IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 October 2023, accepted 12 November 2023, date of publication 16 November 2023, date of current version 27 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3333938

RESEARCH ARTICLE

A Novel Graph Convolutional Gated Recurrent Unit Framework for Network-Based Traffic Prediction

BASHARAT HUSSAIN[1,2], MUHAMMAD KHALIL AFZAL[1], (Senior Member, IEEE), SHERAZ ANJUM[1], IMRAN RAO[3], AND BYUNG-SEO KIM[4], (Senior Member, IEEE)

[1]Depertment of Computer Science, COMSATS University Islamabad, Wah Campus, Wah Cantt 47040, Pakistan
[2]Faculty of Computing, Riphah International University, Islamabad 44000, Pakistan
[3]Blue Brackets Technologies Pvt. Ltd., Islamabad 44000, Pakistan
[4]Department of Software and Communications Engineering, Hongik University, Sejong 30016, Republic of Korea

Corresponding authors: Muhammad Khalil Afzal (khalilafzal@ciitwah.edu.pk) and Byung-Seo Kim (jsnbs@hongik.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea Government [Ministry of Science and Technology in Information and Communication Technology (MSIT)] under Grant 2022R1A2C1003549.

ABSTRACT A *Smart City* is characterized mainly as an efficient, technologically advanced, green, and socially informed city. An intelligent transportation system (ITS) is a subset area of smart cities that enhances the safety and mobility of road vehicles. It essentially makes travel more convenient, time-efficient and improves the citizens' quality of life. Accurate and real-time traffic prediction enables law enforcement agencies with well-informed about traffic congestion. However, accurate traffic prediction has been considered a challenging issue. Traffic prediction has restrictions on road network topology and the patterns of dynamic change in time-series data. We propose a novel deep learning framework GCST-GRU, called graph convolutional Spatio-temporal gated recurrent unit, to determine the next traffic state from traffic data. The proposed model learns complex topological structures by capturing a) spatial dependencies from data by using the graph convolution operator, and b) temporal dependencies by using the GRU neural network. Experimental results demonstrate that our framework can obtain complex Spatio-temporal correlations efficiently from the traffic network and perform better than state-of-the-art baseline models on a real-world traffic dataset. The graphical visualization by using convolution operation over the neural network shows that the model outperforms the reachability of the 3-hops neighbor effect in the traffic data graph. Additionally, the training time of the proposed framework is better than the existing state-of-the-art deep learning studies.

INDEX TERMS Intelligent transportation system, traffic flow prediction, traffic network graph, gated recurrent unit, hyperparameters optimization, deep learning.

I. INTRODUCTION

A *Smart City* is an urban zone that utilizes the latest technologies and sensors to collect data. Such data is incredibly important, collected from various sources and business sectors, involving citizens, devices, infrastructures, and locations. The basic aim of a smart city is to optimally utilize data, technology, and services to enhance operational efficiency and provide well-informed decisions to its citizens. Adopting the latest technology to address wider issues of residents improves their quality of life. Intelligent transportation system is regarded as a significant part of smart cities. A challenging and essential task of an ITS is to determine the accurate traffic prediction in advance. The objective is to estimate traffic state measures (a few steps ahead) provided we are given the on-ground roadway network characteristics and a sequence of historical transportation data. In seeking to improve traffic congestion and safety on roads and highways, there has been an increased interest in ITS [1]. In recent years, the variety and volume of transportation data have

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

increased, leading us to the big-data notion. The data-driven traffic prediction methods are becoming popular due to their strength to outperform classical and simulation-based methods [2]. However, in transportation data by virtue of having stochastic characteristics, the challenge lies in traditional methods to achieve traffic prediction accuracy and efficiency in advance. Major techniques for traffic flow prediction (TFP) proposed by the researchers are categorized into two classes: classic statistical, and machine learning approaches [3], [4], [5], [6], [7], [8].

Classic statistical approaches are inaccurate for unpredictable, random, and nonlinear features present within the transport data. A statistical algorithm is computationally faster, but makes stronger assumptions about the data; the method will perform better in case the supposition turns out to be right or else performs severely. Classical models like regression, moving average, and classification do not take into account the time-series-based data variations and hence are not suitable for traffic predictions. The auto-regressive integrated moving-average (ARIMA) model [4] is a widely used classical estimation approach. Other approaches that are discussed and utilized by the researchers are maximum-likelihood estimate (MLE) and Kalman filtering (KM). However, as already discussed, each classical or statistical method is limited due to the stationary assumption of time sequences and fails to consider the Spatio-temporal correlations, which is a significant characteristic of traffic data. Moreover, only temporal information is mostly grasped and spatial dependency is ignored or barely considered in traffic data analysis [5]. Another example is a linear regression that performs accurately for short intervals ahead but is less effective for relatively longer predictions. Traffic prediction problem is generally classified into two scales: short-term (that includes a few minutes ahead to an hour), and long-term (spans over several hours/days) [6].

Machine learning methods are widely applied to traffic prediction problems. They are better capable of capturing more complicated non-linear correlations discussed in the next section. In recent years, an improvement has been noted in two directions: first, the acquisition of massive Internet of Things (IoT) data through sensors installed across the freeways, and second, the improvement of computational power to learn similar trends using deep neural networks (DNN). Traffic flow data patterns are complex and repeat in day intervals. The researchers applied temporal networks to solve the problem and predict the traffic states ahead, however, they lack utilizing spatial dependency. Recently some researchers applied convolution approaches to induce spatial dependencies using convolution neural networks (CNN) [5]. Graph convolutional network (GCN) [10], on the other hand, has the capability to apply convolution operation to traffic graph data, which is suitable to represent the traffic network structure due to its tendency to recognize spatial relationships.

RNN has the potential to memorize historical data and utilize temporal features to perform traffic flow prediction. However, RNN is susceptible to the renowned vanishing gradient problem. Such a problem loses the capability to recognize the lengthy sequences in the temporal domain. To handle the vanishing gradient issue, some authors proposed variants of RNN. Among the two broadly used versions are the gated recurrent unit (GRU) and the long short-term memory (LSTM) [7] models. GRU is a newer version compared to LSTM, proposed in 2014. Cho et al. [8] to solve short-term memory issues by proposing the GRU-gated model at earlier stages. We attempt to utilize graph convolution network [10] representation of road highway to exploit Spatio-temporal correlation with RNN variants and hypothesize to obtain a superior model for traffic flow prediction. In this work, we study a graph convolution operator applied on the gated recurrent unit neural network, an attempt to modify the input structure of a standard network cell. We propose a *novel deep learning framework* for traffic prediction using a Spatio-temporal graph convolutional method with GRU neural network to improve traffic prediction accuracy and stability over the traditional deep networks. Therefore, we alter the internal structure of a standard GRU cell to input graph convolution graph data in addition to temporal observations. The major contributions in this paper are described as follows:

1) We combined the recurrent neural network with the graph convolutional neural network to exploit the spatial correlations among the road network segments along with the temporal relations from the traffic data. We utilized the concept of free flow reachability as an inspiration to unfold the Graph Convolution Spatio-temporal (GCST) network operations performed on each cell of the gated recurrent unit (GRU). Graph convolution is a motivation for building such a model by characterizing the spatial dependency to build the topological structures from traffic data. RNNs are capable of capturing temporal dependency by describing the dynamic traffic flow variations over time on the road network. We utilized the concept of neighborhood nodes in the road network and explored the optimum value of the k-hops neighborhood.
2) We explored *regularization* in the proposed neural network to improve the loss function. Our motivation for regularization is: a) it provides more stable and interpretable learned-weight values, and b) it confines the features of the graph convolution operator within the defined brackets. For such improvement of the loss function, we added regulation expressions at two levels for better interpretation: firstly, regularize graph convolution network weights for traffic data, and secondly, regularize graph convolution features for traffic data. Applying the concept of regularization is an important contribution to this research work.

The rest of the paper is organized as: Related background and research work are described in Section II; the explanation of the proposed framework is discussed in Section III.

In section IV, we evaluate the predictive performance of the GCST-GRU by using a real-world traffic dataset including the design of the model parameters and experimental results are presented. Lastly, in Section V, we mentioned the conclusion and future directions for this research work.

II. RELATED BACKGROUND

In the last decades, intelligent traffic system has gained greater attention due to their significance in *Smart Cities*. Many researchers led to building competitive and novel models related to traffic flow prediction. Several studies reviewed in the literature have established traffic prediction frameworks for estimating the traffic flow including possible traffic congestion in advance. During recent years, significant traffic prediction performance has been observed with the rise of deep learning (DL) AI methods. In traffic prediction, DL models are used to recognize complex patterns from huge data, including both temporal and spatial dependencies. The results obtained with these models are remarkable. The idea is to learn a hierarchical model to map the original input directly to the expected output. In general, deep learning models utilize several neural network layers to form a deep non-linear architecture, and the entire network is trained end-to-end to predict large-scale and complex Spatio-temporal data. Authors in [11] have proposed a method to predict the generic patterns and characteristics of traffic flow, called Stacked auto-encoder. Later on, transportation data prediction was described as an image processing problem, and the conventional neural network (CNN) is utilized for the large-scale transportation data analysis [5]. CNN is also employed to extract the spatial correlation of the grid-structured data described by images or videos. However, vanilla CNN is capable of extracting spatial correlations from the image data representing traffic network graphs in 2D Euclidean space (as two-dimensional matrices). Therefore, extracting the spatial features from images, when trained using the CNN model does not produce optimal results, when representing images for traffic-network structure as graph [12], [13]. For temporal dependency in time-series, RNN and its two variants LSTM [7] and GRU [8] are widely utilized and can remember longer matching patterns within the data. GRU is used for traffic flow prediction in complex scenarios, if trained carefully with the right parameters/ hyperparameters can lead to significantly accurate results [14]. Graph convolutional network (GCN) [10], on the other hand, extends convolution operation to more general graph-structured data, which is more suitable to represent the traffic network structure due to its tendency to recognize spatial relationships.

