 2018, scientists from companies and institutions such as DeepMind, Google Brain, MIT, and the University of Edinburgh jointly proposed the concept of a graph neural network [66]. Based on Graph Neural Network (GNN), the traffic network can be regarded as a natural graph, where each observation is a node and the connection is the edge. The definitions of a graph are as follows: $G = (V, E)$

2) GNN

An unweighted target traffic graph can be constructed as $G = (V, E)$, where, V is a set of nodes on the traffic graph, and E is a set of edges. $V = \{v_1, v_2, \ldots, v_N\}$ refers to N nodes on the graph, and each node represents an observation. The edge E refers to the connection between two nodes, which is represented by the adjacency matrix $A \in R^{N\times N}$, containing the topological information of the road network. If the targets are connected to the adjacent ones, the element of the matrix is 1, otherwise, it will be 0. For each node i, it has its own characteristics x_i, which can be represented in a feature matrix X_{N*D}. For the feature matrix X_{N*D}, N represents the number of nodes, and D is denoted as the number of features of each node and can also be regarded as the dimension of the feature vector. The problem definition with spatial correlations can be updated as:

$$\begin{aligned} Y &= f(X, G(V, E)) \\ &= f([X^{t-T+1:t}], G(V, E)) \end{aligned} \tag{15}$$

Besides the feature matrix, an adjacency matrix plays an important role in accurate spatial modeling. It is because the expression of the node feature highly relies on the information from the neighbors. The adjacency matrix is the key to describing the spatial relations of the nodes in the graph [22]. In the traffic domain, because of the different assumptions in the specific scenario, the adjacency matrix is classified as a fixed matrix and dynamic matrix with the spatial structure whether changing to the evolving over time [20].

Given the previous achievement in traffic flow prediction, to better describe the topological structure of the traffic graph, many researchers tend to extend generalizing the convolution of CNN in 2-dimensional data to the graph-structured data, which leads to the introduction of the convolution operation applied on graph neural network(GNN).

In image processing, the convolution operation on the 2-d data is to extract features from an input image, which preserves the hidden relationship between pixels. The core technique of graph convolution network(GCN) is also the convolution operation, which is the same as CNN. Hence, the characteristics of CNN on the structural level are also of great significance to GCN [67]: (1) A graph is neural in sparse connectivity. (2) The time cost of the neural network can be reduced through weight sharing. (3) Multiple layers denote the features extracted at a different level. However, compared with CNN, a drawback of GCN is also obvious - it is difficult to define the local kernels and pooling operations in a graph directly because of the different number of neighbor nodes around each target, which leads to the discussion of the methods on how to aggregate information to a central node in a graph.

The GCN is typically classified as spatial GCN and spectral GCN. The spatial GCN works on incorporating the properties of a node based on its k local neighbors by directly multiplying the adjacency matrix to extract the features where the spectral version transfers the adjacency matrix to the Laplacian matrix to return the Fourier basis for the graph to capture the features. The mainstream of the GCN in the traffic domain are spectral graph convolution and diffusion graph convolution [20], the variant of vanilla spatial GCN. Essentially, all kinds of spatial graph-based neural network models are differentiated on aggregating approaches within each layer, while the spectral graph-based neural network models are on the choice of the filter g_θ [22].

The update operation for each layer is to update the states of the nodes in the current layer to the next status, which, in the graph-based neural network, can be written as a nonlinear function [67]:

$$H^{l+1} = f(H^l, A). \tag{16}$$

A is the adjacent matrix of the traffic graph. H^l represents the information of all nodes in layer l that should be updated, and H^{l+1} is the updated layer. f is the update function. It could be the GRU function in Gated Graph Neural Network(GGNN) [68], message function in Message Passing Neural Network(MPNN [69]) or activation function in GraphSage [70] as the model required.

In the spectral domain, the basic idea of convolution operation is to first convert the signal from the spatial domain to the spectral domain by the Fourier Transform as $f(t) - > f\hat{(t)} = U^T f(t)$, and multiply it with the convolution kernel $g_\theta(\lambda)$ in the spectral domain. And then transformed the outcome back to the spatial domain through the Inverse Fourier Transform. This is because the convolution operation in the spatial domain is the same as the multiple operations in the spectral domain. The extracted feature in the spectral domain is $\hat{y} = f\hat{(t)} \cdot g_\theta$. Since we are working on the dependency in the spatial domain, the extracted feature should be converted by the Inverse Fourier Transform as $y = U\hat{y}$, where U is the eigenvalue that is related to the Laplacian matrix of a graph. The place matrix is an important matrix used in graph theory. In a graph $G = (V, E)$, the normalized Laplacian matrix of a graph is defined as $L = D - A$, where D is the degree matrix of the graph, and A is the adjacency matrix of the graph. Hence, can have

$$H^l = \sigma(LH^{l-1}W^l) \tag{17}$$

to describe the update operation in the spectral domain, where W^l is the weighted parameter matrix of the lth layer, and σ is a nonlinear activation function, such as ReLU. The Laplacian matrix L is directly multiplied by the feature matrix H. But there are two problems with this operation: it ignores the impact of the node itself. We can have $L^{sym} = D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}} = D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}} = I_a + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, which introduces the Symmetric normalized Laplacian to solve the self-transmission problem.

$$\begin{aligned} H^{l+1} &= \sigma(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}H^l W^l) \\ &= \sigma(D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}}H^{l-1}W^l) \\ &= \sigma((I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}})H^{l-1}W^l) \end{aligned} \tag{18}$$

is proposed to solve the mentioned problem.

But we take the original definition as an example. Since $L = U\Lambda U^T$, U is the eigenvalue and Λ is the eigenvector, we can expend the formulation:

$$y = U\hat{y} = Uf\hat{(t)} = Ug_\theta U^T f(t) \tag{19}$$

The hidden state of a node refers to the information of a node we mentioned above, for example, defined as $h_v^{t+1} = f(x_v, x_c, h_n^t, x_n)$, which represents the hidden state of node v in the layer $t+1$ is updated by the aggregation of the embedded node feature, the edge feature, the hidden state of the neighbors, and the edge feature of the neighbors. The specific node information depends on the specific requirement. Because the node information and the Laplacian matrix can be obtained directly from a graph, the difference of the spectral GCN lies in the filter g_θ.

For example, a filter in spectral GCN is Chebyshev polynomials [71]. The ChebNet method believes that the value of the convolution kernel in the spectral domain is a function related to the eigenvalue of a graph, which can be used to approximate the Chebyshev polynomials. Since $g_\theta = \sum_{k=0}^{K} \theta_k T_k(\Lambda)$, the function can be as:

$$\begin{aligned} y &= Ug_\theta U^T f(t) \\ &= U\sum_{k=0}^{K}\theta_k T_k(\Lambda)U^T f(t) \\ &= \sum_{k=0}^{K}\theta_k T_k(U(\Lambda)U^T)f(t) \\ &= \sum_{k=0}^{K}\theta_k T_k(L)f(t) \end{aligned} \tag{20}$$

Because the T_k is the Chebyshev polynomials, θ_k is the coefficient to be learned. The [72] is based on the Chebyshev polynomials, which can be regarded as a further simplification of ChebNet. It only considers the 1st-order Chebyshev polynomial and each convolution kernel has only one parameter. However, there is also a drawback of the spectral GCN, the definition of the Laplacian matrix limited the graph in an undirected structure. And it assumes the static graph under the situation, while in practice the graph always time-varying, especially in the traffic domain.

In the spatial domain, it can be seen as two steps: 1). aggregation, which refers to how the center node collects the information from the neighbors, as $h_v^{l+1} = f(h_u^{l+1})$, where u are the neighbors of node v; 2). update operation, which represents how the nodes update their states to the next level based on the information aggregated, as $h^{l+1} = g(h^l)$. Before the aggregation, the nodes on the graph get the feature representation after the feature embedding. The aggregating approach designed in the spatial GCN can be regarded as a message-passing process. The central node, in the aggregation process, will exchange the information with that in the nearby k-hops, and update itself until the equilibrium of all the nodes on the graph is reached. The aggregation process aims to get the new representation for each node on the map by considering the information of the target itself and its neighbors. For example, one of the simplest convolutional operations is to add the hidden states of all neighbors to that of the center node. As in spatial GCN, it is similar to CNN to extract the spatial features from the topological graph directly. The information updated in aggregating could be the feature of the target node n, the information of its neighbors, the edge feature between the node n and its connected neighbors, the hidden state of the neighbors, and the previous hidden state of the node n.

Some of the work considers the spatial graph-based neural network in their model [61], [73], [74]. However, because the traffic graph, in reality, is large, which could cause time-consuming problems when computing the multiplication of the adjacency matrix, it is not common to employ the vanilla spatial GCN in the large traffic graph in the graph-level solution. Unlike spatial GCN, which directly aggregates the information to get the new representations for each node one by one and then updates the states of all nodes in the layer, the spectral graph-based neural network applies the Laplacian matrix to get the signal in the spectral domain from the spatial domain and finally inverts it back to the spatial domain. This is because the convolution operation in the spatial domain is the same as that in the spectral domain, making feature extraction simpler and less time-consuming. They are applied in the node-level solution because the state h, the stacking of k layers of spatial GCN represent the k-hops of the target node when it comes to discussing the impact of the multi-hops on the central node, spatial GCN has been considered to be adapted [36], [75].

There are some restrictions in the GCN model when applied to traffic flow forecasting problems:

- The traffic flow dynamically changes all the time. But GCN cannot reflect the temporal dynamics of the traffic flow since the model construction relies on the static property of the graph.
- GCNs capture the spatial information from the neighbors with the same impact, which cannot reflect the different importance of neighbors to the central node. For example, nodes on busy commuter lines weigh more heavily than those that are not on the line. Additionally, in a road section, the inflow and outflow of the same node could be contrary during the morning rush hour and the night rush hour. Hence, it is necessary to describe the different importance of neighbors to the central nodes from the spatial perspective.
- The aggregation approach of GCN depends on the graph, or the matrix generated from the traffic network. This limits the generalization from one region to others.

