AARGNN: An Attentive Attributed Recurrent Graph Neural Network for Traffic Flow Prediction Considering Multiple Dynamic Factors

Ling Chen, Wei Shao, Mingqi Lv, Weiqi Chen, Youdong Zhang, and Chenghu Yang

***Abstract*— Traffic flow prediction is a fundamental part of ITS (Intelligent Transportation System). Since the correlations of traffic data are complicated and are affected by various factors, traffic flow prediction is a challenging task. Existing traffic flow prediction methods generally take limited static factors (e.g., the distance between sensors and road network topological structure) into consideration and model the correlations of the traffic data separately to predict the future traffic. In this paper, we propose AARGNN (Attentive Attributed Recurrent Graph Neural Network), a GNN (graph neural network) based method considering multiple dynamic factors to predict short-term traffic flow. With multi-source urban data (e.g., POI, road network, incident, weather, etc.), AARGNN considers both static factors and dynamic factors (e.g., spatial distance, semantic distance, road characteristic, road situation, and global context) to predict the short-term traffic flow. Specifically, AARGNN constructs an attributed graph and encodes various factors into the attributes. The correlations of the traffic data are modeled by utilizing the GNN combined with LSTM (long short-term memory). In addition, AARGNN specifies the contributions of each factor based on attention mechanism. Experiments on real-world datasets show that the proposed method outperforms all baseline methods.**

***Index Terms*— Traffic flow prediction, graph neural network, urban computing.**

I. Introduction

TRAFFIC flow prediction, especially short-term traffic flow prediction, has been a focus issue in ITS. The task of traffic flow prediction is to predict the future traffic flow in the road network by leveraging history traffic data and related urban data. Accurate short-term traffic flow prediction not only facilitates people's daily commuting, but also provides great help to governments in transportation management [1].

Manuscript received 17 June 2020; revised 3 February 2021, 22 July 2021, 24 December 2021, and 3 March 2022; accepted 6 April 2022. Date of publication 4 May 2022; date of current version 11 October 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0505000 and in part by the Joint Funds of the National Natural Science Foundation of China under Grant U1936215. The Associate Editor for this article was P. Wang. *(Corresponding author: Mingqi Lv.)*

Ling Chen, Wei Shao, and Weiqi Chen are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: lingchen@cs.zju.edu.cn; shaowei@cs.zju.edu.cn; vc12301@gmail.com).

Mingqi Lv is with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: mingqilv@zjut.edu.cn).

Youdong Zhang and Chenghu Yang are with Alibaba Group, Hangzhou 311100, China (e-mail: flinqing.zyd@alibabainc.com; yexiang.ych@taobao.com).

Digital Object Identifier 10.1109/TITS.2022.3171451

There have been a lot of efforts made in the traffic flow prediction, which mainly fall into two categories: the traditional methods and the recent deep-learning based methods. The traffic data are correlated in temporal, spatial, and semantic aspects. Based on the correlations modeled in prediction process, the traditional methods could be divided into univariate time series learning methods and multivariate time series learning methods. Univariate time series learning methods, e.g., non-parameter regression-based methods [2], Kalman filtering-based methods [3], and ARIMA (auto regressive integrated moving average)-based methods [4]–[6], mainly focus on the temporal correlations of the traffic data from a single sensor. These methods can capture the temporal patterns but ignore the spatial and semantic correlations from multiple sensors. Multivariate time series learning methods, e.g., KNN (k-nearest neighbor)-based methods [7], Bayesian network-based methods [8], STARIMA (spatiotemporal autoregressive integrated moving average)-based methods [9], SVM (support vector machi nes)-based methods [10], [11], and ANN (artificial neural networks)-based methods [12], [13], consider the spatial/semantic correlations of the traffic data from multiple sensors. However, these methods usually utilize the traffic data from sensors within a fixed range, which is either too small to incorporate necessary information, or too large that leads to high computation complexity and overfitting.

The recent deep-learning based methods for traffic flow prediction can model the correlations of the traffic data from multiple sensors and expand the range of related sensors along with iterations. CNN (convolutional neural network)-based methods [14]–[16] represent the traffic data from multiple sensors as an image to capture the spatial correlations of the traffic data. However, traffic sensors are not regularly deployed, and thus GNN (graph neural network)-based methods [18]–[35] represent the traffic data from multiple sensors as a graph and the correlations are modelled as edges in the graph. Previous GNN-based methods usually construct a graph with fixed structure and consider only static factors. However, the correlations of the traffic data are affected by not only static factors but also dynamic factors.

Recently, spatial-based GNN [36] is proposed. Spatial-based GNN can support flexible representations of the attributes of the graph. Thus, they can handle graph-structured data with dynamic node, edge, and global attributes.

In this paper, we propose AARGNN (attentive attributed recurrent graph neural network), a GNN-based method to predict short-term traffic flow. AARGNN takes both static factors (e.g., spatial distance, semantic distance, road characteristic) and dynamic factors (e.g., road situation, and global context) into consideration when modeling the correlations of the traffic data, and quantifies the contributions of each factor based on the attention mechanism. Our contributions are summarized as follows:

(1) We identify factors that affect the correlations of the traffic data based on multi-source urban data and construct an attributed graph for the traffic data to consider both static and dynamic factors.

(2) We propose a hybrid deep neural network to model the temporal, spatial, and semantic correlations of the traffic data for traffic flow prediction. We encode various factors into the attributes of the graph and combine GNN with LSTM to jointly model the correlations of the traffic data. In addition, we utilize the attention mechanism to quantify the contributions of each factor.

(3) We evaluate AARGNN on real-world traffic datasets and make comparisons with other traffic flow prediction methods. The results show that the proposed method is superior to the compared methods.

II. Related Work

Traffic flow prediction is a time series problem, which mainly focuses on predicting the future traffic data in the road network. Accurate prediction of traffic flow plays an important role in many real-world applications. For example, traffic flow prediction is the basis of traffic control systems, along with traffic estimation [38], [39] and traffic state detection [40], [41]. Generally, previous traffic flow prediction methods could be divided into two categories: the traditional methods and the recent deep-learning based methods.

The traditional methods could be divided into univariate series methods and multivariate series methods based on the correlations modeled in prediction process. It has been proved that the traffic data from multiple sensors are correlated in temporal, spatial, and semantic aspects [37]. Univariate series methods mainly focus on the temporal correlations of the traffic data and do not distinguish the traffic data from different sensors. Representative univariate series methods, e.g., ARIMA-based methods, non-parameter regression-based methods, and Kalman filtering-based methods, can capture the patterns of univariate series. For example, Smith *et al.*. [2] applied non-parameter regression to predict the single interval traffic flow on motorways. Iwao and Stephanedes [3] applied Kalman filtering theory to predict the traffic flow both for raw data collected from road networks and for data obtained by an exponential smoothing technique. Hamed *et al.*. [5] applied ARIMA to predict the short-term traffic flow in urban arterial roads. These methods ignore the spatial and semantic correlations of the traffic data from multiple sensors, thus cannot yield the best outcomes for predicting the traffic flow at network level.