Scientists further monitored RNNs and their variations as good candidates for capturing the nonlinear patterns from traffic series with improved accuracy. RNNs use a memory-based learning approach that can remember the patterns in time series-based data sequences with longer temporal dependencies and features. However, there is a problem with RNN that it is susceptible to short-term memory problem and become unsuitable for predicting longer sequences. There are two popular gated variations of RNN named LSTM and GRU. Both models have the superior capability for time series prediction with longer temporal dependency and temporal features learning capability. In 2014, Cho et al. [8] proposed a novel neural network that is capable of solving the vanishing gradient issue when compared with standard RNN. Eventually, researchers successfully utilized both GRU and LSTM for short-term traffic flow prediction and results showed that the models are superior to traditional models, for example, stacked auto-encoders (SAE) [15], and feed-forward neural network (FFNN) [11].

Some authors appended external factors to the original data and presented significant improvement in traffic flow predictions. It is observed that the transportation data possesses repeatable characteristics over time amid the day/night intervals, however, adding environmental characteristics along with the input traffic data improves the performance. For example, weather conditions, changing populations, events happening, and accidents may increase or decrease traffic. In [16], authors applied the same approach by mixing the input sequences with weather conditions to increase the prediction accuracy. GRU has a simpler method than LSTM and has significant applicability to solve traffic prediction problems.

In [17], the work addresses the graph relationship by utilizing a road network adjacency matrix where the neighbor road segments were treated as adjacent nodes. The authors predict the speed of road segments by a technique called T-GCN, a framework that combines vanilla CNN and graph neural network with RNN layers to exploit seasonal patterns in time-series data. They compare the result with the various models including the GCN model with spatial adjacency matrix and without temporal layer, the original GRU NN, and their proposed model. The T-GCN results were superior.

CNN is also applied for traffic prediction, however, by nature, the traditional version of CNN cannot recognize the physical and topological patterns from traffic network data. To overcome these issues, authors in [2], [10], and [24] proposed graph for the representation of traffic road network and applied graph-based convolution operation for obtaining significant features from graph-based transportation network.

In the domain of intelligent traffic flow prediction, some researchers combined GCN with RNN and its variants [2], [10], [25]. They proposed novel models composed of an integrated architecture wherein a GCN is fed to the RNN cells to take care of complex dependencies. These GCN-based models performed well when compared with the traditional DL methods, they converged faster due to the lesser number of weight parameters required. Traffic networks are composed of the complexity of time-based dynamic and location-based spatial dependencies. The authors in [10] proposed a traffic graph convolutional recurrent neural network model called GC-LSTM, a model that combines LSTM with GCN to capture complex dependencies. The basic motivation of this work is to learn graph representation using nodes, road segments, and sub-graphs and predict future traffic states. Authors in [17] proposed a new approach SGC+LSTM that builds a model on LSTM by stacking 1D-layer spectral graph convolution operation. STGCN [2] constructs ST-Conv blocks with spatial and temporal convolution layers and applies residual connection with bottleneck strategies.

We proposed a novel framework that presents an integrated architecture by combining GCN and GRU to find a correlation of complex dependencies from traffic data. Spatio-temporal graphs are recently used in other similar areas. The work [26] utilizes Spatio-temporal recurrent neural network (STRNN) that consists of three sub-parts: spatial, temporal, and residual to work with body motion aggregate detection. The model only considers the body's physical constraints without exploiting the movement coordination and relation. Authors in [27] proposed a model that utilizes Spatio-temporal dependency along with external factors of the crowd flows. It further classifies temporal features into recent, daily, and weekly components. the model helps traffic flow data to dynamically learn the Spatio-temporal features.

In the literature, traffic prediction is considered a connected network of road segments, the traffic states of a node is influenced by the traffic states of its neighboring nodes. Consequently, researchers utilized both temporal and spatial features to predict future traffic states. Shi et al proposed a novel Attention-based Periodic-Temporal neural Network (APTN), which is an end-to-end solution for traffic forecasting that captures spatial, short-term, and long-term periodical dependencies [30]. However, region-level traffic state prediction still faces challenges: The topological structure of the city's complex road network is destroyed when regions are divided into squares or hexagons, and it is difficult to extract accurate spatial features of these regions; irregular regions based on natural roads, administrative divisions, and other factors are typically non-Euclidean distance data. The classic CNN model is difficult to apply to this type of data, however, GCN is a better choice.

A framework for recognizing spatial domains using multi-view graph convolution networks (STMGCN) is proposed by the authors in [31] and [32]. In the graph-based traffic forecast, they aimed to exploit spatial dependencies in GCN. Their approach generates numerous neighbor graphs, known as views, with varying similarity measures based on spatial coordinates. Second, by combining expressions with each neighbor graph via GCNs, it learns various view-specific embeddings. Third, an attention method is utilized to adaptively fuse view-specific embeddings (hidden state) to capture the relevance of diverse graphs and therefore derive the final spot embedding. STMGCN makes excellent use of spatial context to boost the expressive capability of latent embeddings with multiple graph convolutions.

With the rise of Graph Convolutional Networks (GCN), we can capture exact spatial features from irregular regions [10]. GCN takes input as a graph to learn and build a trained model. Recently, researchers tried to merge the GCN and time series model to forecast traffic state. As explained previously, Zhao et al proposed a temporal graph convolutional network (T-GCN) model by combining GCN and GRU for traffic prediction [17]. Researchers in [33] proposed a novel graph-based neural network that expanded the existing GCN to predict road traffic speeds. This study is focused on devising a GCN model that mimics true propagation patterns of traffic. Most studies based on GCN and NN consider the adjacent regions and free flow reachability to improve the prediction accuracy. In this context, we propose a new deep learning method that can capture complex temporal and spatial features from traffic data in this research that uses regularization to improve the results that can be used for traffic state forecasting. Our findings from the related literature can be concluded as follows:

1) GRU is being used in the literature for traffic prediction [8]. Combining GCN and GRU would be a motivation for improving accuracy, although with a slight improvement margin. We explore Graph Convolution operation over the traffic network represented as a graph, to extract significant features from the graph-based traffic network, that can determine future network states with superior accuracy than baseline models.
2) Application of regularization to improve the loss function is a significant area to explore. regularization terms added at two levels make the weight matrices more sparse and therefore enhance the interpretability of the proposed model. We explore the L1-norm on graph convolution weights and the L2-norm on graph convolution features when combined with the model's loss function to explore the improvement.

III. METHODOLOGY

In this section, we describe preliminary notations and then the graph convolution prediction framework using the proposed model.

A. PRELIMINARY NOTATIONS

1) GRAPH-BASED TRAFFIC DATA REPRESENTATION

Traffic data is characterized by road segments (links) and sensors (nodes) installed on the roadside to collect Internet of things (IoT) data at regular intervals. By definition, graphs can be effectively used to model such data having temporal and spatial dependencies. Graphs are used to solve various problems like article citation, social networks, protein sequencing, and molecule graphs. Graph-based traffic data representation is formally defined as: given an input sequence $X_t = \{x_1, x_2, \ldots, x_N\}$ as original traffic state at time t, where $x_i \epsilon R^N$. Assume the traffic network of roads builds a graph G which is undirected and represented with the expression $G = (V, E)$ having N number of nodes. For the nodes $v_i, v_j \epsilon V$ and there are edges $(v_i, v_j) \epsilon E$. In a traffic network structure, sensors are the **nodes** and road segments are the **edges** connecting those traffic sensing locations.

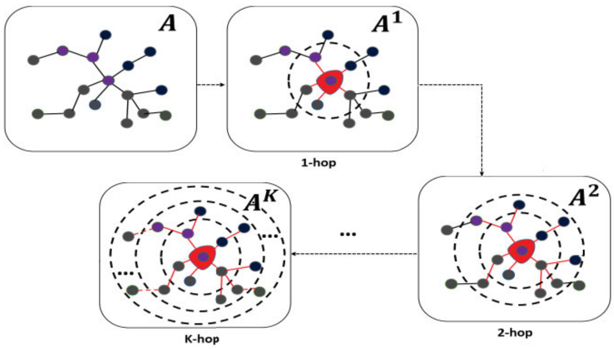

FIGURE 1. K-hop relationship graph based on adjacency matrix.

2) ADJACENCY MATRIX FOR TRAFFIC DATA REPRESENTATION

Traffic data can be represented by an adjacency or neighborhood matrix, which refers to the connectedness of the graph nodes. Assume $A \epsilon R^{N\times N}$ be an adjacency matrix, defining the neighborhood of each element $A_{i,j} = 1$ when there exists a road segment between nodes i and j and otherwise $A_{i,j} = 0$, is used to represent the neighboring nodes in G. An adjacency matrix is a symmetric matrix over an undirected graph, defined as

$$A_{ij} = \begin{cases} 1, & \text{if } (i,j)\epsilon E, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We further compute the degree matrix of the graph using the information from the adjacency matrix. We define a degree matrix where each node counts the number of links reachable to each node, represented by $Deg \epsilon R^{N\times N}$ in which each element is computed as $Deg_{ii} = \sum_j A_{ij}$. By virtue of being a diagonal matrix, Deg has non-zero element s at diagonal nodes, and zero values at non-diagonal nodes.