GAT: Graph attention neural networks have been introduced to solve the above problems based on the attention mechanism [76]. They consider the weighted summation of the features among neighboring nodes. The weights depend entirely on the neighboring nodes, independent of the graph structure. The attention mechanism has been developed to improve the performance of models discovering the spatio-temporal relation in networks [77].

$$alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} \tag{21}$$

where α_{ij} is the obtained attention coefficient of node i connected with node j among all its connections.

Both GCN and GAT focus on aggregating the features of neighboring nodes to the central node and learning new feature expressions of the node based on the local stationery of the graph. The core difference between GCN and GAT is that GAT employs a single attention mechanism to aggregate the information with different weights. We can notice a normalized constant in GCN that is designed based on the graph structure. And because so, it is also why the generalization ability of the GCN model is unsatisfactory. In essence, GAT replaces the normalizing constant in GCN by using the attention weights to contribute to aggregating the neighboring node feature. It achieves to assign different weights to different neighbor nodes due to its impact on the target. To a certain extent, GAT will be powerful because the feature correlation between vertex is better described in the model without the effect of the graph structure.

As the original GAT adapted the single attention mechanism, Gated Attention Network (GaAN) Zhang et al. utilized the multi-head attention mechanism with the self-attention mechanism to gather information from different heads [78]. The idea behind the multi-head attention mechanism is that each attention head only focuses on one subspace of the input sequence, and is independent of each other. It is to obtain much richer information on features in the subspace.

TABLE 7. Categorization of the subcategory of the temporal/spatial dependency.

Spatial / Temporal	Dependency subcategory	Subcategory description	Model Category
Temoral	Short-term temporal correlations	Temporal information in 5-15 minutes	RNN/LSTM/GRU/TCN
	Long-term temporal correlations	Temporal information longer than 15 mins; Periodicity in hourly, daily, weekly	LSTM/GRU/CNN/TCN/Transformer
	Temporal dynamics	Temporal information shift	Attention mechanism/ Dynamic Time Warping (DTW)
Spatial	Spatial static information	Topology information	CNN/GNN
	Spatial dynamics	Distant neighbouring information	Attention mechanism
	Geographical scale	Neighboring scales inforamtion	Attention mechanism/GCN/CNN
	Regional Spatiality	Subgraph information	Attention mechanism

III. CLASSIFICATION OF THE DEPENDENCIES AND THE RELATED TECHNIQUES

We can see that there has been significant success in applying deep learning techniques to traffic flow forecasting problems in recent years. The key to the performance of traffic prediction models lies in the accurate representation of traffic features. Before extracting specific features, it is essential to analyze dependencies at a fine-grained level in detail, which can enrich the feature information for the final prediction. In Table 7, we compare different spatial traffic dependencies, temporal traffic dependencies, and spatio-temporal traffic dependencies at a fine-grained level in terms of discovering the multiple expressions of the features that can affect traffic flow forecasting. In this section, we will review the different perspectives of the spatial and temporal dependencies that researchers consider to solve the problem currently and introduce how they modeled these dependencies.

A. TEMPORAL DEPENDENCY

Traffic flow data naturally exhibits a sequential characteristic, meaning that previous states of traffic flow can directly lead to changes in future traffic states. Initially, researchers focused on depicting the temporal correlations behind the data to predict traffic flow accurately. Over the years, temporal dependency has been divided into two clear types: short-term temporal dependency and long-term temporal dependency. This division makes it easier to discuss detailed solutions for short-term flow forecasting and long-term flow prediction.

Short-term temporal correlation reflects how the traffic state of a previous time step can directly influence the state in the near future (within a 5-15 minute period). Long-term temporal correlation shows the periodicity of traffic states over more extended periods, such as a day or a week. Additionally, research on long-term temporal dependency has highlighted the importance of exploring relationships between the current time step and non-adjacent time steps.

1) SHORT-TERM TEMPORAL CORRELATIONS

Traffic flow shows significant short-term temporal relations. The task of short-term traffic flow forecasting is to predict the changes in traffic flow (e.g., flow, speed) of a road in the next few minutes. For example, due to current traffic congestion, the traffic condition in the next time slot is highly likely to be similar to the current time step since it is directly affected by the states of the last minute. Such strong short-term temporal dependency is the basic information used to predict future traffic flow. Since the connections between the units of the recurrent neural network (RNN) can form a directed loop, RNN and its variants (LSTM, GRU, etc.) are typically used to model short-term temporal data. In other words, RNN and its variants are powerful tools for capturing short-term temporal correlations in traffic data.

Due to the influences on more distant time states that can enhance the patterns of short-term temporal relations, some work has considered adapting the Long-Term Short-Term Memory (LSTM) network to replace classic RNN in modeling short-term temporal dependencies. This is because LSTM can take longer-period temporal information into consideration, while typical RNN only considers the context at the most recent moment. Based on the RNN structure, LSTM adds a filter function to past time states, aggregating the distant impacts on earlier time steps to address short-term correlations.

However, LSTM and GRU models cannot overcome the major drawbacks of RNN-type models, such as gradient vanishing and the computational time cost during the training process. To solve these problems, the Temporal Convolutional Network (TCN) architecture combines dilated convolutions and residual connections. The typical convolutional action, referring to the backpropagation algorithm, employs different paths from the temporal direction of the sequence, computing the gradient of the loss function concerning each weight by the chain rule to address these issues.

Even though TCN appears to be an outstanding architecture, Zhang et al. showed that the choice of the number of TCN layers requires further discussion for short-term flow prediction research [79]. On one hand, the shallow structure of the TCN architecture may struggle to capture complex temporal relationships, while deeper layers could result in overfitting.

Therefore, when considering short-term temporal correlations in traffic flow forecasting, the mentioned methods provide significant insights but should be tailored to specific conditions.

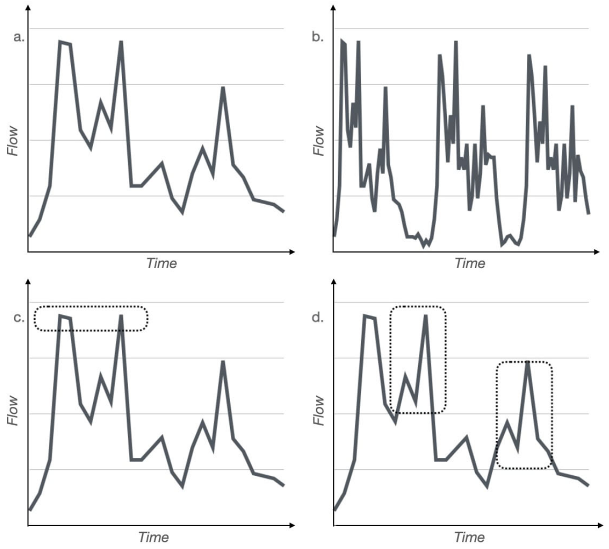

FIGURE 8. The temporal dependency in traffic flow forecasting. a. The short-term dependency in time series. b. The periodic patterns in the long-term dependency. c. The mismatching of the numerical value in time series. d. The local trend of dependency in the long term.

2) LONG-TERM TEMPORAL CORRELATIONS

Besides the direct impact on the traffic condition in the next few minutes, traffic flow data usually contains long-term information, such as fluctuations over an hour, a day, a week, or even seasonal periodic patterns over long temporal distances. As shown in Fig. 8(b), Zhang et al. demonstrated that exogenous information in more detailed segmentation, such as the time of day, weekday or weekend, and historical statistical information, contributes to long-term prediction [32]. However, this does not mean that longer prediction intervals always lead to better performance. Guo et al. showed that the difficulty of prediction increases with longer time intervals [37]. Additionally, Yao et al. pointed out that increasing the length of time steps can enlarge the risk of gradient vanishing, significantly weakening the effects of periodicity [59].

Variant LSTM models are naturally considered to capture long-term temporal features in many works because the memory cell is designed to maintain information over a long distance. However, LSTM and its variants face time-consuming issues due to the longer-period information stored. They also encounter gradient vanishing and explosion problems during the back-propagation process when training. Therefore, choosing between GRU and LSTM for modeling long-term temporal correlations involves a trade-off. GRU has a simpler structure and requires less training time, while LSTM can perform better in other cases.

Besides the general definition of long-term temporal dependency requiring inputs from more distant time steps, traffic flow also shows periodic characteristics within specific periods. Such periodicity refers to fluctuations over an hour, a day, a week, or even seasonal patterns over long temporal distances, as shown in Fig. 8(b). Zhang et al. demonstrated that exogenous information in more detailed segmentation, such as the time of day, weekday or weekend, and historical statistical information, contributes to long-term prediction [32]. For example, rush hour typically occurs every weekday around 9:00 a.m. and 5:00 p.m., which are peak times for most commuters. This daily and weekly periodicity captures essential temporal information for traffic flow. It should be a crucial factor in traffic flow prediction since it reflects the variability of traffic flow over the long term. Wang et al. [80] considered this situation and separately captured temporal information as closeness, period, and trend using the same ConvLSTM structures with different weights.

Wang et al. also embraced the idea of capturing multiple temporal components [81]. They proposed a dynamic mechanism to employ time-varying hypergraphs for capturing the hourly, daily, and weekly changes in the traveling OD time. The applied hypergraph theory outperformed the current state-of-the-art graph-based and non-graph-based methods in capturing long-term temporal dependency. Besides utilizing models to extract different components, Wang et al. manually pre-defined the temporal features from both recent and periodic time steps for temporal modeling, which is another common approach to considering periodicity in prediction [75].