Realizing the limitation of univariate series methods, multivariate series methods consider the spatial correlations and the semantic correlations of the traffic data from multiple sensors to predict the traffic flow. For example, Cai *et al.*. [7] applied KNN to predict the short-term traffic speed based on the spatial correlations of the traffic data. Sun *et al.*. [8] utilized the spatial correlations to select related sensors to predict the traffic flow based on Bayesian networks. Wanli and Wynter [9] incorporated the adjacency matrix of sensors to predict the real-time road traffic based on STARMA, an extension of ARIMA method. Furthermore, more complicated methods are incorporated to model the spatial/semantic correlations of traffic data. For example, Hong *et al.*. [10] regarded the prediction task as nonlinear regression problems and applied SVR to predict the traffic flow. Kranti *et al.*. [13] applied ANN to model the non-linear correlations and used 45 minutes data as the input of ANN to predict the traffic flow data 15 minutes in future. These methods utilize the traffic data from related sensors within a fixed range, which is either too small to incorporate necessary information, or too large that leads to high computation complexity and overfitting.

The recent deep-learning based methods can model the non-linear correlations of the traffic data from multiple sensors and expand the range of related sensors along with iterations. CNN-based methods are popular deep learning methods, which represent the traffic data from multiple sensors at a time interval as an image to capture the spatial correlations of the traffic data. For example, Yu *et al.*. [14] also represent the traffic data as frames of images and predict the traffic speed based on CNN combined with LSTM. Ma *et al.*. [15] learned the traffic data as images to predict the large-scale traffic speed based on CNN. Wu *et al*. [12] combined CNN with the attention mechanism to predict the traffic flow. Wang *et al.* [17] proposed a deep attentive adaptation network to transfer cross-domain spatio-temporal knowledge for urban crowd flow prediction. CNN is originally designed for dealing with the data in Euclidean space, while traffic data recorded by traffic sensors are not regularly deployed. Thus, CNN-based methods cannot fully capture the sptial correlations of the traffic data.

To model the correlations of the traffic data from traffic sensors not regularly deployed, GNN-based methods are proposed. GNN-based methods represent the traffic data from multiple sensors at a time interval as a graph and the correlations are modelled as the edges in the graph. For example, Yu *et al.*. [18] proposed completely convolutional structures, STGCN, on the traffic data to capture both the spatial and temporal correlations. Geng *et al.*. [37] proposed ST-MGCN to model region-wise relationships under multiple modalities for ride-hailing demand forecast. Li *et al.*. [19] proposed DCRNN to model the spatial correlations using bidirectional random walks on a graph constructed with the traffic network and history traffic data based on GNN. Cui *et al*. [28] proposed TGC-LSTM to predict the traffic speed by combining GNN with GRU. Lv *et al.*. [35] proposed T-MGCN, which jointly models the spatial/semantic correlations from different aspects for traffic flow prediction by fusing multiple graphs. Geng *et al.* [29] proposed grouped GCN to produce a compound graph connectivity on multi-modality graph representation and adapted multi-linear relationship GCN to learn better coordinated representations among modalities.

In addition, to utilize the extern factors related to traffic data, some works utilize attributes to improve prediction accuracy. For example, Zhu *et al.* [30] proposed an attribute-augmented spatiotemporal graph convolutional network, which models external factors as dynamic attributes and static attributes and encodes these attributes to predict traffic speed. Pan *et al.*. [31] leveraged attributes to capture the inherent relationships and proposed a deep meta learning based model to predict traffic data. Yang *et al.* [32] designed a path-based speed prediction neural network with given path and attributes information to predict traffic speed. These GNN-based traffic flow prediction methods tend to construct fixed structure of the graph and fail to handle the graph-structured traffic data with dynamic edge attributes.

To handle the dynamic edge attributes, several studies are conducted. For example, Guo *et al.*. [21] proposed AST-GCN with the attention mechanism to learn the dynamic correlations of traffic data. Guo *et al.*. [26] applied a graph updating strategy to construct an optimized graph matrix in the training step of the graph to predict the traffic data. Shin and Yoon [22] incorporated structure features of transportation networks to capture the dynamic attributes of the graph. Chen *et al.* [23] introduced the bicomponent graph convolution to explicitly model the relationship of both nodes and edges. Cui *et al.* [33] learned the traffic network as a graph and incorporated the graph wavelet instead of graph convolution to extract well-localized features to predict traffic flow. Chen *et al.* [34] proposed a spectral spatial retrieval graph convolutional block to extract the geographical structure of the traffic network from global and local perspectives to predict traffic flow. However, the correlations of the traffic data are affected by multiple factors, and previous methods only can consider limited static factors, which cannot fully model the complicated dynamic correlations. Recently, spatial-based GNN is proposed, which formulates graph convolutions as aggregating feature information from neighbors. Spatial-based GNN supports flexible representations of attributes as well as different graph structures. For example, Battaglia *et al.*. [36] proposed the Graph Network (GN) framework, which generalizes and extends various GNNs, e.g., message-passing neural network [43] and non-local neural network [44]. The core computation unit of the GN framework, i.e., GN blocks, can handle the graph-structured data with dynamic node, edge, and global attributes. In this paper, we represent the traffic data as attributed graphs and employ spatial-based GNN to model the complicated dynamic correlations considering multiple factors.

III. Method

A. Preliminaries

In this section, we firstly give the definitions of the basic concepts and terms, and then formulate the problem.

1) Notation: In this paper, upper-case letters denote sets and lower-case letters denote variables. Bold upper-case letters denote matrices and bold lower-case letters denote vectors. [] denotes a sequence, {} denotes a set, and <> denotes a vector. || denotes the operation of vector concatenation.

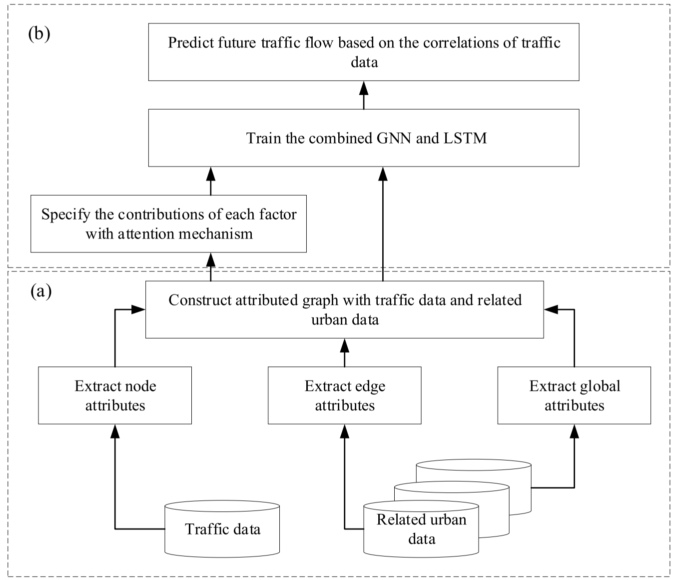

Fig. 1. The architecture of AARGNN: (a) Attributed graph construction. (b) Traffic flow prediction.

2) Definition 1 (Graph-Structured Traffic Data): The traffic data in the road network can be structured as an attributed graph $G = (N, E, \boldsymbol{V}, \boldsymbol{A}, \boldsymbol{u})$, where N is the set of nodes corresponding to the sensors deployed in the road network, E is the set of edges corresponding to the correlations of different sensors, $\boldsymbol{V}$ represents the attributes of the nodes, $\boldsymbol{A}$ represents the attributes of the edges, and $\boldsymbol{u}$ represents the global attributes of the graph.