3) K-HOP NEIGHBORHOOD MATRIX FOR TRAFFIC DATA

Let i and j be two nodes in the graph G. We define a link-count function $d(v_i, v_j)$ by counting the minimum number of links traversed from node i to node j, based on the adjacency matrix. Assume NB represents a set $k-th$ order neighborhood (also called k-hop) for each node i is expressed as

$$NB = \{v_j \epsilon V | d(v_i, v_j) \leq k\} \tag{2}$$

Please note we include the node i itself in the counting process. Since the traffic data is time series in nature taken at regular intervals, the next states can be determined by predicting the influence on existing states. We define the neighborhood of each node i is composed of i itself and a 1-hop adjacency matrix to describe the neighborhood relationship in the graph G, the mathematical expression is as follows

$$\widetilde{A} = A + I_N. \tag{3}$$

where I_N represents an identity matrix of order N. The identity matrix I_N added to A ensures that the nodes are self-accessible in graph G. Fig. 1. shows the one, two, and three-hop sub-graphs concept starting from the red node. The neighborhood correlation of the nodes in the graph at k-hops can be expressed as $(A + I_N)^k$. However, by virtue of counting hops from the origin, some values in $(A + I_N)^k$ matrix will definitely cross over to the value one. The notation used for describing the k-hop neighborhood, for a given node, depicts that there exist k-hop neighbors in the network. However, it's workable to mark node weights by counting the number of hops for representing k-hop neighbors of that node. Therefore, a k-hop neighborhood matrix $\widetilde{A}^k$ is defined for each node, where we clip the elements of this matrix from another matrix $(A + I_N)^k$, and each of its elements is set as 0 or 1. Considering the matrix $\widetilde{A}^k$, every element $\widetilde{A}_{i,j}$ adheres to expression as:

$$\widetilde{A}^k_{i,j} = min((A + I_N)^k_{i,j}, I_N). \tag{4}$$

where min refers to the minimum value function. We mathematically define the k-th order adjacency matrix as:

$$\widetilde{A}^k = clip(\prod_{i=1}^{k}(A^i + I)) = clip(A^k + I). \tag{5}$$

We can note, $\widetilde{A}^1 = A^1 = A$. The symbol $\prod_{i=1}^{k}$ defines the product of the terms till index k. It reveals that the one-hop neighborhood matrix is exactly the adjacency matrix.

4) FREE-FLOW REACHABLE MATRIX FOR TRAFFIC DATA

Considering road segment length in a traffic network connecting the adjacent nodes, we denote $D \epsilon R^{N\times N}$ as a *Distance Adjacency* matrix having every element representing a road segment. Assume $D_{i,j}$ is the real distance measured from node i to j ($D_{i,i} = 0$). We can determine that a traffic network contains roadway segments. Adjacent road segments have inter-dependency to impact the propagation of the traffic characteristics in two manners: 1) traffic congestion, when occurs, slows down the traffic stream; 2) The specific behavior of a driver or vehicle impacts the traveling speed. Therefore, the traffic congestion resolution between two non-adjacent nodes shouldn't ignore the impact on intermediate segments in a graph-based network. We should consider the neighborhood factor while determining the reachability between two graph nodes. Thus, we take advantage of utilizing k-hop neighbor nodes to determine the traffic impact transmission, we determine a free flow reachability matrix as, $FFR \epsilon R^{N\times N}$, determined by expression

$$FFR_{i,j} = \begin{cases} 1, & \text{if } V^{FF}_{i,j} m\Delta t \geq D_{i,j}, \ \forall v_i, v_j \epsilon V \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

where free-flow speed between the two points i and j is represented by $V^{FF}_{i,j}$. It is defined as an average speed, under no congestion conditions, traveled by a proposed vehicle. It helps to compare the speed under various external conditions (such as extreme snowing, rain, and road accident events) posing the adverse effects on speed. The symbol

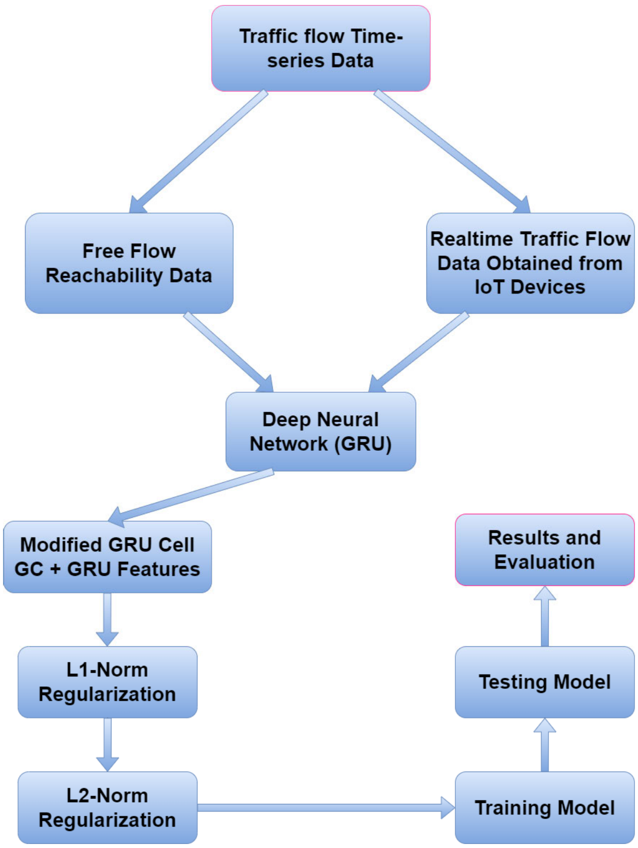

FIGURE 2. An overall flowchart of the proposed GCST-GRU framework.

t represents the interval time duration and m the count of the time intervals where we measure the distance covered provided $V_{i,j}^{FF}$ is applicable. Therefore, m can be considered as a temporal influence used in the representation of FFR. Each element $FFR_{i,j} = 1$ provided a vehicle can reach in m time steps from point i to j. Free-flow speed is presented by the expression $m \times \Delta t$. If vehicles cannot reach from i to j then $FFR_{i,j} = 0$. In short, the $FFR_{i,j}$ denotes a condition where a vehicle is able to travel freely from point i to node j in traffic-network graph G within a specific time interval provided the free-flow speed is maintained. Typically, we imagine FFR as a diagonal matrix where each diagonal value is one depicting the fact that every road is self-reachable by definition. An example of FFR could be presented as red stars (as nodes) are connected by the green lines (road segments). It is described in the right part of Fig. 3. The proposed algorithm (GCST-GRU) designed in this research work is novel and applied using FFR after taking inspiration from the work of [10]. We can imagine unfolding the Graph Convolution Spatio-temporal (GCST) network operations as a part of our proposed framework, at time t, in which $\widetilde{A}^k$ and FFR are represented as red nodes.

B. PROBLEM STATEMENT FOR TRAFFIC PREDICTION

Traffic-flow prediction is defined by foreseeing the future traffic road network states, based on the previously observed traffic measurements from the same road network. Such states may include, but are not limited to, the average traffic flow, congestion occurrence, transportation time, density, or traffic volume. In this research work, the traffic road network is transformed into a graph G. Such traffic graph G consists of N nodes and E edges. The nodes are road-side sensor units for collecting the IoT data while edges are the road-segments linking these sensors. Suppose $x_t \epsilon R^N$ represents measurements obtained by these nodes at time t, the signal is collectively represented as the traffic state.

The short-term traffic-flow prediction issue utilizes a function f to learn from data. In this study, f is a nonlinear mapping function of graph convolution over a GRU neural network. Such a function maps the previously obtained traffic states to the next graph signals in a single or more time-steps ahead. the previous traffic state signals are represented by $X_T = [x_1, \ldots, x_t, \ldots, x_T]$, where T denotes the total time-steps data for which the graph signals are presented. This research work describes f as a function to predict the graphical signals in a subsequent step, i.e., x_{T+1}, by using $GCN + GRU$ method is expressed as $\hat{y}^{T+1}$ in eq. 7 as:

$$\hat{y}^{T+1} = x_{T+1} \tag{7}$$

$$\hat{y}^{T+1} = f([x_1, \ldots, x_t, \ldots, x_T]; G(V, E, \widetilde{A}, FFR)) \tag{8}$$

where, G is a traffic network graph with V nodes and E edges. The nodes V are road-side sensors for collecting the traffic data while edges E are the road-segments linking these sensors. Therefore, our objective is to predict the step value $\hat{y}^{T+1}$. Furthermore, we aim to study the function f to influence transportation with respect to neighborhood nodes (adjacency) in a graphical traffic network G. An overall flow is presented in Fig. 2. We note in Fig. 3, a single GRU cell structure is presented with the proposed GCST model expressed on the left part. we extract the raw input from a dataset and transform it into graph convolution data before feeding it to the GRU network. The right side expands the idea of graph convolution spatio-temporal (GCST) network operation as a component of our framework. The details are more elaborated by displaying a traffic graph convolution step unfolded at time t and observed as red star nodes of $\widetilde{A}^k$ and FFR demonstrated for further calculations.