Additionally, another method for depicting long-term dependency is to construct a stack of CNN layers in the time domain. The application of CNN in temporal modeling has shown that it is possible to utilize 1D CNN layers for sequential modeling, suggesting the superiority of typical RNNs [37], [82]. This approach effectively reduces issues inherent to recurrent neural network architecture, such as vanishing gradient problems, making the model easier to converge and train. The convolution operation merges the temporal information at neighboring time slices in the time dimension to extract long-term temporal dependency, as shown in Fig. 9. After applying GCN to aggregate the neighboring information of each node on the graph, Guo et al. stacked convolution operations in the temporal dimension to collect the context of the next time slots [37]. They extracted hourly, daily, and weekly periodicity using the same network structure with Spatio-temporal (ST) blocks, an ST attention layer, and an ST convolution layer. Additionally, due to the advantage of capturing the sequentiality of the traffic flow graph using CNN, Yang et al. applied a gated dilated CNN to capture long-term temporal relations for traffic flow forecasting [24]. The application of dilated convolution allows the gated dilated CNN to capture long-term temporal dependency in non-adjacent time steps. They demonstrated that final traffic flow prediction depends not only on temporal patterns from adjacent time steps but also on non-adjacent time steps, contributing significantly to the modeling.

Zheng et al. adapted the encoder-decoder structure for traffic flow prediction [83]. They embedded the input by considering the temporal context before encoding traffic features. The long-term relationship was modeled by a transform attention layer between the encoder and decoder,

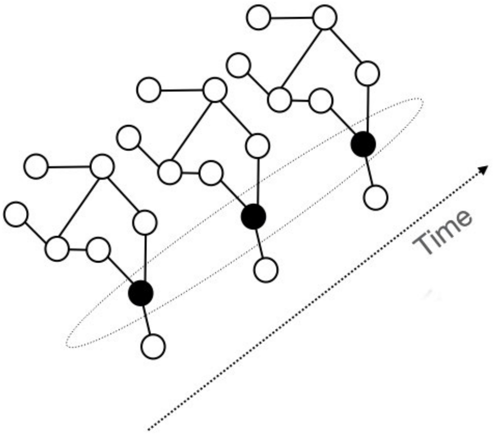

FIGURE 9. Aggregation of the information from the neighboring time slices.

which helped ease the error propagation effect. This approach preserved the long-term temporal relationship by calculating the relevance of future time steps and historical time steps, thus enhancing long-term forecasting.

Currently, Transformer [56] has been considered for long-term dependency extraction because the self-attention mechanism connects data at distant positions [84]. Due to its achievements in fields such as text processing and machine translation for capturing long-term features, the Transformer has also been introduced to the traffic domain [85], [86], [87]. Cai et al. [85] pointed out that Li and Moura [84] ignored the network-wide information in forecasting with the Transformer. They addressed traffic flow prediction by considering both long-term temporal dependency and spatial dependency. They extracted the spatial features using GNN and then encoded them for the continuity and periodicity of the time-series data as the global and local temporal features to the Encoder layer in the Transformer.

Guo et al. obtained global receptive fields using the multi-head self-attention mechanism in the Transformer structure [87]. They expanded the self-attention layer based on the causal convolution operation to fulfill the continuous trend with the local temporal context. The local temporal context referred to the local periodicity combined with the global periodicity as the temporal input. They manually defined the input with the global periodic pattern, which was the time segment from the same day in the past w weeks, while the local periodicity is that in the past d consecutive days. Then, they concatenated the two pieces of periodic information into the spatio-temporal encoder-decoder architecture.

3) TEMPORAL DYNAMICS

In addition to static temporal correlations like short-term and long-term temporal dependencies, traffic flow also exhibits dynamic characteristics in the time domain. Yao et al. revealed that even though traffic flow shows an obvious periodic tendency, it is not as strictly periodic as one might intuit [59]. In other words, the exact rush hours are not always the same every weekday. For example, evening peaks might be busy from 4:00 pm to 6:00 pm from Monday to Thursday but vary on Friday from 2:00 pm to 4:00 pm, demonstrating temporal shifting in the temporal domain. They developed the Periodically Shifted Attention Mechanism based on LSTM combined with the attention mechanism to capture the temporal shifting of periodicity. First, they obtained $h_{i,t}^{p,q}$, which is the representation of the predicted time t between the time interval q in the region i, by the LSTM models applied to collect the prior information within the previous day p. What's more, the attention coefficient $\alpha_{i,t}^{p,q}$ is calculated to measure the importance of the time interval q in day p. And finally, the long-term temporal representation $h_{i,t}^{p}$ that included the shifting periodic information is formed as $h_{i,t}^{p} = \sum_{q \in Q} \alpha_{i,t}^{p,q} h_{i,t}^{p,q}$.

Li et al. used the Dynamic Time Warping (DTW) method to capture shifting temporal information [88]. DTW finds a suitable matching pair with stronger similarity by considering the signature features in the time domain, thus calculating the distance between the similarities of two-time series to obtain the shifting temporal feature on the time axis. Guo et al. employed the traditional self-attention mechanism to match traffic flow with the same numerical value and the CNN layer to capture the temporal context for the local trend [87].

In this part, we have explained how traffic flow data represent the sequential nature, where past states affect future traffic conditions. Early research aimed to uncover temporal correlations for prediction, distinguishing between short-term (5)-15 minutes) and long-term (daily/weekly) dependencies. This led to clearer solutions for extracting fine-grained short and long-term temporal features in the further section. We also discussed the temporal dynamic characteristics in traffic data, which differ from the typical short-term and long-term dependencies.

B. SPATIAL DEPENDENCY

Because of the significant improvement in performance when considering graph-structured information in current work, we should acknowledge that the introduction of graph-based neural networks in traffic flow forecasting models outperforms previous non-graph-based network models [24], [89]. The popular spatial models depicting traffic graph information are GCN. However, the theory of GCN [71], [72] shows that existing GCN and its variant models only consider the static topological structure of the graph, while the traffic road graph in the real world is dynamic and highly non-linear. The expected graph-based spatial models should discuss not only the local impact of nearby nodes but also the multiple spatial dependencies in reality. For example, the spread of influence is not isotropic for each node on the traffic graph. This is because the geographical location of road sections and the functional role of road segments both significantly impact the final traffic flow prediction. Thus, the correlation of traffic states cannot be solely judged by local spatial proximity.

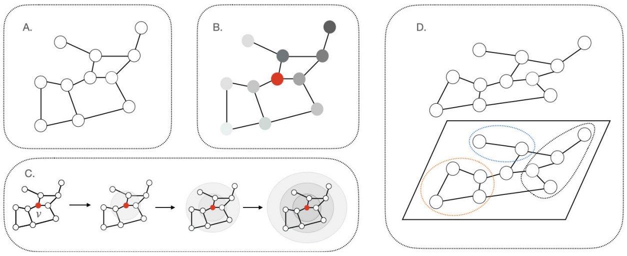

FIGURE 10. The spatial dependency in the traffic flow graph. A. The topology of the graph structure. B. The impact of the neighboring nodes. C. The impact within k-hops. D. The regional similarity.

Initially, when employing graph-based neural networks in traffic flow forecasting, it was essential to select a model that best described the different topologies of the local road network and the spread of influence on traffic conditions. With further study in this area, the discovery of more detailed categories of spatial dependencies should be addressed in future work. It is worth noting that increasing research focuses on discussing the influence of multiple spatial dependencies, such as global and local spatial dependencies, static and dynamic spatial dependencies, and different hops in spatial scale.

1) STATIC SPATIAL INFORMATION

Yang et al. [24] illustrated that the topological structure of the traffic graph should be addressed since graph-based methods outperform grid-structured methods in prediction, as shown in Fig. 10(a). The graph topology reflects the global stationary spatial information on a network-wide scale. An $N \times N$ adjacency matrix represents the graph structure in the graph convolutional network (GCN), reflecting the connectivity among each node on the traffic road graph. It is obtained directly when the traffic graph is constructed and is used throughout the whole process.

When GCN appeared in the traffic flow prediction field, most work considered the static connective topology information for prediction and achieved good performance. In the spatial GCN domain, operations on the graph can be regarded as extracting the spatial dependency from the traffic graph. The convolution for each node can be seen as the update of static local information. On the other hand, spectral GCN captures static global spatial information through transformation. Based on its theory, the Laplacian matrix is transformed by the adjacency matrix, reflecting the connectivity of each node on the graph, which can directly convey the global spatial dependency.

2) DYNAMIC SPATIAL INFORMATION

Models that consider topology information have outperformed previous ones with non-graph dependencies. However, the impacts of the weights on neighboring nodes are not inherently the same, as shown in Fig. 10(b). Affected by adjacent segments or nearby neighbors, the real-time traffic conditions of individual segments change over time. For example, morning traffic congestion will lead to high volumes for nodes on the commute path from residence to workplace, while connected neighbors not on the commute path will not be as significantly affected. Since neighbors do not affect the target node equally, using equal weights for all neighboring nodes loses the dynamic spatial information of the target. Therefore, the solution should not only rely on the pre-designed static adjacency matrix throughout the entire prediction process but should also provide a better expressive representation of the weights of different nodes.

A simple way to model time-varying spatial correlations is by using a predefined adjacency matrix with prior knowledge of the traffic graph. In other words, components referring to dynamic spatial correlations can be represented by static properties. For example, Fang et al. considered that the static adjacency matrix of the graph updated in each time interval can be used as the dynamic adjacency matrix [30]. Feng et al. constructed an adaptive adjacency matrix predicted by the predefined adjacency matrix based on the dynamic graph learning (DGL) concept [90].

Methods considered for learning spatial dynamics focus on calculating different weights of the real-time spatial information obtained directly from the data. Different kinds of aggregators are adapted to model these locality dynamics. Zhang et al. pointed out that some GCN and its variants tried to assign a non-parametric weight, such as graph pooling aggregator and graph pairwise sum aggregator, to the connected nodes, thus producing dynamic spatial correlations [78]. However, non-parametric aggregators cannot differentiate the edge weights between the target and all its neighbors. The attention mechanism is a popular and more effective aggregator introduced to specify the weights of the nodes in the neighborhood without understanding the topology of the traffic graph.