The node attributes consist of the traffic data and other node-level factors. The edge attributes represent the correlations between two sensors, which are affected by both static factors and dynamic factors. The global attributes are features shared by all the nodes in the graph.

3) Problem Formulation: Let $G^t = (N, E, \boldsymbol{V}^t, \boldsymbol{A}^t, \boldsymbol{u}^t)$ denote the graph-structured traffic data at time slot t, where $\mathbf{V}^t$, $\boldsymbol{A}^t$, and $\boldsymbol{u}^t$ represent different attributes at time slot t, Y^t represents the traffic flow at time slot t. Then, the short-term traffic flow prediction problem can be formulated as: given τ time slots of the graph-structured traffic data $[G^{t-\tau+1}, G^{t-\tau+2}, \ldots, G^t]$, learning a function $h()$that maps the input data to the future traffic flow:

$$[G^{t-\tau+1}, G^{t-\tau+2}, \ldots, G^t] \xrightarrow{h()} Y^{t+l} \tag{1}$$

B. Architecture

As shown in Fig. 1, AARGNN is composed of two parts: attributed graph construction and traffic flow prediction. The attributed graph construction part structures the traffic data as an attributed graph with node attributes, edge attributes, and global attributes, and encodes various factors into the graph. The traffic flow prediction part utilizes the attention mechanism to quantify the contributions of each factor and models the correlations of the traffic data with a hybrid network with GNN and LSTM.

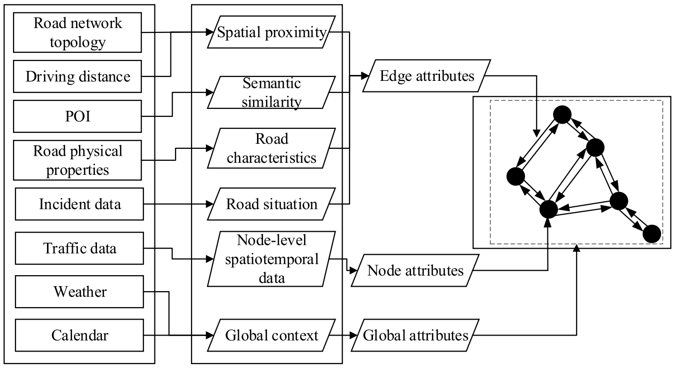

Fig. 2. Attributed graph construction.

C. Attributed Graph Construction

The urban traffic data are correlated in temporal, spatial, and semantic aspects. On one hand, the traffic data from the same sensor are temporally correlated. On the other hand, the traffic data from different sensors are spatially and semantically correlated. In order to capture the correlations of the traffic data, we structure the traffic data and related urban data as an attributed graph and encode both static and dynamic factors into the attributes of the graph, as shown in Fig. 2. Specifically, given the traffic data and related urban data, we can construct an attributed graph with node attributes, edge attributes, and global attributes for each time slot. For the attributed graph at time slot t, the nodes represent sensors in the road network and the edges represent correlations of different sensors. The node attributes represent the current attributes of the sensors. The edge attributes and global attributes represent the factors that affect the correlations of the traffic data from different sensors. To simplify the model, we construct the graph as a directed graph according to the topology of the traffic system, i.e., there is an edge between two sensors if there is a road segment connecting one sensor to the other.

1) Node Attributes: The nodes in the graph represent the sensors in the traffic system. We aggregate the traffic flow data and the node-level spatiotemporal data (e.g., vehicle occupancy) as the node attributes. The traffic flow data are the target to predict. The vehicle occupancy data are used as dynamic factor in node attributes. Suppose there are n_{se} nodes in the graph, then the node attributes at time slot t are defined as $\boldsymbol{V}^t = \{\boldsymbol{v}_i^t\}_{i=1:n_{\mathrm{se}}}$, where $\boldsymbol{v}_i^t = <\boldsymbol{f}_1^t||\boldsymbol{f}_2^t||\ldots||\boldsymbol{f}_{z_1}^t>$ and $\boldsymbol{f}_k^t$ is one of the feature vectors of the traffic data or related node-level spatiotemporal data at time slot t.

2) Edge Attributes: The edges in the graph represent the correlations of pair-wise sensors, which may be affected by various factors. For example, if there is an incident happened on the road, the correlations of the traffic data between two sides of the road would change. To model the correlations, we construct the attributes of an edge with static factors and dynamic factors. Suppose there are n_{ed} edges in the graph, then the edge attributes at time slot t are defined as $\boldsymbol{A}^t = \{\boldsymbol{a}_i^t\}_{i=1:n_{\mathrm{ed}}}$, where $\boldsymbol{a}_i^t = <\boldsymbol{s}_1^t||\boldsymbol{s}_2^t||\ldots||\boldsymbol{s}_{z_2}^t>$ and $\mathbf{s}_k^t$ is one of the feature vector of the factors that affect traffic data correlations.

Previous traffic flow prediction methods ignore the impact of dynamic factors when modeling the correlations. To take both static and dynamic factors into consideration, the attributes of an edge contain two parts: the static part and the dynamic part. We extract three sets of static factors from related urban data for the static part as follows:

(1) **Spatial proximity** measures the sensor correlations based on geographic distances. Let $dis(i,j)$ denote the road driving distance between two upstream sensor i and downstream sensor j, then $\boldsymbol{s}_{\mathrm{sp}(i,j)} = 1/dis(i,j)$ is the spatial proximity of the edge connecting i and j.

(2) **Semantic similarity** measures the sensor correlations from semantic aspect. POIs in the region where a sensor is deployed indicate the land use of the region, which can be utilized to calculate the semantic distance of two sensors. Given the location of a sensor, we collect POIs around the sensor and calculate the density distribution of the POIs of different categories. Suppose there are n_{poi} categories of POIs, the land use of a sensor i can be formulated as a vector $\boldsymbol{p}_i$ of length n_{poi}, where each dimension denotes the density of nearby POIs of a specific category. Then, the semantic similarity of the edge connecting sensor i and j can be calculated as $\boldsymbol{s}_{\mathrm{lu}(i,j)} = \boldsymbol{p}_i \cdot \boldsymbol{p}_j$.

(3) **Road characteristics** refer to the physical properties of the road between two sensors. In our method, we consider road width, the number of road lanes, and road speed limit as the road characteristic features. Let $\boldsymbol{s}_{\mathrm{rc}(i,j)}$ denote the road characteristics of the edge connecting sensor i to sensor j, then $\boldsymbol{s}_{\mathrm{rc}(i,j)} = <\boldsymbol{r}_1||\boldsymbol{r}_2||\ldots||\boldsymbol{r}_{z_3}>$ and $\boldsymbol{r}_k$ is one of the road characteristic features.

The dynamic part contains dynamic factors, which would change over time. In our method, we utilize road situation represented by the incidents happened on the road as dynamic factors. Suppose there are n_{inc} types of incidents, the road situation at time slot t can be formulated as $\boldsymbol{s}_{\mathrm{rs}(i,j)}^t \in \mathbb{R}^{n_{\mathrm{inc}}}$, and the k-th dimension denotes the influence of the k-th type of incident on the correlations of the traffic data. Specifically, we set the value to 0 if the corresponding type of incident happens on the road, and 1 otherwise.