C. GRAPH CONVOLUTION FRAMEWORK

The idea of neighborhood nodes that influence the query node creates a logical region around it, called the *Receptive Field*. Therefore, the receptive field is a localized region in the input space features that affect the results obtained from the graph convolution operation in the neural network. The results of a convolution layer are the 2-dimensional or 3-dimensional matrices obtained from the input data. Authors in the base article [10] suggest a graph convolution model based on the adjacency matrix, therefore, we take advantage of the One-hop neighborhood matrix obtained by multiplication of three factors as expressed in eq. 9.

$$\widetilde{A}^1 = W * \widetilde{A} * x_t \tag{9}$$

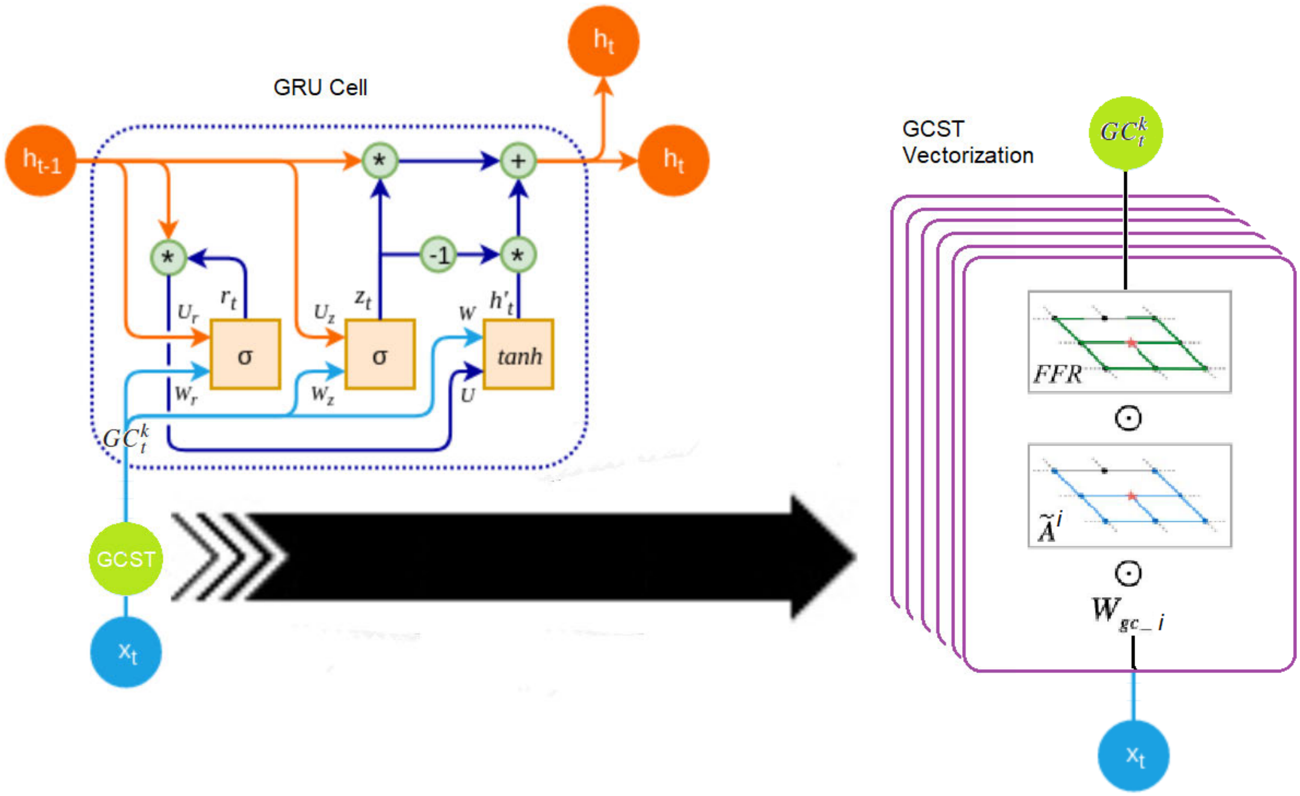

FIGURE 3. **The GRU cell structure is presented with the proposed GCST structure expressed in the left part. The right part of the diagram expands the idea of graph convolution spatio-temporal (GCST) network operation as a component of our framework depicting the traffic graph convolution operation at time t. We demonstrate the structures of $\tilde{A}^k$ and FFR, where the query (red-star) nodes are presented for clarity and utilized in the formula referred in Equation 10.**

where, x_t represents the input data vector, $\tilde{A}$ is the neighborhood matrix, and W is the matrix of the training weights. Consequently, a one-hop neighborhood matrix is defined as the receptive field of graph convolutional operation to extract the localized features of a node as discussed in [28] and [29]. We can extend the concept to consider k-hop neighborhood matrix. Since, in the one-hop neighborhood ($\tilde{A}$) is a single step away from neighboring nodes, therefore, the influential field of graph-convolution operation is limited and confined. This restriction can be resolved by increasing the neighbors to $k > 1$. To obtain an extended receptive field of GCN, we tend to substitute the 1-hop neighborhood matrix $\tilde{A}$ with the k-hop $\tilde{A}^k$. Besides higher-order neighborhood, [10] presented a new concept of applying the free-flow reachability defined in Eq. 6. to solve graph network-wide prediction problems. [10] proposed a traffic graph convolution framework using the LSTM model. It further argues, that the present research studies tend to neglect the edges features in graphs, for example, distances between various road sensor units (let us call it the segment length in G) and the free-flow reachability. Moreover, these studies neglect the higher-order neighbor effect ($k > 1$) in a traffic-network graph G.

Consequently, to confidently define the traffic-network graph convolution problem, we considered both factors: *segment properties* and *high-order neighborhood node characteristics* in G. We define the traffic-network graph convolution problem by utilizing segment properties and high-order neighborhood node characteristics in graph network G. In Equation 10, we define the neighborhood reachability as a function of the average weight matrix, adjacency neighborhood, and free-flow reachability setup for some integer k>1 hop nodes. This is our contribution. We further tested the proposed method on GRU to capture spatial and temporal dependencies on the data to predict short-term traffic states on graph networks, which is the novelty of our work described visually in Fig. 3. Therefore, we are extending the baseline concept proposed in [10], the k-order graph convolution operator for traffic data is defined as

$$GCF_t^k = (W_{gck} \odot \tilde{A}^k \odot FFR) * x_t \tag{10}$$

where the symbol $\odot$ is defined as an element-wise matrix multiplication operator, also named the Hadamard product operator. The $x_t \epsilon R^N$ is the traffic state matrix (or vector) at time t. It means, x_t represents the speed of the traffic flow. x_t can also be described as the state at time t constructed the measure the speed of all the nodes in the traffic network. The $W_{gck} \epsilon R^{N \times N}$ is a weight matrix obtained by training/searching of k^{th}-order traffic network graph convolution. The $GCF_t^k \epsilon R^N$ is an extracted k-order feature of the traffic network. In fact, both $\widetilde{A}^k$ and FFR are sparse matrices that contain either zero or one as element values, the result of element-wise multiplication ($W_{gck} \odot \widetilde{A}^k \odot FFR$) should obtain a sparse matrix as well. In addition, the variable W_{gck}, called trained weight, is capable of measuring the interactive effect between graph nodes, and in this way, improving the interoperability of the proposed strategy.

In Eq. 10, we have a symbol k, a non-negative value integer that represents the degree of the neighborhood. A higher value of k means a relatively higher receptive field exists for the traffic-network graph convolution, therefore, considering the bigger value of k provides an opportunity to extract more neighborhood features from G. However, this leads to the following problems: 1) it demands higher computation time and cost, and 2) the value of k cannot be increased bluntly. When continuously increasing the value of k for a specific traffic-network graph, the expression $\widetilde{A}^k \odot FFR$ tends to converge to FFR. The recent research (e.g., [10]) reveals that when solving transportation prediction problems, we extract the graph convolution features by performing the computation, it's not required for k rises to K_{max}. Furthermore, there is a trade-off between the feature richness (k-hops) and the prediction accuracy of traffic flow. However, the computation cost increases with the feature richness and should be considered to be balanced out with an *optimal* value of k. Considering such an optimality discussion, let $k \leq K_{max}$ denote the optimal value of neighborhood/hops/nodes for the traffic network obtained by the graph-convolution operator. The corresponding traffic graph-convolution features are expressed by GCF_t^k applied on the input data x_t. As we comprehend, we will manipulate values of k as an independent variable. Different values of k (hops with the neighborhood) in traffic graph convolution will end up with different derived features (dependent variable as prediction accuracy). To obtain the aggregated effect from feature space, we concatenate the features obtained from different values of k = [1, ..., K] into a vector. Each value of k represents a neighborhood reachability when convolution operation is applied on the traffic-network graph. It is expressed in the following equation

$$GCF_t^{\{k\}} = [GCF_t^1, GCF_t^2, \ldots, GCF_t^k] \tag{11}$$

The $GCF_t^{\{k\}} \epsilon R^{N \times K}$ contains all the k orders of GCST features from input data x_t. In other words, $GCF_t^{\{k\}}$ is the modified form of x_t transformed to be fed to the neural network (GRU) cell, an intended proposed architecture shown as the left side of Fig. 3. The right side unfolds the idea of proposed graph convolution spatio-temporal (GCST) network operates as a component of our framework. The diagram further explains the convolution operation on the traffic graph at k-hop. We demonstrate the structures $\widetilde{A}^k$ and FFR with respect to the query (red-star) node for clarity and utilized in the combined formula referred to in Equation 10.