Some work, such as [37], [87], and [91], defined the spatial attention mechanism to adaptively adjust the attention scores with the neighbors to capture the dynamics on the graph. However, there are differences among the applications. Guo et al. [37] utilized the attention mechanism to extract the spatial dynamics before the spatial convolutional operations on the graph, while Bai et al. [91] and Guo et al. [87] employed it to obtain the attention coefficients based on the extracted spatial information on the traffic graph. The attention mechanism is incorporated in GNNs as graph attention networks (GAT) [76], where the weights can be directly represented by attention coefficients under the graph structure. GAT uses masked self-attention layers for assigning weights in the aggregation within the same hop. The application of the standard attention mechanism can be regarded as a single attention head under a multi-head attention mechanism, as it represents the data under one subspace. The multi-head attention mechanism extends this further by running a single attention head several times in parallel, enhancing the ability to explore representation under multiple subspaces. For example, image features can be regarded as under multiple subspaces since they represent different aspects like color, line, and texture spaces. Lu et al. aimed to get richer latent information from different representation subspaces [88]. Additionally, Yang et al. [24] and Zheng et al. [83] extended the self-attention mechanism to the multi-head one to stabilize the learning process.

3) GEOGRAPHICAL SCALE

Most of the graph-structured models simply consider generating the spatial representation given the information from one-hop neighbours, such as GCN and GAT. However, the impact of a larger scale on the centre node could reflect the richer informative spatial dependency because the nodes at k-hops can describe the different influences towards the target segment [24], as seen from Fig. 10(c). Given a city-wide traffic network, when traffic congestion happened in a specific road segment, the linked neighbouring roads could be actively affected by this great event in varying degrees. Because the radiation zone could be affected in multi-hops, the traffic jam trouble can last for several hours in a specific road segment when happened. A larger k hop of the node can be involved to help capture the broader spatial dependency compared with the one-hop model.

To distinguish the contributions of neighbours in different hops, Wang et al. constructed a stack of k graph convolutional layers to get the information from different neighbouring scales and computed the attention coefficients to each layers [75]. Given the multiple scales of the spatial and temporal features are discovered to affect the final results, Yang et al. adapted MST and MSS sub-blocks to capture the spatial and temporal correlations separately [24]. They fused the correlations from the different spatial scales and temporal scales in generating the outputs of the spatial-temporal relationships. The various impacts of the ST correlations in multiple scales are based on a weighted sum fusing method, which expresses the ability of the multi-scale ST correlations.

4) SPATIAL HOMOGENEITY

The time computation on the traffic graph is usually consuming since the traffic graph is huge in modelling. The inflow/outflow interactions between adjacent regions can lead to local similarity(Fig. 10(d)), hence strong spatial relations might exist in the nearby regions. Zhang et al. designed the transportation neighbourhood adjacency matrix based on the spatial proximity between nodes in the road graph to show the improvement of the model considering the local similarity [35]. However, some work also pointed out that the nodes on the traffic road network could own the similarity no matter the distance between each other. Wang et al. [92] discovered that even the road segment had a great similar impact to the one even though it was distant comparatively, because of the similar road environment. The more detailed solutions discussing the region similarity will be shown in the next session since region-level similarity is a heated topic in recent years.

In this part, we have stated that graph-based neural networks have significantly improved traffic flow forecasting compared to non-graph models. On the one hand, popular models like GNNs capture static graph structures. But on the other hand, real traffic graphs are dynamic and nonlinear due to geography and road function, which requires effective graph-based models to consider local and diverse spatial dependencies. Significantly, selecting appropriate spatial models for local road networks depends on a deep understanding of diverse statical and dynamic impacts of traffic networks was vital. Current research delves into spatial dependencies like global/local dependency, static/dynamic dependency, and different geographical scales and spatial homogeneity to better model complex traffic patterns.

C. THE RELATIONSHIP OF SPATIAL DEPENDENCY AND TEMPORAL DEPENDENCY

Even though we can see so many surveys in this area have realized the importance of utilizing spatial and temporal correlations to traffic flow forecasting, previous research still needs to include a view of how to describe the spatial and temporal dependency relationship to design the spatio-temporal architecture.

In fact, as the similar idea of ensemble methods in machine learning, the architecture of spatio-temporal modelling has shared familiar points with it, which builds the submodels in a parallel and sequential way. The parallel and sequential modelling of spatial and temporal dependency shows the inspiration of the solution to traffic flow forecasting.

After figuring out the spatial and temporal dependency, we further categorize the architecture of capturing spatio-temporal dependency when considered in modelling, which can be divided into parallel and sequential modelling.

1) SEQUENTIAL MODELLING IN SPATIO-TEMPORAL ARCHITECTURE

The basic submodels are integrated sequentially in ensemble modelling, and the same is in spatio-temporal modelling. The principle of sequential modelling is exploiting the submodels' dependencies by assigning hidden relationships that we can not discover by eyes; the overall prediction performance can be improved when compared by sole modelling. First, in sequential modelling, applying the spatial-related models to get the spatial information from the raw data and then getting the processed features into the temporal models to obtain the temporal extraction for the final results, which can be seen in Fig. 11 A. Zhao et al. proposed the typical sequential architecture with the separate spatial and temporal extraction applied to spatio-temporal modelling in the early research [74] – the spatial dependence referred to the topology of the traffic road network that is captured by GCN; and the temporal dependence was represented by temporal dynamics, with the extraction of the GRU. Zhu et al. were inspired and developed the sequential modelling for each moment based on Zhao et al. by introducing an attention mechanism to capture the importance of information at each time slot. For this idea of construing the spatio-temporal modelling, we can see so many similar consideration in the following research [39], [93], [94], [95], [96].

Yu et al. and Diao et al. [39], [95] showed the basic structure of the ST (spatio-temporal) blocks that involved the stack of multiple spatial and temporal submodels in capturing the dependencies in sequence in the blocks. A stack of the ST blocks is also under sequential modeling to capture the spatio-temporal dependencies for more extended time information.

2) PARALLEL MODELLING IN SPATIO-TEMPORAL ARCHITECTURE

In paralleling modelling, the base submodels are constructed in parallel structure, the principle of which is to exploit the independence between the submodels and make up the ignored information of the other learners. Depending on the dependencies ' characteristics, the base submodels can be the same methods. Guo et al. proposed the typical parallel architecture for processing spatio-temporal dependency in sub-categories [37]. The input data is divided into sub-categories (hourly, daily, weekly) first, fed into three branches with the same blocks extracted the spatiotemporal pattern from hourly, daily, and weekly levels. The following work, such as [31], [97], [98], [99], [100], and [101] considered parallel structure under the similar idea and have different branches for extracting the sub-category dependency. With Han et al. [97] and Hong et al. [98] further developed the spatio-temporal feature extraction of daily, weekly, and recent information on sub-models in parallel modelling, The other work employed more branches on extracting multiple fine-grained dependencies in parallel structure. Sun et al. proposed the models with seven branches on extracting spatial and temporal features in sub-categories separately and then fused for representing spatio-temporal dependency [99], which is as shown in Fig 11 D. Luo et al. mined the heterogeneous information in three branches that were constructed by different sub-models [100]. Within the branches, the parallel modelling can be achieved with sub-models which are constructed by parallel modelling as well. Zhang et al. came up the models with spatial branch and temporal branch, which are with sub-branches on extracting fine-grained spatial and temporal dependency separately [31].

3) SPATIAL-TEMPORAL DEPENDENCY

As that researchers prefer to model the spatial and temporal dependency separately, it is also necessary to consider how to fuse the components to can make best utilize of the hidden spatio-temporal relationship from the combination. With the discovery of the effective and informative spatial-temporal expression in traffic flow prediction, multiple fusion methods have been employed in feature fusion for the different features, thus further contributing to the improvement of the model performance. Even the external factors, such as weather conditions, car accidents, and events hosted near the target, seems as no directly relationship for the prediction, some work still shows that the availability of the external data enhance the model performance [34], [102]. Hence, the methods on fusing the features in the current traffic flow forecasting field is worth discussing for integrating the spatial and temporal features extracted from the raw data. In fact, it is through the information superposition of the input to enrich the obtained features, thereby improving the performance of the model.

a: CONCATENATION

Feature Concatenation is a common method of feature fusion. It is to contact the different feature vectors directly in the same order. Assume that we have $v_1 \in R^m$ and $v_2 \in R^n$, the fused vector could be $v = [v_1, v_2] \in R^{m+n}$. The concatenation of the features has been applied in many types of research from different perspectives. Reference [83] concatenated the information of the day-of-week $v_1 \in R^7$ and time-of-day $v_2 \in R^T$ of each time step into the temporal vector $v \in R^{7+T}$. Yao et al. [59] reconstructed the temporal representation that reserved both short-term and long-term dependencies for traffic flow predicting by directly concatenating the short-term representation $h_{i,t}$ and long-term representation $h^P_{i,t}$ as $h^c_{i,t}$. Yao et al. took the temporal representation by the concatenation of the spatial dependency and the external factors before feeding it to the LSTM. The extracted information is further concatenated with the feature from the semantic view for the final demand prediction [93].

Wu et al. and Li et al. introduced the graph attention network (GAT) with a multi-head mechanism to capture the dynamic expression of the node feature [29], [62]. The features under the K independent attention mechanisms are concatenated after the extraction. Cui et al. enriched the spatial information by concatenating the features extracted from the k hops of neighborhood on the traffic road graph [103]. Guo et al. [104] concatenated the region feature j with the road feature i if the segment i belongs to the region j.

b: ELEMENT-WISE OPERATION

As we can see that $v = [v_1, v_2] \in R^{m+n}$, the dimension (number of channels) of the features v has increased after the operation of concatenation, but the processed feature keeps its original information as v_1, v_2. The element-wise operation is to richer the information but keeps the same dimension of the feature after processing. For example, as the element-wise product, assume that we have $v_1 \in R^m$ and $v_2 \in R^m$, the fused vector could be $v = v_1 \circ v_2 \in R^m$.