Since both static factors and dynamic factors can affect the correlations of the traffic data, we concatenate both types of factors as the edge attributes of the graph. The static factors remain unchanged while the dynamic factors change with time. Thus, for the edge attributes at time slot t, the edge attributes are $\boldsymbol{A}^t = \{\boldsymbol{a}_i^t\}_{i=1:n_{\mathrm{ed}}}$, where $\boldsymbol{a}_i^t = <\boldsymbol{s}_{\mathrm{sp}}||\boldsymbol{s}_{\mathrm{lu}}||\boldsymbol{s}_{\mathrm{rc}}||\boldsymbol{s}_{\mathrm{rs}}^t>$ and n_{ed} is the number of edges.

Global attributes The global attributes of the graph refer to the factors shared by all sensors. Since the static factors shared by all sensors may not affect prediction results, we mainly consider the dynamic factors in the global attributes. In our method, we consider weather condition and calendar features as the global attributes, which are strongly related to the traffic data [37]. The calendar features include time of day, day of week, day of month, month of year, and day type (i.e., work day or vacation). The weather condition and current calendar features of all sensors in a city are regarded as the same. Let $\boldsymbol{u}^t$ represent the global attributes of the attributed graph at time slot t, then $\boldsymbol{u}^t = <\boldsymbol{u}_{\mathrm{wea}}^t||\boldsymbol{u}_{\mathrm{cal}}^t>$, where $\boldsymbol{u}_{\mathrm{wea}}^t$ denotes

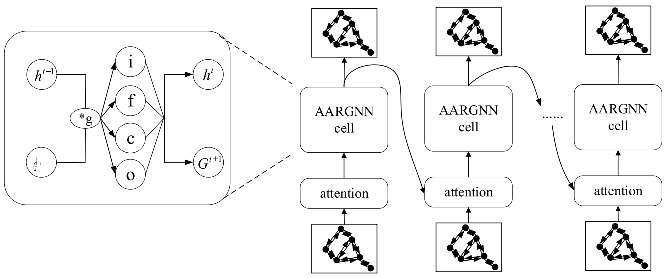

Fig. 3. Overview of AARGNN.

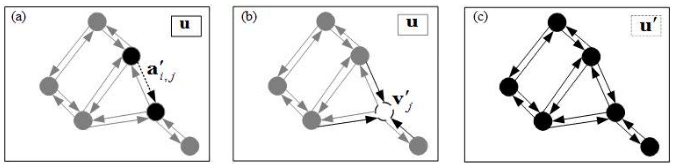

Fig. 4. Convolution on the graph-structured traffic data: (a) Update the edge attributes. (b) Update the node attributes. (c) Update the global attributes.

the weather condition feature and $\boldsymbol{u}_{\mathrm{cal}}^t$ denotes the calendar features.

D. Traffic Flow Prediction

Based on the graph-structured traffic data, the correlations of the traffic data are modeled with a hybrid network with GNN and LSTM, as shown in Fig. 3. The GNN is utilized to model the spatial and semantic correlations and the LSTM is utilized to model the temporal correlations. In addition, to quantify the contributions of various factors, we add the attention mechanism to the attributes of the graph-structured traffic data.

When modeling the spatial and semantic correlations, most previous methods are based on spectral-based GNN, i.e., ChebNet [42] and its extensions. These methods are efficient if the structure of the graph is fixed and a global adjacency matrix is predefined. However, if the adjacency matrix needs to be updated frequently, the convolution operation would become inefficient. Moreover, spectral-based GNN cannot handle the flexible edge attributes. Thus, we utilize spatial-based GNN to model the spatial and semantic correlations instead of spectral-based GNN. The spatial-based GNN generalizes and extends previous GNNs, e.g., message-passing neural network [43] and non-local neural network [44], which can learn the correlations of the traffic data and handle flexible edge attributes.

We conduct the convolution on the graph-structured traffic data to model the spatial and semantic correlations as a "graph to graph" process. Generally, the convolution operation is processed by updating the edge attributes, followed by the node attributes, and finally the global attributes, as shown in Fig. 4. The process is implemented via updating functions ϕ and aggregation functions ρ.

Let G^t denote the input graph-structured traffic data of the GNN at time slot t. Hereafter, we omit the superscript t for simplicity. Then, the convolution $*_g$ on the graph-structured traffic data $G = (N, E, \boldsymbol{V}, \boldsymbol{A}, \boldsymbol{u})$ is defined as:

$$\boldsymbol{a}'_{i,j} = \phi^a(\boldsymbol{a}_{i,j}, \boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{u}) \tag{2}$$

$$\bar{\boldsymbol{a}}'_{\mathcal{N}(i)} = \rho^{a\to v}(\boldsymbol{A}'_{\mathcal{N}(i)}) \tag{3}$$

$$\boldsymbol{v}'_i = \phi^v(\bar{\boldsymbol{a}}'_{\mathcal{N}(i)}, \boldsymbol{v}_i, \boldsymbol{u}) \tag{4}$$

$$\boldsymbol{v}' = \rho^{v\to u}(\boldsymbol{V}') \tag{5}$$

$$\bar{\boldsymbol{a}}' = \rho^{a\to u}(\boldsymbol{A}') \tag{6}$$

$$\boldsymbol{u}' = \phi^u(\bar{\boldsymbol{a}}', \bar{\boldsymbol{v}}', \boldsymbol{u}) \tag{7}$$

where $\boldsymbol{a}_{i,j}$ denotes the attribute of the edge whose upstream node is i and downstream node is j, $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ denote the attributes of node i and node j, and $\boldsymbol{u}$ denotes the global attributes. $A'_{N(i)} = \{\boldsymbol{a}'_{j,i}\}_{j=1:n_i}$ represents the updated attributes of the edges connected to node i and n_i is the number of edges connected to node i. $V' = \{\boldsymbol{v}'_i\}_{i=1:n_{\mathrm{se}}}$ represents the updated node attributes and n_{se} is the number of nodes. $\boldsymbol{A}' = \cup_r \boldsymbol{A}'_r = \{\boldsymbol{a}'_r\}_{r=1:n_{\mathrm{ed}}}$ represents all per-edge outputs. Firstly, the edges are updated by ϕ^a as Equation 2. Here, ϕ^a is a function, which is used to map across all edges to compute per-edge updated attributes. The updated edge attribute $\boldsymbol{a}'_{i,j}$ can be regarded as the influence from the upstream node to the downstream node. Secondly, the influence on each node from connected nodes is aggregated by $\rho^{a\to v}$ as Equation 3, which is a function that takes all the updated edge attributes connected to a certain node as the input and reduces it to one vector to represent the aggregated information. Thirdly, each node is updated by ϕ^v as Equation 4, which is a function of the aggregated influences, the current node attributes, and the current global attributes. Fourthly, the elements representing the aggregated information of updated edge attributes and node attributes are computed by $\rho^{v\to u}$and $\rho^{a\to u}$ as Equation 5 and Equation 6. Finally, the global attributes are updated by ϕ^u as Equation 7, which is a function of the current global attributes and aggregated information of the node and edge attributes.