D. GCST-GRU: A PROPOSED NN ALGORITHM

We propose a traffic network-based graph convolutional architecture to capture spatio-temporal dependencies from data using GRU neural network model. The framework helps to learn both dependencies from transportation data: a) the latent and spatial dependencies, and b) the temporal dependencies. The proposed neural network architecture is denoted by GCST-GRU in this study. The inner structure of GCST network is explained on the right side of Fig. 3. where a component (in the proposed framework) is exhibited by expressing the graph convolution operations at k-hop. That is the essence. We modified the direct input of traffic-network data x_t fed to the neural network with the GCST operation. The GCST structured matrix contains graph convolution features that constitute an output vector $GCF_t^{\{k\}} \epsilon R^{N \times K}$. The inner structure of the GRU cell with the forget gate r_t, the output gate h_t, and the intermediate gate z_t at time interval t is expressed in the following expressions

$$z_t = \sigma(W_z.GCF_t^{\{k\}} + U^{(z)}h_{t-1}) \tag{12}$$

$$r_t = \sigma(W_r.GCF_t^{\{k\}} + U^{(r)}h_{t-1}) \tag{13}$$

$$h'_t = tanh(W_h.GCF_t^{\{k\}} + r \odot U^{(h)}h_{t-1}) \tag{14}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \tag{15}$$

where matrix multiplication operation is depicted by $\odot$ symbol. W_r, W_z, $W_h \epsilon R^{N \times N}$ are the matrices for searched weights. The σ denotes an activation function called *sigmoid* and *tanh* denotes *hyper-tangent* activation function. The σ is utilized in Equations 12. & 13. and *tanh* in Equation 14.

Let T be the final time step, then the hidden state h_T is observed as the final output of the proposed GCST-GRU network, described as the predicted value $\hat{y}^T = h_T$. Assume that $y^T \epsilon R^N$ denotes the true value of $X_T \epsilon R^N$ (the input observation sometimes called ground truth). For the traffic flow prediction problem the label at time T is then fed as an in-parameter vector to predict the time step ahead *(T+1)*, defined as $y^T = x_{T+1}$. The loss function for training observations is expressed by

$$Loss = L(\hat{y}^T, y^T) = \zeta(h_T, x_{T+1}) \tag{16}$$

where L denotes a loss function. L calculates the delta between the actual value y^T and the predicted value $\hat{y}^T$. At times, mean square error (MSE) measure is utilized to depict the L function for better prediction.

Pseudo code of the proposed NN method is presented in Algorithm 1. To formally state the proposed GCST-GRU model architecture, we describe it this way. Given an original traffic flow time series observations $X_T = [x_1, \ldots, x_t, \ldots, x_T]$ also known as the traffic state; a set of k-hop neighbourhood matrices $\widetilde{A}^1, \widetilde{A}^2, \ldots, \widetilde{A}^k$, and a free-flow reachable matrix FFR. Our target is to calculate a prediction value $\hat{y}^T$ which is equal the the output value h_T. The pseudo-code of the proposed graph-based traffic-network algorithm is expressed with a mechanism to determine the predicted value h_T after T steps of iteration.

Algorithm 1 The GCST-GRU Algorithm to Depict the Output

Input: Original traffic flow time series data $X_T = [x_1, \ldots, x_t, \ldots, x_T]$; a set of k-hop neighbourhood matrices $\widetilde{A}^1, \widetilde{A}^2, \ldots, \widetilde{A}^k$, Free-Flow Reachable Matrix FFR
Output: The prediction value $\hat{y}^T$, Validation error ϵ_v.
Parameters: A set of weights $W_{gc1}, W_{gc2}, \ldots, W_{gcK}$
Initialization: $h_0 = 0 \epsilon R^N$

```
1:  for t ← from 1 to T do
2:    for k ← from 1 to K do
3:      assign: GCF_t^k ← (W_gck ⊙ Ã^k ⊙ FFR) ∗ x_t
4:    end for
5:    assign: GCF_t^{k} ← [GCF_t^1, GCF_t^2, ..., GCF_t^k]
6:    call GCST-GRU-STEP(x_t, GCF_t^{k}, h_{t−1}) {
7:    for i ← 1 to k do
8:      assign: gc ← CONCAT(GCF_t^i, x_t)
9:    end for
10:   assign: combined ← CONCAT(gc, h_{t−1})
11:   assign: z_t = σ(W_z.combined + U^(z) h_{t−1})
12:   assign: r_t = σ(W_r.combined + U^(r) h_{t−1})
13:   assign: h'_t = tanh(W_h.combined + r ⊙ U^(h) h_{t−1})
14:   assign: h_t = z_t ⊙ h_{t−1} + (1 − z_t) ⊙ h'_t
15:   return h_t, gc }
16: end for
17: return h_T
```

In Algorithm 1, at line 6, a function GCST-GRU-STEP() refers to the whole calculation process for a single step during forward processing described in Equations [12-15] in this section. Considering the GCST component, which essentially includes traffic graph convolutional operations, the generated training outcome is a set of graph convolution features $GCF_t^{\{k\}}$ and the learned GC weights $W_{gc1}, W_{gc2}, \ldots, W_{gcK}$. It grants a better chance to build an interpretable model. The proposed model is constructed by analysis of the learned weights GC. Such weights are required to be more stable and interpretable in order to confine the GC features within a proper scale. We apply a couple of the optional regularization terms added to the loss expression defined in Eq. 16. In this work, we specifically build and compare the proposed GCST-GRU model with SCG-LSTM and HGC-LSTM originally applied in [10].

We additionally compared our model with two up-to-date GCN models, Graph WaveNet [34] and Graph Multi-Attention Network (GMAN) [35]. It is observed that Graph WaveNet is a single structured GCN-based technique that can obtain results, but its convergence speed and model accuracy (MAE) are limited. GMAN is another well-known self-learning GCN model. It accomplished a significant improvement in prediction accuracy results being closer to the proposed framework, in spite of the fact that its execution is essentially impacted by the definition of the embedding (hidden state) functions. Since GCST-GRU uses adjacency, free flow reachability, and network node readings present in time and space by applying spatial and temporal attention to a three-order GCN, the proposed approach performed well in terms of performance and accuracy. Due to these very reasons, the experimental results of the proposed method are better in comparison to other state-of-the-art models.

E. GRAPH CONVOLUTION REGULARIZATION FOR TRAFFIC DATA

According to the structure of a proposed GCST-GRU network cell, at any time step T, the output of the cell is expressed as the hidden state h_T, essentially reflecting the predicted value $\hat{y}y^T = h_T$. The loss function is defined as

$$Loss = L(\hat{y}^T, y^T) = \zeta(\hat{y}^T - y^T) \tag{17}$$

where the difference between the true value and the predicted value is presented by the loss function L. Let $y^T \epsilon R^N$ indicate the true value of the input observation in time-series $X_T \epsilon R^N$.To perform the traffic flow prediction, the label at time T is equal to the input vector at the next time interval $(T+1)$, such that $y^T = x_{T+1}$. So the above formula becomes as

$$Loss = L(h_T, x_{T+1}) = \zeta(h_T - x_{T+1}) \tag{18}$$

Let's prepare some background we built so far. We presented a proposed traffic graph convolutional-based network prediction model (GCST-GRU) that clearly replaces the input features to the graph convolution features, and a new reshaped vector is denoted as $GCF_t^k \epsilon R^N$ as presented in Eq. 10. The $W_{gck} \epsilon R^{N \times N}$ is a weight matrix obtained in training process for the k-order traffic graph convolution. Therefore, $W_{gck} = \{W_{gc1}, W_{gc2}, \ldots, W_{gcK}\}$ is a set of weights passed as input parameters to the proposed algorithm. These weights are useful for the interpretation of the proposed model. Regarding Eq. 18, to decrease the loss function, we utilize the power of regularization in two ways: 1) Regularization provides more stable and interpretable learned-weight values, 2) confines the features of the graph convolution operator within reasonable brackets. For such improvement of the loss function, we add two regulation terms for better interpretation.

1) REGULARIZATION ON GC WEIGHTS FOR TRAFFIC DATA

The graph convolution weights W_{gck} can be both positive and negative. Further, each node features are influenced by k neighbouring nodes, therefore we observe a significant variation in graph convolution wights determined during the training process. To get the balanced weights, we need to avoid very high or low weights which start randomly and tend to cancel each other out leading the variations. Regularization is the most likely mechanism for the stabilization of the convolution weights. Although, the high or low weights can still characterize the informative network features, but they are not truly representative of the relationship among nodes in the graph. The first improvement is made by appending

TABLE 1. Comparison of the prediction accuracy of various approaches. The value of *k* is kept to 3, wherever applicable for a model. Epochs are set to 200 for the methods. Some methods stopped earlier due to convergence.