When the different components are fused, the different weights are learned from the data, and the final results can be regarded as obtained by the linear weighting method. Wang et al. applied the element-wise operation to fuse the closeness TD_{t_c}, period TD_{t_p} and trend TD_{t_t} representation from three predicted branches to the final result [80].

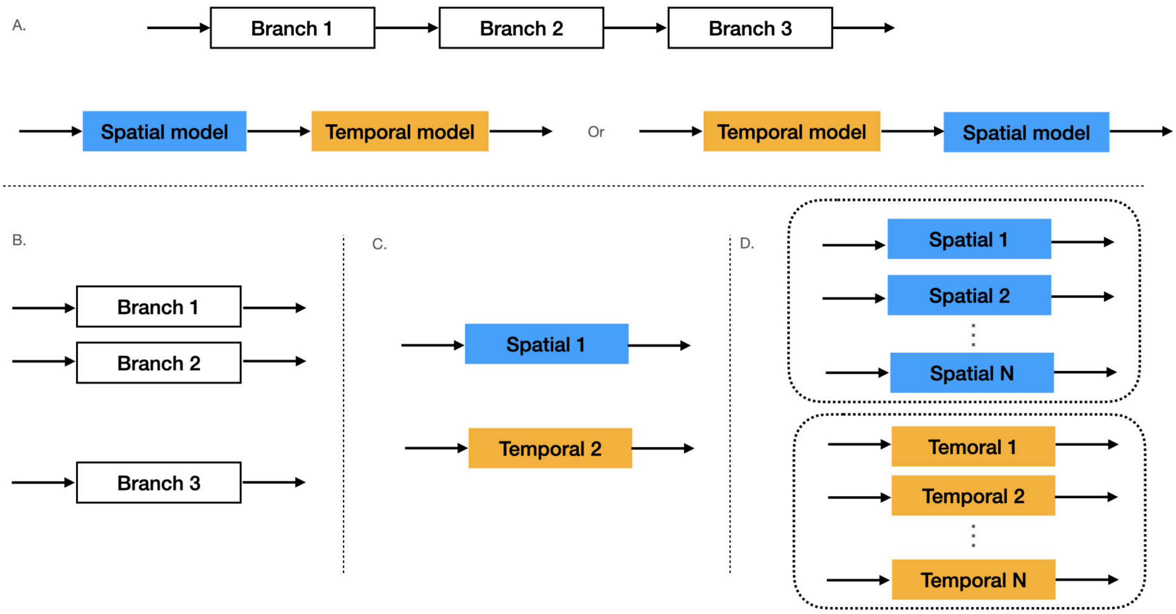

FIGURE 11. The differences of sequential and parallel modelling in machine learning and spatio-temporal modelling. A. Sequential modelling of ensemble learning and spatio-temporal modelling B. Parallel modelling of ensemble learning C. Parallel modelling of spatio-temporal modelling D. Parallel modelling of spatio-temporal modelling in subcategory.

$TD_t = W_c \odot TD_{t_c} + W_p \odot TD_{t_p} + W_t \odot TD_{t_t}$, where TD_t is the final result, and the W_c, W_p, W_t are the weight matrix of the three temporal predictors respectively. Reference [37] considered the hourly, daily and weekly components in the same way. Other examples (for example, [61], [62], [102], [103], [105]) with the application of the element-wise addition and product will not be listed in details.

Zheng et al. [83] employed a gated fusion mechanism which can adaptively control the spatial and temporal information $H^l = z \odot H_S^l + (1 - z) \odot H_T^l$, where H_S^l and H_T^l represents the spatial and temporal dependency obtained from the last layer, and z is the gate. Reference [78] mentioned the fusion methods, such as pairwise sum and pooling approaches, applied to the node feature. It studied the gated attention-based fusion method with the consideration of a multi-head mechanism and outperformed the pairwise sum and pooling on the graph nodes with fewer number parameters.

c: LIKELIHOOD

Since the random walk-based network embedding approaches constructed a co-occurrence matrix to help describe the spatial dependency and temporal dependency [106] and received a relatively satisfactory result, Liu et al. [107] proposed to compare the co-occurrence matrix generated by the random walk approach, to maximize the likelihood of the time-evolving traffic graphs. It jointly captured spatial-temporal dependency instead of extracting the spatial and temporal correlations from two independent sources.

In this part, we mentioned the model structures that can be considered for spatio-temporal feature extraction: sequential modeling, parallel modeling, and fusion modeling (spatial-temporal). When discussing the feature extraction structure of sequential modeling and parallel modeling, we can find the similarity with ensemble learning in machine learning. We also give work under these two spatio-temporal feature extraction structures. In addition to serial modeling and parallel modeling, we also mentioned the way of direct fusion, and mentioned three classic methods of spatio-temporal feature fusion.

IV. DISCUSSION ON THE CURRENT SOLUTIONS

When categorising the solution to traffic flow prediction til now, they can be mainly divided into two aspects: one is the introduction of new components, which can be referred to as the introduction of the graph-based neural network applied to model the non-euclidean spatial dependency; and the other is the presentation of the new scenarios, for example, the general temporal dependency has been detailed categorized as hourly temporal patterns, daily temporal patterns, and weekly temporal patterns.

A. FEATURE ENGINEERING ON TEMPORAL PERSPECTIVE

The topic for further research in recent years has attached to the discovery of informative feature engineering for the input to the traffic flow forecasting models. For example, the researchers tend to focus on digging for much more appropriate approaches to build the general dependency under the detailed categories, such as temporal position embedding in the transformer which enhances the ability of temporal order expression instead of the time-series models to describe the general temporal dependency; and, Dynamic Time Warping

(DTW) algorithm applied as clustering method to describe the locality of the spatial dependency, etc, instead of the spatial models to represent general spatial features. This direction requires researchers to have a comprehensive understanding of the factors that could affect traffic flow forecasting. It leads to feature reconstruction considering the multiple facts from different perspectives of the raw data. In other words, previous work extracts the features from the raw data with the spatial and temporal models, while current work aims to reconstruct the feature vectors before feeding the raw data. Feature engineering can better reflect and enhance the spatial and temporal correlations in real-world traffic conditions.

The combination of the multiple temporal correlations is necessary to be utilized in the long-term prediction [24]. The temporal features, on one hand, are represented as the short-term dependency that is affected within the most recent historical time step; on the other hand, are under the influence of the strong periodic characteristics in daily and weekly cycles. Most of the recent papers employed the idea on trading of the utilization of the short-term and long-term in the temporal feature pre-construction, the variables of which are always in hourly, daily, and weekly periodicity to reflect the variation in the time domain [31], [36], [75], [81], [85], [108], [109], [110]. The multi-dimension time dependency shows the periodic variation of the hour, day and week, and can be fused as the overall temporal input to the model based on the task required.

B. FEATURE ENGINEERING ON SPATIAL PERSPECTIVE

Besides the multiple temporal dependencies, the spatial features can also be enriched by the multi-dimensions representation [31], [35], [36], [109], [111]. Zhang et al. focused on obtaining the richer spatial information by introducing the content similarity adjacency matrix, the transportation neighborhood adjacency matrix, and the graph betweenness adjacency matrix to represent spatial correlations in traffic flow graph as the geography correlation, region similarity, and the road connectivity [35]. Li et al. proposed a multi-scale graph convolutional layer to assign different weights to three kinds of spatial dependencies [86]. The three spatial features referred to the normalized adjacency matrix with the self-connected unit that is represented by the geographic adjacency matrix, the extraction of hidden spatial dependencies that are represented by the self-adaptive matrix, and the similarity of the traffic patterns that is represented by the similarity matrix. Wang et al. pointed out that the existing GCN methods adapted the spatial feature of the latest structure (e.g., k-hops), ignoring the information of the previously obtained stages (e.g., before $k-1$-hops) [75]. They utilized the completed spatial information obtained from the previous neighboring range and proved the efficiency of the spatial enhancement.

Guo et al. considered the feature vectors embedded in spatial and temporal respective separately [87]. In the temporal dimension, they manually defined the global and local periodicity and concatenated them with the time steps X of the historical traffic records, which finally led to the new temporal input. In the spatial dimension, the GCN with Laplacian smoothing form is employed subsequently to obtain the spatial embedding vectors. This paper also discussed spatial heterogeneity, which referred to the observation that the traffic records generated in different locations obtained different traffic patterns, and further discussion on regional solutions proposed to model it. Fang et al. proposed a convolution model named Dilated Attention Graph Convolution(DAGC) that can generate both the spatial dependency and temporal dependency [112]. The non-local spatial correlations in multi hops are captured by DAGC as the spatial input and the adjusting parameters of which also applied to fuse the temporal correlations as the temporal input.

C. FEATURE ENGINEERING ON SPATIO-TEMPORAL PERSPECTIVE

Besides embedding the spatial and temporal features separately, Bai et al. adapted the MLP to learn the new spatio-temporal features from the raw input, which can directly be used as the predictors in the proposed model [113]. It has been proved that direct extraction of the new features with MLP works well in image recognition and voice translation. And it is more effective than the manual design of the new feature from the raw data. Jiang et al. came up with a similar architecture to capture the dependencies as previous research. But they introduced a spatio-temporal relation matrix to represent the traffic road topology as the spatial input [33]. It is because the spatio-temporal relation matrix refers to the space-time constraint on the traffic road conditions. As the spatial proximity matrix is usually applied to represent the relationship between the target road segment and its corresponding neighbours, it is utilized to depict the topology of the road network in this work. And they supposed the historical time interval to represent the traffic speed time series in the temporal dimension, which is the common approach as before. However, because they adapted a cross-correlation function to fuse the time information in each time interval, the after-fused spatio-temporal relation matrix is defined as the one that combined the spatio-temporal information of the road traffic in a matrix before feeding into the model.