To model the temporal correlations, RNNs are employed in the prediction process. LSTM is a variant of RNNs, which can prevent the gradient from vanishing or exploding problem. Inspired by previous work [14], [16], [36]–[42], we combine GNN with LSTM by replacing the matrix multiplications in the LSTM cell with GNN operation to model the temporal, spatial, and semantic correlations jointly. Let $*_g$ denote the convolution on the graph-structured traffic data, G^t denote the input graph-structured traffic data of the LSTM at time slot t, then the LSTM model can be modified as:

$$\boldsymbol{i} = \sigma(\boldsymbol{W}_{g_i} *_g G^t + \boldsymbol{W}_{h_i} *_g \boldsymbol{h}^{t-1} + \boldsymbol{W}_{c_i} \odot \boldsymbol{c}^{t-1} + \boldsymbol{b}_i) \tag{8}$$

$$\boldsymbol{f} = \sigma(\boldsymbol{W}_{g_f} *_g G^t + \boldsymbol{W}_{h_f} *_g \boldsymbol{h}^{t-1} + \boldsymbol{W}_{c_f} \odot \boldsymbol{c}^{t-1} + \boldsymbol{b}_f) \tag{9}$$

$$\begin{aligned}\boldsymbol{c}^t = &\ \boldsymbol{f} \odot \boldsymbol{c}^{t-1} + \boldsymbol{i} \odot tanh(\boldsymbol{W}_{g_c} *_g G^t \\ &+ \boldsymbol{W}_{h_c} *_g \boldsymbol{h}^{t-1} + \boldsymbol{b}_c)\end{aligned} \tag{10}$$

$$\boldsymbol{o} = \sigma(\boldsymbol{W}_{g_o} *_g G^t + \boldsymbol{W}_{h_o} *_g \boldsymbol{h}^{t-1} + \boldsymbol{W}_{c_o} \odot \boldsymbol{c}^{t-1} + \boldsymbol{b}_o) \tag{11}$$

$$\boldsymbol{h}^t = \boldsymbol{o} \odot tanh(\mathbf{c}^t) \tag{12}$$

where $\odot$ denotes the Hadamard product, $\sigma(\cdot)$ denotes the sigmoid function, $\boldsymbol{i}$, $\boldsymbol{f}$, and $\boldsymbol{o}$ are the input, forget, and output gates, $\boldsymbol{h}$ is the hidden state, $\boldsymbol{c}$ is the cell state, and $\boldsymbol{W}$ are the weights. The LSTM layer is shared by all the nodes in the graph.

The correlations of the traffic data are affected by various factors. However, the contributions of each factor are not equal, and would dynamically vary with sensors and time slots. To quantify the contributions of each factor, we utilize the attention mechanism in our method.

As defined in Section III-A, the traffic data at a time slot are structured as a graph $G^t = (N, E, \boldsymbol{V}^t, \boldsymbol{A}^t, \boldsymbol{u}^t)$. With the static and dynamic factors encoded in the graph, each type of attributes consists of multiple feature vectors. Suppose there are n_{fa} types of factors encoded in attribute $\boldsymbol{v}^t$, i.e., $\boldsymbol{v}^t = \{\boldsymbol{v}_k^t\}_{k=1:n_{\text{fa}}}$, where $\boldsymbol{v}_k^t$ is the feature vector of the k-th factor at time slot t. The attention mechanism is defined as:

$$\chi_k^t = z_l\, tanh(\boldsymbol{W}_l \boldsymbol{h}^{t-1} + \boldsymbol{U}_l \boldsymbol{v}_k^t + \boldsymbol{b}_l) \quad (13)$$

$$\varepsilon_k^t = \frac{exp(\chi_k^t)}{\sum_{m=1}^{n_{\text{fa}}} exp(\chi_m^t)} \quad (14)$$

$$\tilde{v}_k^t = \varepsilon_k^t \boldsymbol{v}_k^t \quad (15)$$

where $\boldsymbol{z}_l$, $\boldsymbol{W}_l$, $\boldsymbol{U}_l$, and $\boldsymbol{b}_l$ are learnable parameters and $\boldsymbol{h}^{t-1}$ is the hidden states. The attention mechanism would assign higher weights to the factors that contribute more to the current traffic state, which are also applied to edge attributes and global attributes. For edge attributes, we define the hidden state of the edge connecting node i to j as the combination of corresponding hidden states $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. For global attributes, we define the hidden state as the summary of all hidden states of the nodes.

To obtain the final prediction result, we add a fully connected layer to the output of the LSTM layer. The entire network is trained by minimizing the error of the predicted traffic flow using backpropagation through time. Suppose the output of combined GNN and LSTM layer is Y^{t+l} and the ground truth is Y_0^{t+l}, then the loss function is defined as:

$$L = \sum \left\| Y_0^{t+l} - Y^{t+l} \right\|_2^2 + \lambda_1 \left\| \boldsymbol{W}_1 \right\|_2 + \lambda_2 \left\| \boldsymbol{W}_2 \right\|_2 \quad (16)$$

where $\mathbf{W}_1$, $\mathbf{W}_2$ are the weight matrices in combined GNN and LSTM and the attention mechanism, and λ_1, λ_2 are penalty terms.

IV. Experiments

A. Experimental Setup

1) Datasets: We evaluate our method on two real-world traffic datasets, i.e., PEMS and HZJTD.

2) PEMS Dataset: The PEMS dataset is collected from the website of California Department of Transportation,which stores the data from CalTrans Performance Measurement System (PeMS). PeMS is a real-time Archive Data Management System that collects, stores, and processes raw traffic data in real-time. We select two parts of the PEMS dataset in experiments, PEMS-S and PEMS-L. The PEMS-S dataset contains traffic data of 30 sensors deployed in Stockton, and the time period is from January 1st 2017 to March 31th 2017. The traffic data are aggregated every 5 minutes. The PEMS-L dataset contains traffic data of 362 sensors deployed in the District 10 of California. The traffic data are also aggregated every 5 minutes, and the time period is also from January 1st 2017 to March 31th 2017.

Related urban data of the PEMS dataset include POI data, geographic data, road characteristics data, incident data, and weather data. The POI data are collected form Google Maps, which contain the details of POIs, i.e., name, location, categories, etc. The geographic data are collected from OpenStreetMap, which contain the topology of the road network and the driving distance between each pair of sensors. The road characteristics and the incident data are collected from PeMS. The road characteristics data contain the details of road width, the number of road lanes, and road speed limit. The incident data contain the incident category and delay time happened on the road. The weather data are collected from the California Irrigation Management Information System (CIMIS), which contain solar radiation, soil temperature, air temperature, relative humidity, and preipitation. We collect the data according to the areas and time periods corresponding to the two traffic datasets.

3) HZJTD Dataset: The HZJTD dataset is collected from Hangzhou Integrated Transportation Research Center. The HZJTD dataset contains the traffic data from the sensors of 202 roads in Hangzhou. The time period of the collected traffic data is from October 16th 2013, to October 3rd, 2014. The traffic data are aggregated every 15 minutes.

Related urban data of the HZJTD dataset include POI data and geographic data. The POI data are collected from Baidu Map API. The geographic data contain the topology of the road network and the driving distance between each pair of sensors. Different from the PEMS dataset, the incident data and weather data related to the HZJTD dataset are unavailable. Therefore, the attributed graph of HZJTD is constructed with traffic data, POI data, geographic data, and calendar features.

4) Evaluation Metrics: We select traffic flow as the predicting target and measure the performance of the prediction based on RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error).