#	Name	MAE±STD	RMSE	MAPE%
1	ARIMA	6.09 ± 1.09	10.65	13.85
2	SVR	6.85 ± 1.17	11.12	14.39
3	FNN	4.45 ± 0.81	7.83	10.19
4	RNN	2.803 ± 0.13	4.069	6.979
5	LSTM	2.782 ± 0.13	4.046	6.959
6	Bi-LSTM	2.985 ± 0.17	4.552	7.949
7	LSGC + LSTM	2.793 ± 0.13	4.059	6.979
8	SGC + LSTM	2.783 ± 0.13	4.034	6.918
9	HGC + LSTM	2.779 ± 0.14	4.048	6.938
10	GRU	2.805 ± 0.12	4.067	6.999
11	Graph Wavenet	2.798 ± 0.13	4.034	6.992
12	GMAN	2.772 ± 0.12	4.072	6.997
13	**GCST-GRU**	**2.761 ± 0.11**	**4.019**	**6.858**

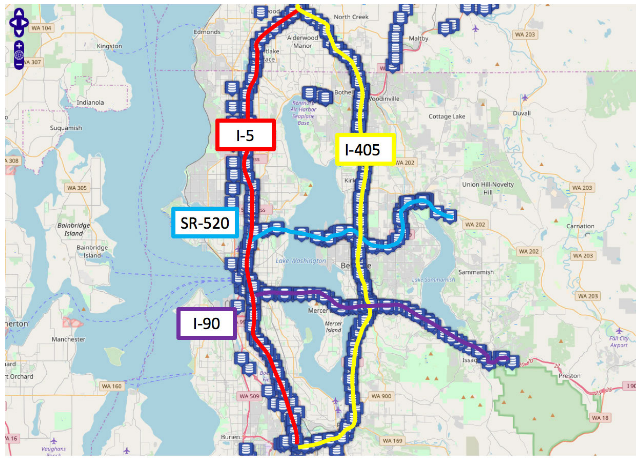

FIGURE 4. Map of the Seattle Area representing the LOOP dataset, displayed as a freeway graph network.

to the loss formula the regularization term, also known as L1-norm on the graph convolution weight matrices W_{gck}. The L1-norm term inculcates randomness by making the weight matrices possibly sparse. The mathematical expression of L1 regularization term is defined as:

$$R^{\{1\}} = ||W_{gc}||_1 = \sum_{i=1}^{K} |W_{gc_i}| \tag{19}$$

where K denotes the number of reachable hops in the neighborhood and i is the index that ranges the neighborhood level starting from self. Therefore, obtaining more stable weights, eventually helps us to distinguish the influential neighboring nodes or their clusters in the traffic network, which contributes more to the prediction process.

2) REGULARIZATION ON GC FEATURES FOR TRAFFIC DATA

The impact of various characteristics of neighboring nodes from the query node must be transmitted or resolved via intermediate nodes. The query node is called the node of interest and neighboring nodes that impact in any way are called influencing nodes. Increasing the number of hops should not impact the graph convolution features abruptly. Therefore to confine the loss between extracted features GCF_t^k of graph convolution on adjacent hops (value of k), we append another L2-norm regularization term, mathematically represented with the following formula

$$R^{\{2\}} = ||GCF_t^K||_2 = \sqrt{\sum_{i=1}^{K-1} (GCF_t^i - GCF_t^{i+1})^2} \tag{20}$$

Therefore, we get more stable features using graph convolution operation on adjacent hops and appending regularization terms. The values in the matrices will not deviate drastically, consequently, the graph convolution operator will keep information more closer to the true values of the correlation hidden in the traffic network graph G. The two regularization terms of the *Loss* function L1 and L2-norm are expressed at the time interval t as follows

$$Loss = \zeta(h_T - x_{T+1}) + \mu_1 R^{\{1\}} + \mu_2 R^{\{2\}} \tag{21}$$

where ζ is the standard loss function that differentiates the real and predicted values. The symbol μ_1 is a coefficient constant used to control the weight of the magnitude of the L-1 regularization term on graph convolution weights while μ_2 is a coefficient constant used to control GCN features using L-2 norm regularization. The original Loss function was unconstrained, arbitrarily varying the values of weights and features. Adding two terms to the Loss function and controlling using coefficient constants ensures that the Loss function is conserved. Consequently, it contributes to the construction of a more stable and interpretable model, resulting in the primary goal of modifying the Loss function. Adding regularization terms to the graph convolution weights and features quickly confines the model. There is a trade-off between prediction accuracy and the magnitude of the penalty terms μ_1 and μ_2, though. It is observed that adding regularization terms to the loss function with penalty rates limited to 0.01 enhances the accuracy of the proposed model tested on the LOOP dataset to roughly 0.4, which stays superior to the other models.

IV. EXPERIMENTAL RESULTS

The data in this research work is collected from a real-time traffic flow time-series web link.[1] The experimental setup is built using python environment with anaconda and experiment is carried out with a 64 bit machine having Windows 10 operating system. In this research work, we further used a few libraries including pytorch, pandas, numpy and matplotlib to ensure experiment evaluation.

[1] https:// github.com / zhiyongc / Seattle-Loop-Data

A. DATASET DESCRIPTION

We used the LOOP dataset for real-time traffic data of highway links situated in the Greater Seattle Area, USA. Sensors are installed along the roads to collect experimental data. Records are collected from inductive loop sensors which are located on four joined freeways (I-5, I-405, I-90, and SR-520), as shown in Fig. 4. This dataset is publicly available at the URL mentioned in the footnote. Data observations contain transportation state records from 323 detectors deployed as road-sides units and measuring observation after every 5-minute interval during the year 2015. According to [10], the author suggested the maximum speed same as the *FFR*, which for the LOOP dataset is fixed to 60mph overall. The distance adjacency matrices D and free-flow reachable matrices *FFR* for the dataset is calculated based on the freeway features and topology.

B. THE EXPERIMENTAL ENVIRONMENT

1) BASELINES

We investigated and compared the performance of the proposed model GCST-GRU with other prediction methods described in TABLE 1. All the models are developed and implemented in a python 3.0 environment with Anaconda installed on the Windows 10, 64-bit operating system with GPU enables machine, using PyTorch 1.6.0. In this experiment, all training and evaluation is performed on a single GPU having 16GB memory. For the proposed GCST-GRU framework, the size of the hidden states is fixed to the number of nodes or adjacent connections in the traffic graphs. The initial value of hops is kept as $K = 3$ and utilized to analyze and evaluate the model in the first experiment. The hop size can fluctuate in the graph convolution for diversity. In this case, the FFR matrices are calculated based on three-time steps. We used RMSProp as the gradient descent optimizer that helps converge the neural network faster.

2) EVALUATION METRICS

To evaluate the effectiveness of the proposed method, we adopted three performance metrics, namely, the mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean square error (RMSE). MAPE is the sum of the individual absolute errors divided by the demand (each period separately). It is the average of the percentage errors. The MAE is a popular KPI to measure prediction accuracy and is the mean of the absolute error. The RMSE is another metric that is helpful for time-series predictions and is defined as the square root of the average squared error. $e_T = |\hat{y}^T - y^T|$ is the mathematical expression of these measures, defined as the error defined by the difference between the predicted value and ground truth.

C. EXPERIMENTAL RESULTS

Results of the proposed GCST-GRU and other baseline methods are compared in TABLE 1 applied with the LOOP dataset by fixing the number of EPOCHs to 200. The proposed NN model exceeds other models as its accuracy measures are lesser than other methods on the experimental dataset. The traditional methods ARIMA, SVR, and FNN cannot be judged fairly with the deep learning methods because of their inefficiency in incorporating data variation patterns for traffic network prediction. Basic FNN is a shallow learning model that achieves results better than both ARIMA and SVR but still performs worse for predicting traffic data with spatio-temporal dependencies.

Fig. 5 shows the model accuracy measure - mean absolute error (MAE) on the x-axis while presenting the traffic prediction algorithm in comparison to the influence of graph convolution in the proposed GCST-GRU method. All experiments are performed on the LOOP dataset. It's observed that the proposed model performance is directly proportional to the value of k. For evaluation, We fixed $k = 3$ in our experiments for the comparison of the results because some algorithms are traditional and don't use neighborhood concepts at all, as shown in TABLE 1.

The rest of the sophisticated deep learning models: RNN, LSTM, Bi-LSTM, and GRU, and graph convolution methods: LSGC + LSTM, SGC + LSTM, and HGC + LSTM, Graph Wavenet, and GMAN have produced competitive results. We can observe the proposed method GCST-GRU outperforms the other graph convolution methods: LSGC + LSTM, SGC + LSTM, and HGC + LSTM when experimenting on the proposed dataset. Generally, we can observe that simple LSTM also performed well from some graph convolution methods. We believe when using the right value of k, results can exceed the LSTM as presented in TABLE 1. Furthermore, an observation for GRU is expected to perform comparable to LSTM but that is not in this experiment. However, as expected, the GRU's training time is less than LSTM's due to structural less overhead taking advantage. GRU doesn't utilize the the cell-state as compared to the LSTM model. In LSTM, such a cell-state is important to track historical information in the data. The proposed framework GCST-GRU, which can capture both spatial and temporal features of graph-based traffic-network wide data, is observed to perform better than all other approaches.