D. FEATURE ENGINEERING ON EXTRA FACTORS

Except from reconsidering the representation of the spatial and temporal feature vectors for traffic flow data, Zhu et al. introduced the external factors to reconstruct the input to the model [34]. They employed the other factors which could affect the traffic conditions as the auxiliary attributes, such as static geographic information, which is a static factor; and weather condition, which is the dynamic factor because of its time-varying determinant. The static factors, dynamic factors and flow values are combined as the traffic characteristic information at time t in the proposed model.

E. REGION-LEVEL SOLUTION

Another direction of the work for traffic flow forecasting is the proposition of region-level solutions for spatial feature extraction in heterogeneous regions have been proposed. As the previous work paid much attention to capturing the spatial dependencies on graph level that depicts the information of all the nodes on the graph, the researchers try to mine the spatial relationship among the irregularity of the region [35], [36], [75], [114], [115], [116]. For example, every day, the commuters leave the residential area and driving to the workplace could lead to the morning peak and the opposite to the evening peak. So this displays an instinctive transition flow that existed because of the different regional heterogeneity at different periods in a day. Hence, it can be concluded that all the nodes on the traffic road network are not isolated while owning functional similarity in a certain region. It also leads to the introduction of the community algorithm [36], [115] in capturing the regional spatial dependency to rich the extracted spatial features for prediction. The community structure implies the nodes with high similarities could belong to the same community, referring to the same functionality in the traffic road network.

In [83], a group spatial attention mechanism was proposed based on the sub-graph partition of the intra-group spatial attention part and inter-group spatial attention part. But the graph-wide computation takes time and memory consumption. The Louvain algorithm, proposed by Blondel et al. [117], was adapted in [36] for the discovery of the different functional communities in the road network and contributed to the construction of the functional similarity graph(FSG) for the representation ability of the spatial dependency. To avoid the bias of the static adjacency matrix, which only represents the connectivity of each node, Zhang et al. employed the fuzzy-graph generation network to enhance the expression of the intrinsic correlation for the regional spatial feature [116]. The method utilizes the fuzzy logic relationship among nodes on the traffic road network to generate the cluster in similarity.

V. FUTURE DIRECTIONS

In this section, we discuss future directions based on recent advancements in traffic flow forecasting using relevant spatio-temporal models. These advancements include the introduction of graph-based models to capture the topological structure of traffic road networks, as opposed to extracting grid-based spatial information as done in previous work. Additionally, new scenarios and more detailed subcategories of spatial and temporal dependencies in modeling have been proposed, such as the periodicity in hourly, daily, and weekly under the category of long-term temporal dependency. Furthermore, expressive feature engineering has been emphasized for solving traffic flow prediction problems. Besides the work mentioned earlier, we believe other techniques have potential in addressing problems in this area.

A. THE RECONSTRUCTION OF THE GRAPH STRUCTURE

With the success of region-level solutions, we recognize the idea behind it: the high similarities among nodes contribute to the formation of node groups with similar properties. Based on this inspiration, future solutions should consider appropriate graph partitioning to reduce redundant graph information for the targets, thus improving the efficiency and performance of the model.

In realistic scenarios, the graph-structured traffic road network is large-scale, bringing unnecessary information for prediction, which is directly reflected by time-consuming problems that are often discussed. Two reasons explain this in a large graph: First, compared to all the nodes on the traffic graph, the links to the target nodes are limited when the adjacency matrix is constructed graph-wide. Hence, the significant traffic graph shows a highly sparse characteristic when all the nodes are connected only with their neighbors. Additionally, as the traffic network generally expands in city size, it brings enormous topological edge information to the graph matrix. Second, another relevant factor is the increasing number of features proposed to model the scenario in recent work. The feature aggregation for each node could lead to extensive efforts because more features need to be considered in the computation.

Researchers have attempted to introduce graph reduction to tackle this issue in large traffic graphs in recent years. For example, Guo et al. [104] introduced spectral clustering (pooling) of the traffic graph on the Laplacian matrix of the adjacency matrix to reconstruct the graph structure. However, it was shown that the reduced graph contributes to model performance in short-term prediction tasks, while long-term prediction did not see the same improvement. Wu et al. [22] pointed out that partial information of the graph might be lost if graph reduction is introduced. Therefore, whether and when to investigate the performance on the reduced graph could be a topic for future research.

B. DEPENDENCY TRADE-OFF

Even though the models for traffic flow forecasting emphasize the importance of the spatial correlations and temporal correlations, it is still an open question on how to balance the contribution of the temporal correlations and spatial correlations in the final prediction [107]. Partially because the existing methods applied to the traffic flow prediction have received relatively satisfying results without considering the spatial or temporal contribution weights; in other words, does it mean that the importance of temporal dependency should outweigh that of spatial dependency, or in reverse? The typical architecture of the traffic flow prediction is to separately consider the spatial and temporal features, whether sequential or parallel. However, balancing the importance of the extracted temporal and spatial information and utilizing it to the fullest must be considered.

C. MULTI-SOURCE SPATIAL AND TEMPORAL CORRELATIONS

Most previous works focused on extracting the spatio-temporal dependency from a single source. For example, researchers investigated the hidden relationships between spatial and temporal perspectives using traffic flow data. However, Fang et al. [30] discovered that different data sources from the same or similar traffic areas exhibited similar distributions in both temporal and spatial dimensions, guiding future research in exploring hidden traffic relationships from the target area. They verified this with traffic flow graphs of taxi and bike data in NYC, capturing the spatio-temporal dependency in a specific NYC area. This implies that trajectory data from different sources could be combined to extract hidden complex spatio-temporal dependencies for traffic flow prediction.

Yao et al. showed another approach to using different data sources [93]. They provided the graph's spatial and semantic views and combined the extracted features for forecasting. Therefore, future research should emphasize exploring the shared relationships of multi-source data from the same target area since it could potentially improve forecasting accuracy.

D. REPRESENTATION LEARNING

Inspired by Transformer, we can see the model's ability to express context information through position embedding. Guo et al. [87] adapted spatial position embedding and temporal position embedding methods to induce order information in spatial and temporal dependencies, resulting in more accurate predictions. Hence, applying several appropriate learning representations on the raw data, which better represent different dependencies of the datasets, can potentially contribute to model performance. In addition to the models for spatial and temporal dependency representation mentioned in this paper, Zhang et al. [31] revealed that graph embedding technologies such as DeepWalk [118] and Node2Vec [119] can extract and mine hidden spatial patterns by capturing the details of spatial dependency in graph-structured data.

Zheng et al. [83] considered spatial and temporal embeddings to provide static and dynamic representations among traffic sensors, fusing them as spatio-temporal embeddings to obtain time-variant vertex representations for combined static-dynamic spatio-temporal feature expression. Wang et al. [120] proposed an adaptive GCN with multi-channel to inspire future work in extracting informative spatial features on traffic graphs. They extracted spatio-temporal information by adopting two convolution operators from the given feature and topology information. Therefore, informative representation learning is still required to mine hidden spatio-temporal patterns and provide a comprehensive understanding of the overall behavior of the data.

VI. CONCLUSION

In this paper, we believe that a rigorous understanding of the task can contribute to the appropriate design of an efficient deep-learning-based solution to the traffic problem. Therefore, we analyzed the traffic flow forecasting problem from spatial and temporal perspectives at a fine-grained level, which could inspire future researchers to design models for specific scenarios. First, we presented the problem definition of traffic flow forecasting from a sole temporal perspective to a spatio-temporal perspective, along with corresponding techniques for spatial and temporal feature extraction, including architectures such as Seq2Seq and Transformer. Then, we categorized and summarized a new taxonomy of the fine-grained dependencies in spatial and temporal features of traffic flow forecasting problems and provided methods for addressing these dependencies.

In terms of spatial and temporal feature extraction for constructing spatio-temporal dependencies, we provided guidance on how to extract spatio-temporal features when building the ST architecture–sequential modeling, parallel modeling, and fusion methods. Additionally, we discussed how current work focuses on addressing spatio-temporal feature extraction, considering multi-scale temporal information (e.g., weekly, daily, and recent temporal data) and spatial information (e.g., k-hop neighboring information and solutions focusing on fusing region-level graph information).

We pointed out future directions and discussed possibilities in traffic flow forecasting, such as the reconstruction of given graph information, balancing the employment of temporal and spatial information, utilizing multi-source data to capture hidden shared spatio-temporal information in the target area, and informative representation learning for traffic context information.

REFERENCES

[1] A. Bull, *Traffic Congestion: The Problem and How to Deal With it*. Santiago, Chile: ECLAC, 2003.

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[3] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1840–1852, Mar. 2021.

[4] P. Sun, N. Aljeri, and A. Boukerche, "Machine learning-based models for real-time traffic flow prediction in vehicular networks," *IEEE Netw.*, vol. 34, no. 3, pp. 178–185, May 2020.

[5] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.

[6] J. F. Gilmore and N. Abe, "Neural network models for traffic control and congestion prediction," *J. Intell. Transp. Syst.*, vol. 2, no. 3, pp. 231–252, 1995.

[7] Z. Xie, W. Lv, S. Huang, Z. Lu, B. Du, and R. Huang, "Sequential graph neural network for urban road traffic speed prediction," *IEEE Access*, vol. 8, pp. 63349–63358, 2020.

[8] S. Jiang, W. Chen, Z. Li, and H. Yu, "Short-term demand prediction method for online car-hailing services based on a least squares support vector machine," *IEEE Access*, vol. 7, pp. 11882–11891, 2019.