Let $y_i^t \in Y^t$ denote the ground-truth and $\tilde{y}_i^t \in \tilde{Y}^t$ denote the predicted target traffic data, then the evaluation metrics are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n_{\text{se}}} \sum_{i=1}^{n_{\text{se}}} (y_i^t - \tilde{y}_i^t)^2} \quad (17)$$

$$\text{MAE} = \frac{1}{n_{\text{se}}} \sum_{i=1}^{n_{\text{se}}} \left| y_i^t - \tilde{y}_i^t \right| \quad (18)$$

$$\text{MAPE} = \frac{1}{n_{\text{se}}} \sum_{i=1}^{n_{\text{se}}} \left| \frac{y_i^t - \tilde{y}_i^t}{y_i^t} \right| \times 100\% \quad (19)$$

where n_{se} is the number of all predicted values.

B. Parameters

First, we empirically set the learning rate to 0.001, the batch size to 32, and the training epoch to 5000. The datasets are split in chronological order with the first 70% for training, the last 20% for test, and the other 10% for validation. Second, we perform a grid search for the number of AARGNN layers $\in \{1, 2, 3\}$ and the number of LSTM hidden units $\in \{16, 32, 64, 128\}$. The optimal number of AARGNN layers is 2 and the optimal number of LSTM hidden units is 32.

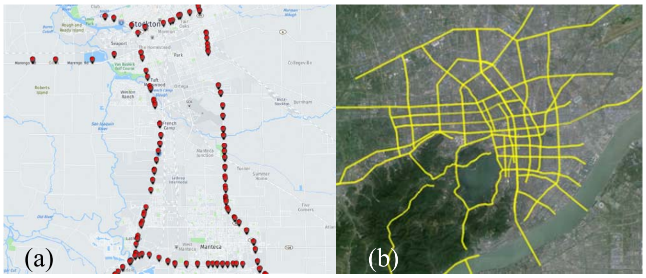

Fig. 5. Overview of the datasets: (a) PEMS. (b) HZJTD.

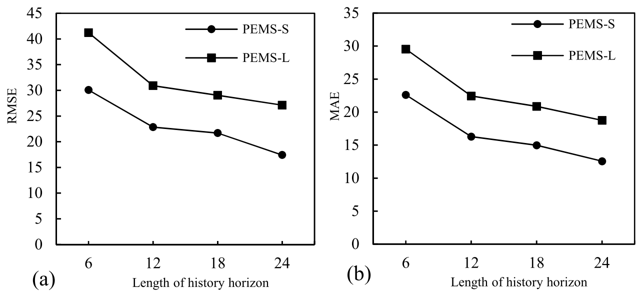

Fig. 6. Results of different lengths of history horizon on PEMS. (a) RMSE of the prediction results. (b) MAE of the prediction results.

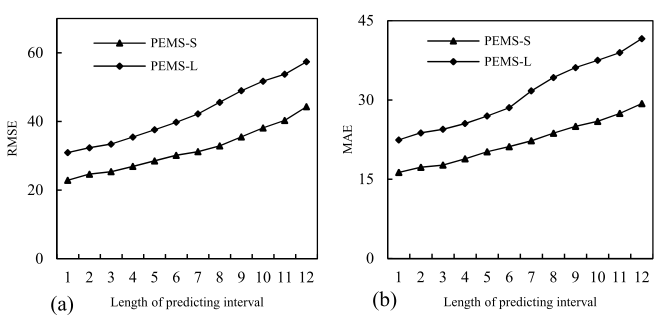

Fig. 7. Results of different lengths of predicting interval on PEMS. (a) RMSE of the prediction results. (b) MAE of the prediction results.

Third, the length of history horizon is an important parameter of the model. If we set the length of history horizon too short, it would be difficult to learn the correlations of the traffic data. If we set the length of history horizon too long, the computation complexity would increase. To evaluate the impact of the length of history horizon, we select the length of history horizon within {6, 12, 18, 24} time slots, and predict the traffic data of the next time slot. As shown in Fig. 6, RMSE and MAE decrease when the length of history horizon increases on both the two datasets. The performance becomes relatively stable when the length of history horizon is larger than 12 time slots. Considering the cost of computation, we set the length of history horizon to 12 time slots as a trade-off.

Predicting interval is also an important parameter of the model. To investigate the predicting ability of the proposed method, we set the length of history horizon to 12 and predict the traffic data in future. As shown in Fig. 7, as the prediction interval increases from 1 to 12, the prediction errors also increase. Both RMSE and MAE increase significantly when interval becomes larger than 4 (corresponding to 20 minutes). The result indicates that the proposed method is more suitable for short-term prediction.

C. The Comparison of Baselines

To evaluate the effectiveness of the proposed method, we compare AARGNN with the following baselines:

1) HA (Historical Average) [45]: HA uses the average of the traffic data from previous periods as the prediction, which mainly focuses on the temporal correlations of the traffic data.
2) ST-ARIMA[9]: ST-ARIMA is an extension of ARIMA, which processes with a spatiotemporal matrix to include extra spatial and temporal correlations. ARIMA regards the traffic data as a time series and fits a parametric model to predict the future traffic. ST-ARIMA utilizes the driving distance between two sensors to construct the spatiotemporal matrix, which can model the spatial correlations of the traffic data.
3) SVR [9]: SVR formulates the nonlinear relationship between input data and output data with a linear function in high dimensional feature space.
4) ANN [13]: ANN predicts the traffic data with the feed forward neural network with back propagation algorithm.
5) FC-LSTM (Fully Connected LSTM) [46]: FC-LSTM regards the input of the traffic data as a long vector and predicts the future traffic with LSTM, which can model the temporal correlations of the traffic data.
6) DCRNN (Diffusion Convolutional Recurrent Neural Network) [19]: DCRNN is one of the state-of-the-art traffic prediction methods based on GNN. DCRNN models the traffic flow as a diffusion process on a directed graph, which incorporates both spatial and temporal correlations of the traffic data. In addition, we extend the DCRNN by adjusting the adjacency matrix to encode various factors as AARGNN and the result model is DCRNN+.
7) TGC-LSTM (traffic graph convolutional LSTM) [28]: TGC-LSTM is one of the state-of-the-art traffic prediction methods based on GNN. TGC-LSTM proposes two regularization terms, including an L1-norm on the graph convolution weights and an L2-norm on the traffic graph convolution features. We use the average speed of related sensors to estimate the free flow speed required by TGC-LSTM. To compare with AARGNN, we also adjust the adjacency matrix.
8) GMAN (Graph Multi-Attention Network) [25]: GMAN is one of the state-of-the-art traffic prediction methods based on GNN. GMAN models the correlations of the traffic data with spatial and temporal attention mechanisms. GMAN focuses on the spatial and temporal attention mechanisms and does not conduct convolution to predict traffic data. In addition, we also extend the GMAN by adjusting the adjacency matrix and the spatiotemporal embedding to encode various factors and the result model is GMAN+.