D. OPTIMUM VALUE OF K-HOPS NEIGHBORHOOD

We determine the optimal value of k to be used in this study. Such value is important to perform further experimentation. A proposed GCST-GRU model execution is performed for various values of k-hops {2, 3, 4, 5, 8, 16, 32} and vertical-axis is one of the metrics like mean absolute error, root-mean square error, and mean-absolute percentage error as expressed in Fig. 5. Another part of the diagram at the bottom-right contains the training time of each k-hop network measured in seconds. All experiments are calculated using the proposed GCST-GRU method using the LOOP dataset. Our objective of this approach is to determine the optimum value of k. We found that $k = 3$ is a better value to continue with our experiment to analyze and compare the models in this experiment as expressed with the results in

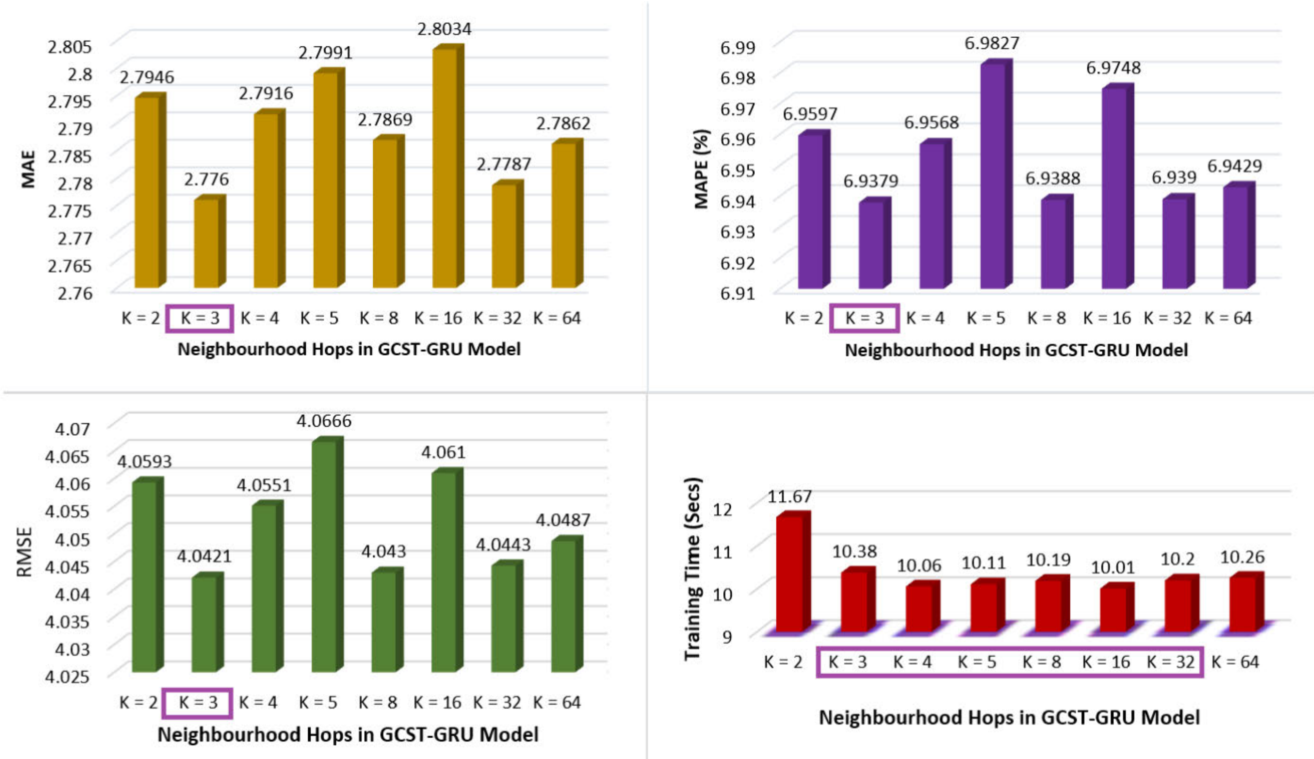

FIGURE 5. The proposed GCST-GRU model execution is performed. The horizontal axis in the diagram is expressed as various values of k-hops and vertical-axis is one of the metrics. All experiments are calculated using the LOOP dataset. We found that k=3 is a better value to continue with our experiment.

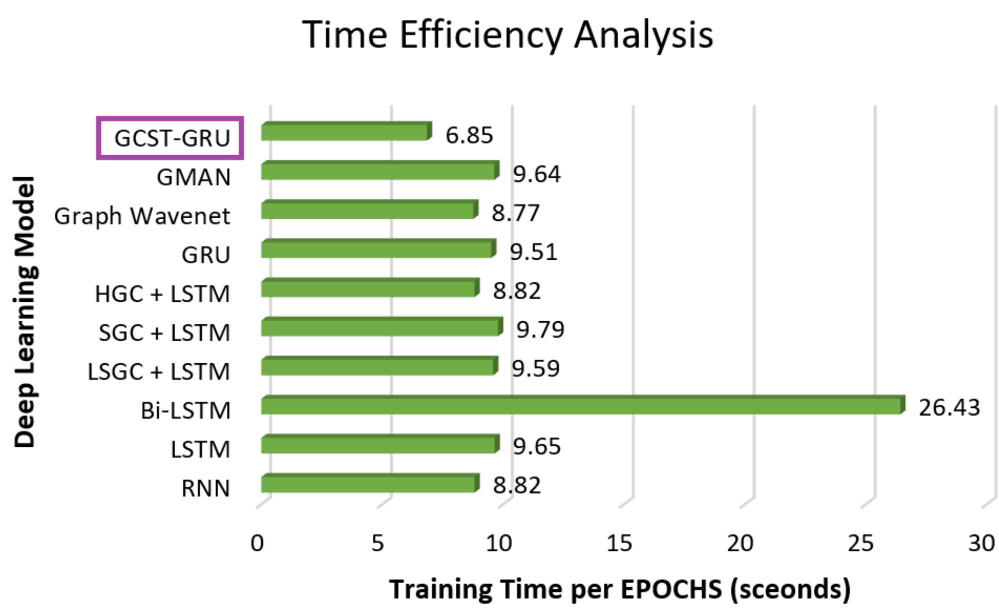

FIGURE 6. Comparison of various methods and their training time per EPOCH in seconds for deep learning models.

TABLE 1. We have already discussed in a previous section, k is the segment-length obtained by counting the minimum number of links traversed from the source node to other nodes in the given graph. The adjacency matrix is useful to determine the k-th order neighborhood for each node. Its outcome is a matrix called k-hop neighborhood for traffic

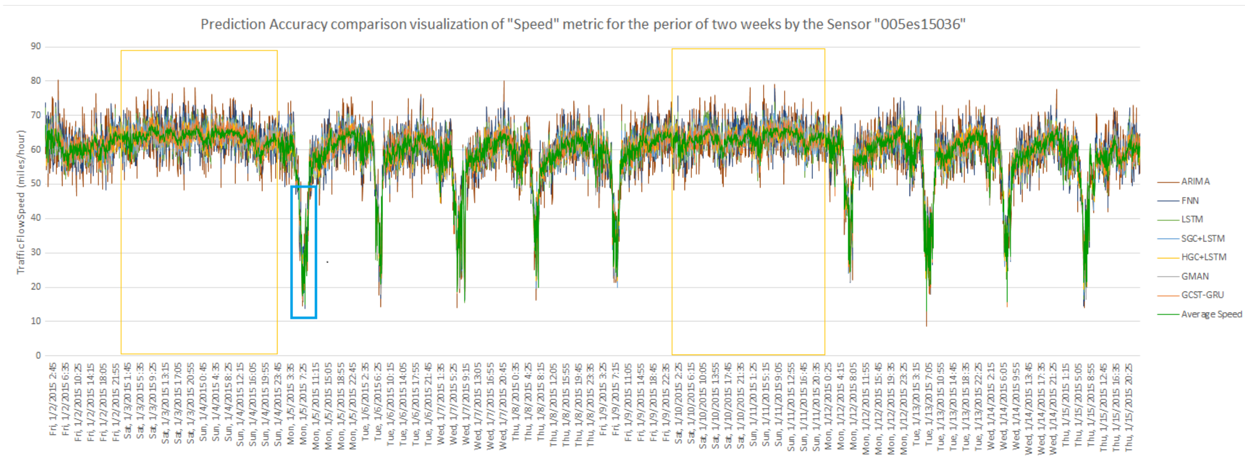

FIGURE 7. Loop dataset speed data for sensor# 005es15036 captured during the year 2015 after the interval of 5 minutes.

data. We keep the k value minimal and focus on achieving accuracy, although we can observe that the training time is not minimal at only $k = 3$ as expressed in the bottom-right part of Fig. 5.

E. TRAINING EFFICIENCY

In this subsection, we compare various methods and their training time as presented in Fig. 6. It shows the training time with a bar graph of each deep learning method. The proposed NN model exceeds other models as its training time is less than other methods on the experimental data. A key focus of the proposed model (GCST-GRU) is to explore the temporal and spatial dependencies using matrices, road-segment graphs, and dynamic patterns. We have shown that GCN operation over a GRU network provides better performance because GRU is a simpler version than LSTM. The actual numbers of training epochs are variable because some algorithms converge earlier and stop during the training.

F. IMPACT OF REGULARIZATION ON MODEL INTERPRETATION

The loss function is modified to include two regularization terms L1 and L2-norms. The former is used to regularize the graph convolution weights while later providing regularization on the graph convolutional features. Adding both regularization terms helps to improve the trained model by obtaining the graph convolution weights to be more stable, sparse, clustered, intuitive, and interpretable. We compute the averaged matrices for graph convolution weight for the experimental dataset (LOOP in this case), setting the value of k equals 3. The expression that can determine the average weight formula is defined as $W_{avg} = \frac{1}{K}\sum_{i=1}^{K}(W_{gck} \odot \widetilde{A}^k \odot FFR)$. In this work, we determine the traffic states as the node's self-connectedness (imagine identity matrix I is added to adjacency matrix A earlier) so most of the weights are interpreted in training data at the diagonal elements of the weight matrix. The proposed method (GCST-GRU) helps capture spatio-temporal dependencies comparable to [10] in the traffic network to identify the most influential road segments or group of nodes, depicted by comparing the average graph convolution weight matrix. The steps used to include the L1-norm factor and express the improved error is described in Algorithm 2.