[9] Y. Li, D. Deng, U. Demiryurek, C. Shahabi, and S. Ravada, "Towards fast and accurate solutions to vehicle routing in a large-scale and dynamic environment," in *Proc. Int. Symp. Spatial Temporal Databases*. Cham, Switzerland: Springer, 2015, pp. 119–136.

[10] Z. Peng, W. Jian-Wei, S. Mao-Peng, and Z. Ya-Xin, "Vehicle scheduling for mountainous expressway traffic emergency," *China J. Highway Transp.*, vol. 31, no. 9, pp. 175–181, 2018.

[11] Z. Bai, W. Shangguan, B. Cai, and L. Chai, "Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic," in *Proc. Chin. Control Conf. (CCC)*, Jul. 2019, pp. 8600–8605.

[12] M. Asghari, D. Deng, C. Shahabi, U. Demiryurek, and Y. Li, "Price-aware real-time ride-sharing at scale: An auction-based approach," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Oct. 2016, pp. 1–10.

[13] Y. Li and C. Shahabi, "A brief overview of machine learning methods for short-term traffic forecasting and future directions," *SIGSPATIAL Special*, vol. 10, no. 1, pp. 3–9, Jun. 2018.

[14] Z. Wang, P. Sun, Y. Hu, and A. Boukerche, "A novel mixed method of machine learning based models in vehicular traffic flow prediction," in *Proc. Int. Conf. Model. Anal. Simul. Wireless Mobile Syst. Int. Conf. Modeling Anal. Simulation Wireless Mobile Syst.*, Oct. 2022, pp. 95–101.

[15] R. K. Oswald, W. T. Scherer, and B. L. Smith, "Traffic flow forecasting using approximate nearest neighbor nonparametric regression," Final Project ITS Center Project: Traffic Forecasting: Non-Parametric Regressions, 2000.

[16] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022.

[17] L. N. N. Do, N. Taherifar, and H. L. Vu, "Survey of neural network-based models for short-term traffic state prediction," *WIREs Data Mining Knowl. Discovery*, vol. 9, no. 1, p. e1285, Jan. 2019.

[18] K. Lee, M. Eo, E. Jung, Y. Yoon, and W. Rhee, "Short-term traffic prediction with deep neural networks: A survey," *IEEE Access*, vol. 9, pp. 54739–54756, 2021.

[19] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Veh. Commun.*, vol. 20, Dec. 2019, Art. no. 100184.

[20] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2022.

[21] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117921.

[22] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[23] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 3656–3663.

[24] C. Yang, Z. Zhou, H. Wen, and L. Zhou, "MSTNN: A graph learning based method for the origin-destination traffic prediction," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[25] Q. Luo and Y. Zhou, "Spatial–temporal structures of deep learning models for traffic flow forecasting: A survey," in *Proc. 4th Int. Conf. Intell. Auto. Syst. (ICoIAS)*, May 2021, pp. 187–193.

[26] H. Mu, "Deep learning based feature engineering for discovering spatio-temporal dependency in traffic flow forecasting," Ph.D. dissertation, College Eng., EECS, Univ. Ottawa, Ottawa, ON, Canada, 2023.

[27] H.-G. Zimmermann, C. Tietz, and R. Grothmann, "Forecasting with recurrent neural networks: 12 tricks," in *Neural Networks: Tricks of the Trade*. Cham, Switzerland: Springer, 2012, pp. 687–707.

[28] X. Dai, R. Fu, Y. Lin, L. Li, and F.-Y. Wang, "DeepTrend: A deep hierarchical neural network for traffic flow prediction," 2017, *arXiv:1707.03213*.

[29] T. Wu, F. Chen, and Y. Wan, "Graph attention LSTM network: A new model for traffic flow forecasting," in *Proc. 5th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2018, pp. 241–245.

[30] Z. Fang, L. Pan, L. Chen, Y. Du, and Y. Gao, "MDTP: A multi-source deep traffic prediction framework over spatio-temporal trajectory data," *Proc. VLDB Endowment*, vol. 14, no. 8, pp. 1289–1297, Apr. 2021.

[31] S. Zhang, Y. Guo, P. Zhao, C. Zheng, and X. Chen, "A graph-based temporal attention framework for multi-sensor traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7743–7758, Jul. 2022.

[32] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies," *Transp. Res. C, Emerg. Technol.*, vol. 105, pp. 297–322, Aug. 2019.

[33] M. Jiang, W. Chen, and X. Li, "S-GCN-GRU-NN: A novel hybrid model by combining a spatiotemporal graph convolutional network and a gated recurrent units neural network for short-term traffic speed forecasting," *J. Data, Inf. Manage.*, vol. 3, no. 1, pp. 1–20, Mar. 2021.

[34] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, "AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," *IEEE Access*, vol. 9, pp. 35973–35983, 2021.

[35] Z. Zhang, Y. Li, H. Song, and H. Dong, "Multiple dynamic graph based traffic speed prediction method," *Neurocomputing*, vol. 461, pp. 109–117, Oct. 2021.

[36] J. Tang, J. Liang, F. Liu, J. Hao, and Y. Wang, "Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network," *Transp. Res. C, Emerg. Technol.*, vol. 124, Mar. 2021, Art. no. 102951.

[37] 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.

[38] C. Tang, J. Sun, Y. Sun, M. Peng, and N. Gan, "A general traffic flow prediction approach based on spatial–temporal graph attention," *IEEE Access*, vol. 8, pp. 153731–153741, 2020.

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

[40] W. Zhao, Y. Gao, T. Ji, X. Wan, F. Ye, and G. Bai, "Deep temporal convolutional networks for short-term traffic flow forecasting," *IEEE Access*, vol. 7, pp. 114496–114507, 2019.

[41] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1003–1012.

[42] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[43] C. Tian and W. K. Chan, "Spatial–temporal attention wavenet: A deep learning framework for traffic prediction considering spatial–temporal dependencies," *IET Intell. Transp. Syst.*, vol. 15, no. 4, pp. 549–561, Apr. 2021.

[44] C. Gang, W. Shouhui, and X. Xiaobo, "Review of spatio-temporal models for short-term traffic forecasting," in *Proc. IEEE Int. Conf. Intell. Transp. Eng. (ICITE)*, Aug. 2016, pp. 8–12.

[45] Q. Shi and M. Abdel-Aty, "Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 380–394, Sep. 2015.

[46] L. Yang, Q. Yang, Y. Li, and Y. Feng, "K-nearest neighbor model based short-term traffic flow prediction method," in *Proc. 18th Int. Symp. Distrib. Comput. Appl. Business Eng. Sci. (DCABES)*, Nov. 2019, pp. 27–30.

[47] W.-C. Hong, Y. Dong, F. Zheng, and S. Y. Wei, "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model," *Appl. Math. Comput.*, vol. 217, no. 15, pp. 6733–6747, Apr. 2011.

[48] J. Ouyang, F. Lu, and X. Liu, "Short-term urban traffic forecasting based on multi-kernel SVM model," *Image Graph*, vol. 15, pp. 1688–1695, Jan. 2010.

[49] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2001–2013, Jun. 2019.

[50] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 1997, pp. 999–1004.

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

[52] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

[53] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, ''Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning,'' *Transportmetrica A, Transp. Sci.*, vol. 15, no. 2, pp. 1688–1711, Nov. 2019.

[54] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

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

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[57] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, ''Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5243–5253.

[58] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, ''Traffic flow prediction via spatial temporal graph neural network,'' in *Proc. Web Conf.*, Apr. 2020, pp. 1082–1092.

[59] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, ''Revisiting spatial–temporal similarity: A deep learning framework for traffic prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5668–5675.

[60] X. Wang, C. Chen, Y. Min, J. He, B. Yang, and Y. Zhang, ''Efficient metropolitan traffic prediction based on graph recurrent neural network,'' 2018, *arXiv:1811.00740*.

[61] C. Chen, K. Li, S. G. Teo, X. Zou, and Z. Zeng, ''Gated residual recurrent graph neural networks for traffic prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 485–492.

[62] D. Li and J. Lasenby, ''Spatiotemporal attention-based graph convolution network for segment-level traffic prediction,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8337–8345, Jul. 2022.

[63] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, and Y. Zhang, ''Deep and embedded learning approach for traffic flow prediction in urban informatics,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3927–3939, Oct. 2019.

[64] S. Du, T. Li, X. Gong, Y. Yang, and S. J. Horng, ''Traffic flow forecasting based on hybrid deep learning framework,'' in *Proc. 12th Int. Conf. Intell. Syst. Knowl. Eng. (ISKE)*, Nov. 2017, pp. 1–6.

[65] B. M. Williams, ''Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling,'' *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1776, no. 1, pp. 194–200, Jan. 2001.

[66] P. W. Battaglia et al., ''Relational inductive biases, deep learning, and graph networks,'' 2018, *arXiv:1806.01261*.

[67] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, ''Graph neural networks: A review of methods and applications,'' *AI Open*, vol. 1, pp. 57–81, Jan. 2020.

[68] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, ''Gated graph sequence neural networks,'' 2015, *arXiv:1511.05493*.

[69] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, ''Message passing neural networks,'' in *Machine Learning Meets Quantum Physics*. Cham, Switzerland: Springer, 2020, pp. 199–214.

[70] W. L. Hamilton, R. Ying, and J. Leskovec, ''Inductive representation learning on large graphs,'' in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[71] M. Defferrard, X. Bresson, and P. Vandergheynst, ''Convolutional neural networks on graphs with fast localized spectral filtering,'' in *Proc. NIPS*, 2016, pp. 3844–3852.

[72] T. N. Kipf and M. Welling, ''Semi-supervised classification with graph convolutional networks,'' 2016, *arXiv:1609.02907*.

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

[74] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, ''T-GCN: A temporal graph convolutional network for traffic prediction,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[75] F. Wang, J. Xu, C. Liu, R. Zhou, and P. Zhao, ''On prediction of traffic flows in smart cities: A multitask deep learning based approach,'' *World Wide Web*, vol. 24, no. 3, pp. 805–823, May 2021.