Among these methods, HA, SVR, ANN, and FC-LSTM simply model temporal correlations. ST-ARIMA, DCRNN, TGC-LSTM, and GMAN model both temporal and spatial correlations. DCRNN, TGC-LSTM, and GMAN represent the traffic data with graphs. Compared to AARGNN, DCRNN

TABLE I
COMPARISON OF DIFFERENT TRAFFIC FLOW PREDICTION METHODS

	PEMS-S			PEMS-L			HZJTD		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
HA	38.07	27.97	14.53%	47.51	34.96	20.33%	6.41	6.10	19.20%
ST-ARIMA	36.69	25.28	14.14%	45.40	32.30	18.44%	8.33	7.91	21.50%
SVR	37.12	28.85	14.73%	52.33	32.44	21.34%	6.30	5.97	18.20%
ANN	32.87	23.76	13.10%	38.90	29.70	20.80%	5.81	5.35	15.52%
FC-LSTM	28.66	21.10	11.29%	36.45	28.76	16.72%	5.70	5.12	14.43%
DCRNN	24.57	17.55	9.87%	32.60	23.56	16.51%	5.21	4.62	13.67%
DCRNN+	24.07	17.02	9.60%	31.62	22.86	15.98%	5.18	4.34	13.54%
TGC-LSTM	24.05	17.21	9.95%	33.01	23.46	14.40%	5.30	4.42	13.22%
TGC-LSTM+	23.80	17.12	9.70%	32.30	23.21	14.40%	5.20	4.18	13.07%
GMAN	24.01	16.90	9.96%	32.12	23.13	14.47%	5.02	4.49	13.21%
GMAN+	23.04	16.56	9.52%	31.47	22.65	14.08%	4.81	4.26	13.10%
AARGNN	**22.85**	**16.29**	**8.72%**	**30.91**	**22.45**	**12.75%**	**4.70**	**3.80**	**12.60%**

bases the diffusion process on fixed adjacency matrix and does not update the adjacency matrix. TGC-LSTM updates the edge attributes by multiplying adjacency matrix with a free-flow reachable matrix. The free-flow reachable matrix simply measures the connection of two sensors with binary value, which does not distinguish the contributions of different factors. GMAN models the correlations with attention mechanisms and does not construct edge attributes and global attributes for graph. Table I shows the performances of AARGNN and the baselines. We set the length of history horizon to 12 and predicting interval to 1. The predicting target data of PEMS are traffic flow and the predicting target data of HZJTD are speed. From the results, we can see:

1) ST-ARIMA performs better than HA. As an extension of ARIMA, ST-ARIMA considers the spatial and temporal correlations, while HA just considers the history average value of the traffic data. It indicates that considering the spatial correlations of the traffic data can improve the prediction performance.
2) ANN performs better than other traditional methods. The result indicates that ANN can model the non-linear correlations of traffic data. ANN performs worse than other neural network models. The underlying reason may be that the shallow network of ANN cannot fully model the correlations.
3) Deep learning methods, including FC-LSTM, DCRNN, DCRNN+, TGC-LSTM, TGC-LSTM+, GMAN, GMAN+, and AARGNN, perform better than traditional methods. The result indicates that deep learning methods can model the non-linear correlations of the traffic data better than traditional methods.
4) GNN-based methods, e.g., DCRNN, DCRNN+, TGC-LSTM, TGC-LSTM+, GMAN, GMAN+, and AARGNN, perform better than FC-LSTM. GNN-based methods construct graphs to model the traffic data, while FC-LSTM ignores the correlations of the traffic data from different sensors. The result indicates that considering the correlations of the traffic data from different sensors and modeling with GNN can improve the prediction performance.
5) AARGNN, DCRNN+, TGC-LSTM+, and GMAN+, perform better than DCRNN, TGC-LSTM, and GMAN in most situation, which indicates that encoding multiple factors can improve the prediction performance.
6) AARGNN performs better than DCRNN+, TGC-LSTM+, and GMAN+, which furtherly verifies the effectiveness of constructing attributed graph and encoding both static and dynamic factors into the attributes to model the temporal, spatial, and semantic correlations of traffic data.
7) All methods perform better on the dataset PEMS-S than on the dataset PEMS-L. The underlying reason may be that PEMS-L was mainly collected from freeways with less correlation with extra urban data.

D. Ablation Study

To futherly investigate the contributions of different components, we compare AARGNN with its variants as follows:

1) AARGNN-nn: It only uses traffic flow as the attributes of the nodes and ignores other node-level data.
2) AARGNN-ne: It ignores the edge attributes in the prediction process.
3) AARGNN-ng: It ignores the global attributes in the prediction process.
4) AARGNN-na: It ignores the different contributions of various factors (i.e., removing the attention machinism).

The results on PEMS datasets are shown in Table II, from which we can see:

1) AARGNN perfoms better than AARGNN-nn, AARGNN-ne, and AARGNN-ng, which verifies the effectiveness of all the attributes in modeling the correlations of the traffic data.
2) The impacts of the attributes can be ranked as: edge attributes > node atttibutes > global attributes. The reason may be that the edge attributes directly related to the correlations of the traffic data from different sensors.

TABLE II
EVALUATION OF DIFFERENT COMPONENTS ON PEMS

	PEMS-S			PEMS-L		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
AARGNN-nn	23.36	16.54	9.79%	32.30	23.76	15.24%
AARGNN-ne	28.40	20.24	11.08%	34.62	28.56	16.45%
AARGNN-ng	23.35	16.44	9.52%	31.64	23.26	14.82%
AARGNN-na	23.97	16.95	9.67%	31.36	22.56	14.85%
AARGNN	**22.85**	**16.29**	**8.72%**	**30.91**	**22.45**	**12.75%**

TABLE III
EVALUATION OF DIFFERENT COMPONENTS ON HZJTD

	HZJTD		
	RMSE	MAE	MAPE
AARGNN-ne	5.41	4.89	14.05%
AARGNN-ng	4.85	4.36	13.20%
AARGNN-na	4.73	3.86	12.70%
AARGNN	4.70	3.80	12.60%

In addition, the calendar features utilized in the global attributes are mainly related to temporal correlations, which can also be learned from the traffic data utilized in the node attributes.

3) AARGNN performs better than AARGNN-na, which verifies the effectiveness of specifying the contributions of different factors.

The results on HZJTD are shown in Table III. Note that the the incident data and weather data related to the HZJTD dataset are unavailable. So we construct the node attributes with speed data, construct the edge attributes with POI data and geographic data, and construct the global attributes with calendar features. Thus we compare AARGNN with AARGNN-na, AARGNN-ne, and AARGNN-ng. It can be seen that:

1) AARGNN-na performs similar to AARGNN. The underly reason may be that the attention machism here is only applied to edge attributes and does not contribute a lot to the results.
2) AARGNN-ne performs worst among the variants. The result indicates that the impacts of the edge attributes are greater than that of the global attributes, which is consistent with the results on PEMS.
3) AARGNN performs better than AARGNN-ng and AARGNN-ne, which verifies the effectiveness of constructing the graph with edge attributes and global attributes.

To furtherly investigate the effect of the attention mechanism, we compare AARGNN with its variants by removing the attention component on different attributes:

1) AARGNN-w/o-na: it removes the attention component on node attributes.
2) AARGNN-w/o-ea: it removes the attention component on edge attributes.
3) AARGNN-w/o-ga: it removes the attention component on global attributes.

Note that the incident data and weather data related to the HZJTD dataset are unavailable, we only conduct the experiment on the PEMS dataset. The results are shown in Table IV, from which we can see:

TABLE IV
EVALUATION OF DIFFERENT ATTENTION COMPONENTS ON PEMS

	PEMS-S			PEMS-L		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
AARGNN-w/o-na	22.96	16.44	9.09%	31.03	22.65	14.23%
AARGNN-w/o-ea	25.56	19.22	10.08%	33.82	25.70	15.22%
AARGNN-w/o-ga	23.44	17.23	9.60%	32.57	22.87	14.24%
AARGNN	**22.85**	**16.29**	**8.72%**	**30.91**	**22.45**	**12.75%**

1) AARGNN performs better than AARGNN-w/o-na, AARGNN-w/o-ea, and AARGNN-w/o-ga, which verifies the effectiveness of attention components.
2) AARGNN performs better than AARGNN-w/o-na, but the results are similar. The underlying reason may be that the contributions of factors in node attributes are similar.
3) AARGNN-w/o-ea performs worst among the variants. The results indicate that the attention mechanism plays a more important role on edge attributes than the on node attributes and global attributes. The underlying reason may lie in two aspects: the complexity of the factors in edge attributes and the different contributions of each factor in edge attributes.

V. CONCLUSION AND FUTURE WORK

In this paper, we investigate the problem of traffic flow prediction and propose a GNN-based traffic flow prediction method, AARGNN, which takes both static factors and dynamic factors (e.g., spatial distance, semantic distance, road characteristic, road situation, and global context) into consideration when modeling the correlations of the traffic data. We encode these factors into the attributes of the graph and combine GNN with LSTM to jointly model the temporal, spatial, and semantic correlations of the traffic data. In addition, we utilize the attention mechanism to quantify the contributions of each factor. Through a series of experiments based on real-world traffic datasets, we demonstrate that AARGNN can perform better than the compared baselines.

Our work could be extended in the following directions. First, the attributes of the edges contain dynamic and static parts, which are concatenated with the attention mechanism. We will investigate and improve the attention machinism to improve the effectiveness of capture the relationship of multiple factors. Second, node, edge, and global attributes could be extended to construct hierarchical graphs to avoid oversmoothing and model the spatial and semantic correlations of the traffic data from distant nodes in the graph. Third, POIs of different levels may have different degrees of influences on traffic conditions, which is not distinguished in the proposed method. We will investigate the influences by incorporating extra location based social network datasets.

REFERENCES

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, no. 1, pp. 3–19, Jun. 2014.

[2] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.

[3] O. Iwao and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, Feb. 1984.

[4] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, 1996.

[5] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, 1995.

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

[7] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[8] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.

[9] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, 2011.

[10] W.-C. Hong, Y. Dong, F. Zheng, and C.-Y. Lai, "Forecasting urban traffic flow by SVR with continuous ACO," *Appl. Math. Model.*, vol. 35, no. 3, pp. 1282–1291, 2011.

[11] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, 2009.

[12] V. Lelitha and L. R. Rilett, "A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 194–199.

[13] K. Kumar, M. Parida, and V. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Proc. Social Behav. Sci.*, vol. 104, no. 2, pp. 755–764, Dec. 2013.

[14] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, pp. 1501–1517, 2017.

[15] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, pp. 818–835, Apr. 2017.

[16] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 166–180, May 2018.

[17] S. Wang, H. Miao, J. Li, and J. Cao, "Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 10, 2021, doi: 10.1109/TITS.2021.3055207.

[18] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 1–7.

[19] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Represent.*, Apr. 2018, pp. 1–16.

[20] C. Chen *et al.*, "Gated residual recurrent graph neural networks for traffic prediction," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Jan. 2019, pp. 485–492.

[21] S. Guo *et al.*, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Jan. 2019, pp. 922–929.

[22] Y. Shin and Y. Yoon, "Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2082–2092, Mar. 2022.

[23] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proc. 34nd AAAI Conf. Artif. Intell.*, Feb. 2020, pp. 3529–3536.

[24] Z. Diao *et al.*, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Jan. 2019, pp. 890–897.

[25] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. 34th AAAI Conf. Artif. Intell.*, Feb. 2020, pp. 1234–1241.

[26] K. Guo *et al.*, "Optimized graph convolution recurrent neural network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 1138–1149, Feb. 2021.

[27] L. Zhao, Y. Song, C. Zhang, and Y. Liu, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[28] 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. 2019.

[29] X. Geng, X. Wu, L. Zhang, Q. Yang, Y. Liu, and J. Ye, "Multi-modal graph interaction for multi-graph convolution network in urban spatiotemporal forecasting," 2019, *arXiv:1905.11395*.

[30] 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.

[31] Z. Pan *et al.*, "Spatio-temporal meta learning for urban traffic prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1462–1476, Mar. 2022.

[32] H. Yang, C. Liu, M. Zhu, X. Ban, and Y. Wang, "How fast you will drive? Predicting speed of customized paths by deep neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2045–2055, Mar. 2022.

[33] Z. Cui, R. Ke, Z. Pu, X. Ma, and Y. Wang, "Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102620.

[34] X. Chen *et al.*, "TSSRGCN: Temporal spectral spatial retrieval graph convolutional network for traffic flow forecasting," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 954–959.

[35] M. Lv, Z. Hong, L. Chen, T. Chen, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3337–3348, Jun. 2020.

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

[37] G. Xu *et al.*, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Jan. 2019, pp. 3656–3663.

[38] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu, "Mining road network correlation for traffic estimation via compressive sensing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1880–1893, Jul. 2016.

[39] W. Shao and L. Chen, "License plate recognition data-based traffic volume estimation using collaborative tensor decomposition," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3439–3448, Nov. 2018.

[40] S. Tang and H. Gao, "Traffic-incident detection-algorithm based on nonparametric regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 38–42, Mar. 2005.

[41] M. Lv, L. Chen, X. Wu, and G. Chen, "A road congestion detection system using undedicated mobile phones," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3060–3072, Dec. 2015.

[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, Dec. 2016, pp. 1–9.

[43] G. Justin, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 1263–1272.

[44] X. Wang, G. Ross, A. Gupta, and K. He, "Non-local neural networks," in *Proc. 34th Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[45] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," *J. Highway Transp. Res. Develop.*, vol. 13, no. 3, pp. 82–85, 2004.

[46] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

Ling Chen received the B.S. and Ph.D. degrees in computer science from Zhejiang University, China, in 1999 and 2004, respectively. He is currently a Professor with the College of Computer Science and Technology, Zhejiang University. His research interests include ubiquitous computing and data mining.

Wei Shao received the B.S. degree in software engineering from Zhejiang University, China, in 2012, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. His research interests include urban computing and intelligent transportation systems.

Mingqi Lv received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2012. He is currently an Associate Professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. His research interests include spatiotemporal data mining and ubiquitous computing.

Weiqi Chen received the B.Eng. degree in computer science and technology from Xi'an Jiaotong University, China, in 2017. He is currently pursuing the M.S. degree with the College of Computer Science and Technology, Zhejiang University, China. His research interests include urban computing and time series modeling.

Youdong Zhang received the M.A. degree in computer science from the Huazhong University of Science and Technology, China, in 2012. He is currently a Staff Engineer at Alibaba Group. His research interest includes distributed storage and database.

Chenghu Yang received the B.S. degree in computer science from Northeast Forestry University, China, in 2009. He is currently a Senior Staff Engineer and leads the NoSQL-Database Team, Alibaba Group. His research interest includes distributed storage and database.