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

[77] S. Wang, J. Cao, and P. S. Yu, ''Deep learning for spatio-temporal data mining: A survey,'' *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3681–3700, Aug. 2022.

[78] 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*.

[79] K. Zhang, F. He, Z. Zhang, X. Lin, and M. Li, ''Graph attention temporal convolutional network for traffic speed forecasting on road networks,'' *Transportmetrica B, Transp. Dyn.*, vol. 9, no. 1, pp. 153–171, Jan. 2021.

[80] D. Wang, Y. Yang, and S. Ning, ''DeepSTCL: A deep spatio-temporal ConvLSTM for travel demand prediction,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[81] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, and B. Yin, ''Metro passenger flow prediction via dynamic hypergraph convolution networks,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7891–7903, Dec. 2021.

[82] 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, pp. 914–921.

[83] C. Zheng, X. Fan, C. Wang, and J. Qi, ''GMAN: A graph multi-attention network for traffic prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[84] Y. Li and J. M. F. Moura, ''Forecaster: A graph transformer for forecasting spatial and time-dependent data,'' 2019, *arXiv:1909.04019*.

[85] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, ''Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting,'' *Trans. GIS*, vol. 24, no. 3, pp. 736–755, Jun. 2020.

[86] X. Wu, J. Fang, Z. Liu, and X. Wu, ''Multistep traffic speed prediction from Spatial–Temporal dependencies using graph neural networks,'' *J. Transp. Eng., A, Syst.*, vol. 147, no. 12, Dec. 2021, Art. no. 04021082.

[87] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, ''Learning dynamics and heterogeneity of spatial–temporal graph data for traffic forecasting,'' *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5415–5428, Nov. 2022.

[88] B. Lu, X. Gan, H. Jin, L. Fu, and H. Zhang, ''Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting,'' in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1025–1034.

[89] C. Shang, J. Chen, and J. Bi, ''Discrete graph structure learning for forecasting multiple time series,'' 2021, *arXiv:2101.06861*.

[90] D. Feng, Z. Wu, J. Zhang, and Z. Wu, ''Dynamic global-local spatial–temporal network for traffic speed prediction,'' *IEEE Access*, vol. 8, pp. 209296–209307, 2020.

[91] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li, ''A^{3}T-GCN: Attention temporal graph convolutional network for traffic forecasting,'' *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 7, p. 485, Jul. 2021.

[92] X. Wang, X. Guan, J. Cao, N. Zhang, and H. Wu, ''Forecast network-wide traffic states for multiple steps ahead: A deep learning approach considering dynamic non-local spatial correlation and non-stationary temporal dependency,'' *Transp. Res. C, Emerg. Technol.*, vol. 119, Oct. 2020, Art. no. 102763.

[93] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, ''Deep multi-view spatial–temporal network for taxi demand prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1.

[94] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, ''GSTNet: Global spatial–temporal network for traffic flow prediction,'' in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2286–2293.

[95] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, ''Dynamic spatial–temporal graph convolutional neural networks for traffic forecasting,'' in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 890–897.

[96] C. Zhang, J. J. Q. Yu, and Y. Liu, ''Spatial–temporal graph attention networks: A deep learning approach for traffic forecasting,'' *IEEE Access*, vol. 7, pp. 166246–166256, 2019.

[97] 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.

[98] H. Hong, Y. Lin, X. Yang, Z. Li, K. Fu, Z. Wang, X. Qie, and J. Ye, ''HetETA: Heterogeneous information network embedding for estimating time of arrival,'' in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2444–2454.

[99] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, ''Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks,'' *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2348–2359, May 2022.

[100] W. Luo, H. Zhang, X. Yang, L. Bo, X. Yang, Z. Li, X. Qie, and J. Ye, ''Dynamic heterogeneous graph neural network for real-time event prediction,'' in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3213–3223.

[101] X. Zhang, C. Huang, Y. Xu, and L. Xia, ''Spatial–temporal convolutional graph attention networks for citywide traffic flow forecasting,'' in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1853–1862.

[102] J. Zhang, Y. Zheng, and D. Qi, ''Deep spatio-temporal residual networks for citywide crowd flows prediction,'' in *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1.

[103] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, ''Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.

[104] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, ''Hierarchical graph convolution networks for traffic forecasting,'' in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 151–159.

[105] A. Roy, K. K. Roy, A. A. Ali, M. A. Amin, and A. K. M. M. Rahman, ''Unified spatio-temporal modeling for traffic forecasting using graph neural network,'' 2021, *arXiv:2104.12518*.

[106] S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, and A. Alemi, ''Watch your step: Learning node embeddings via graph attention,'' 2017, *arXiv:1710.09599*.

[107] Z. Liu, D. Zhou, and J. He, ''Towards explainable representation of time-evolving graphs via spatial–temporal graph attention networks,'' in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2137–2140.

[108] M. Fang, L. Tang, X. Yang, Y. Chen, C. Li, and Q. Li, ''FTPG: A fine-grained traffic prediction method with graph attention network using big trace data,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5163–5175, Jun. 2022.

[109] C. Pan, J. Zhu, Z. Kong, H. Shi, and W. Yang, ''DC-STGCN: Dual-channel based graph convolutional networks for network traffic forecasting,'' *Electronics*, vol. 10, no. 9, p. 1014, Apr. 2021.

[110] J. J. Q. Yu, C. Markos, and S. Zhang, ''Long-term urban traffic speed prediction with deep learning on graphs,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7359–7370, Jul. 2022.

[111] T. Zhang, W. Ding, T. Chen, Z. Wang, and J. Chen, ''A graph convolutional method for traffic flow prediction in highway network,'' *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–8, Jul. 2021.

[112] S. Fang, V. Prinet, J. Chang, M. Werman, C. Zhang, S. Xiang, and C. Pan, ''MS-Net: Multi-source spatio-temporal network for traffic flow prediction,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7142–7155, Jul. 2022.

[113] B. Sun, D. Zhao, X. Shi, and Y. He, ''Modeling global spatial–temporal graph attention network for traffic prediction,'' *IEEE Access*, vol. 9, pp. 8581–8594, 2021.

[114] Q. Zhou, J.-J. Gu, C. Ling, W.-B. Li, Y. Zhuang, and J. Wang, ''Exploiting multiple correlations among urban regions for crowd flow prediction,'' *J. Comput. Sci. Technol.*, vol. 35, no. 2, pp. 338–352, Mar. 2020.

[115] X. Yang, Q. Zhu, P. Li, P. Chen, and Q. Niu, ''Fine-grained predicting urban crowd flows with adaptive spatio-temporal graph convolutional network,'' *Neurocomputing*, vol. 446, pp. 95–105, Jul. 2021.

[116] S. Zhang, Y. Chen, and W. Zhang, ''Spatiotemporal fuzzy-graph convolutional network model with dynamic feature encoding for traffic forecasting,'' *Knowl.-Based Syst.*, vol. 231, Nov. 2021, Art. no. 107403.

[117] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, ''Fast unfolding of communities in large networks,'' *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[118] B. Perozzi, R. Al-Rfou, and S. Skiena, ''DeepWalk: Online learning of social representations,'' in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[119] A. Grover and J. Leskovec, ''node2vec: Scalable feature learning for networks,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[120] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, ''AM-GCN: Adaptive multi-channel graph convolutional networks,'' in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1243–1253.

HONGFAN MU received the bachelor's degree in computer science from Chongqing University, China. She is currently pursuing the master's degree in computer science (applied artificial intelligence) with the Department of Electrical Engineering, University of Ottawa, Canada. Her research interests include applied artificial intelligence, spatio-temporal dependency, time series forecasting, and architectural design of AI solutions.

NOURA ALJERI (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Ottawa, Canada. She is currently an Assistant Professor with the Computer Science Department, Kuwait University, Kuwait, and a Research Fellow with the PARADISE Research Laboratory, University of Ottawa, Canada. She is a member of ACM. She was a recipient of the most outstanding Ph.D. thesis Award from the University of Ottawa, in 2020. She was also a recipient of the Pierre Laberge Award for her research contribution on mobility management for autonomous and connected vehicular networks, in 2021. Her current research interests include smart mobility, topology management, prediction models for connected and autonomous vehicular networks, smart transportation systems, and mobility networks. She has published extensively in those areas and she received the best paper award in the 16th IEEE AICCSA Conference. She served as a member of the Technical Program Committee for several ACM and IEEE conferences, including MSWIM, PE-WASUN, and ICC. She served on the editorial board for ACM ICPS and the Technical Program Track Co-Chair for IEEE Globecom'24 IoT and Sensor Networks Track. She is also an Associate Editor of *ACM Computing Surveys*.

AZZEDINE BOUKERCHE (Fellow, IEEE) is currently a Distinguished University Professor and the Canada Research Chair Tier-1 with the University of Ottawa, where he is also the Founding Director of the PARADISE Research Laboratory and the DIVA Strategic Research Centre. His current research interests include wireless ad hoc and sensor networks, wireless networking and mobile computing, wireless multimedia, QoS service provisioning, performance evaluation and modeling of large-scale distributed and mobile systems, and large scale distributed and parallel discrete event simulation. He has published extensively in these areas and received several best research paper awards for his work. He is a fellow of the Engineering Institute of Canada, Canadian Academy of Engineering, and American Association for the Advancement of Science. He has received the C. Gotlieb Computer Medal Award, Ontario Distinguished Researcher Award, Premier of Ontario Research Excellence Award, G. S. Glinski Award for Excellence in Research, IEEE Computer Society Golden Core Award, IEEE CS-Meritorious Award, IEEE TCPP Leaderships Award, IEEE ComSoc ASHN Leaderships and Contribution Award, and University of Ottawa Award for Excellence in Research. He serves as an associate editor for several IEEE Transactions and ACM journals and is also the steering committee chair for several IEEE and ACM international conferences.