Algorithm 2 The L1-Norm Term Inclusion Algorithm to Improve the Error Loss When Applied Over the Proposed Algorithm Using Pytorch Library.

Input: Original traffic flow observation x_t;
Output: MAE and STD values after including L1-norm term with improved error loss
Given:
testDataSet: a sequence of observations of the traffic data as test dataset.
speedMatrix: a matrix with spatio-temporal speed data specified for each segment over traffic network

$max\_Speed = speedMatrix.max()$
for each *dataPoint* in *testDataSet* **do**
  apply $GCST\_GRU(dataPoint)$
  assign $L1\_loss = torch.nn.L1Loss()$
  assign $L1\_matrix = L1\_matrix.Append(L1\_loss)$
end for
assign $L1\_mean = mean(L1\_matrix) * max\_Speed$
assign $L1\_std = np.std(L1\_matrix) * max\_Speed$
return $L1\_mean, L1\_std$

The term $L1\_mean$ is expressed as the mean absolute error provided in TABLE 1. One can further explore the utilization of the L1-norm to reduce error loss and improve the model by applying other parameters to determine significant road sections within the dataset. In this study, we presented the graph convolution operation over a neural network and observed that the proposed model (GCST-GRU) exceeds

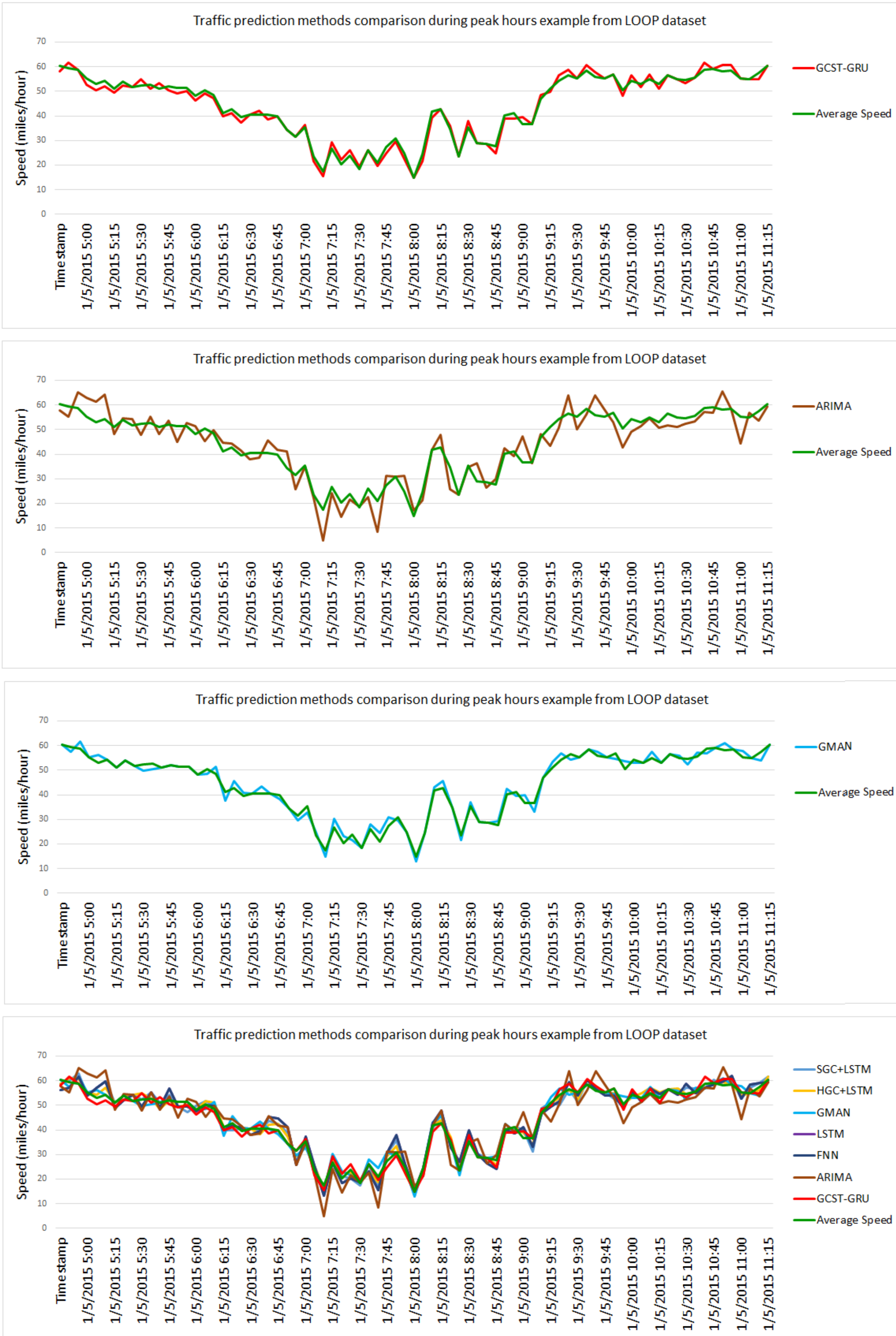

FIGURE 8. **Loop dataset speed data for sensor# 005es15036 captured during the peak hours on 1/5/2015. The top figure shows the prediction comparison of average speed with STGC-GRU, GMAN, ARIMA, and the consolidated view. our proposed method has performed well over baseline methods as shown in the top graph under congestion situation.**

TABLE 2. Comparison of the prediction accuracy of various approaches under peak hours (congestion state). The value of k is kept to 3, wherever applicable for the model.

#	Model	MAE	RMSE	MAPE%
1	ARIMA	4.55	9.64	14.93
2	FNN	3.01	6.56	8.78
3	LSTM	1.56	**3.03**	5.75
4	SGC + LSTM	3.83	5.30	5.60
5	HGC + LSTM	4.24	5.62	8.44
6	GMAN	4.08	5.14	8.01
7	**GCST-GRU**	**1.42**	5.11	**4.08**

other models with lesser training time when applied to the experimental traffic data, referred to the Fig. 7. By virtue of the traffic data nature that includes both spatial and temporal dependency relationships, our focus is to utilize the adjacency matrix, k-hop matrix for the road-segments in the traffic network graph and dynamic latent patterns. We have shown that GCN operation over the GRU network provides better performance because of the neural network simplicity over other RNN models.

G. IMPACT OF TRAFFIC PREDICTION UNDER CONGESTION AND FREE-FLOW CONDITIONS

We evaluated the effect of traffic speed on traffic next state prediction in two scenarios: congestion and free-flow. We randomly picked one sensor from the LOOP dataset to assess prediction accuracy in order to visualize the prediction gaps between the proposed model and other baseline models. The influence of daily and weekly periods on traffic flow measurements is clearly visible in Fig. 7 since the commute usually iterates through a week, as marked by contour rectangles in the graph. The green color in the graphs represents the ground truth values of traffic flow in the LOOP dataset taken from sensor# *005es15036* with a 5-minute interval over the two weeks of January 2015. Fig. 7 presents the speed prediction values of seven algorithms *GCST-GRU, GMAN, HGC+LSTM, SGC+LSTM, LSTM, FNN, and ARIMA*. The data captured by a specific sensor# *005es15036* for fifteen consecutive days is investigated.

For short-term traffic prediction, the results reveal that the graph values of all baseline models are closer to the true traffic flow values. When there is free-flow, conditions occur, the average traffic speed stays around 60 miles per hour, whereas, during peak hours (congestion scenarios), the traffic flow is considerably less than 20 miles per hour. On weekends, the trend is such that roads are less occupied, leading to an increased free-flow situation. There are traffic spikes in a downward direction during the peak hours on weekdays. Fig. 8 is the zoomed-in presentation of Fig. 7, where it's marked in the blue rectangle region. In Fig. 8 during peak hours, the proposed approach performs well, since the prediction flow divergence from actual speed is closer than existing baseline methods. The results presented in TABLE 2 indicate that it is difficult to anticipate traffic flow when there is congestion, however, findings are encouraging when compared to baseline methods. The comparison of the prediction accuracy of various approaches under peak hours when the congestion state is active to the transition phase when congestion is being resolved is difficult to predict. We are skipping additional comparison results in this research endeavor due to space constraints.

V. CONCLUSION AND FUTURE WORK

This paper presents a novel deep learning framework, called Graph Convolutional with Spatio-temporal Gated Recurrent Unit (GCST-GRU) neural network, to predict short-term traffic on graph networks. The model is evaluated on a traffic speed dataset containing detectors deployed in the Greater Seattle Area, USA. The experiments demonstrate that this framework is superior to other state-of-the-art baseline models with respect to accuracy measured by the metrics denoted by RMSE, MAE, and MAPE. We also discuss the spatial and temporal distribution of the validation errors produced by the GCST-GRU model. Two regularization terms are used to improve the loss function to determine the graph convolution averaged weight matrix. This work can be extended in the following possible ways. First, the framework should be tested on a larger dataset, for that reason, we need to be careful about the computation cost of $N \times N$ adjacency and FFR matrices and access the citywide traffic predictions. Secondly, some researchers discussed calculating an efficient loss function by incorporating adjacency factors for faster convergence. Thirdly, the goal of this research is to create a deep learning framework that uses spatial-temporal data along with GRU to predict short-term traffic states on graph networks. We analyzed the spatial and temporal distribution of the validation errors induced by the proposed approach. We also added two regularisation terms to improve the loss function and therefore the model by providing a more interpretable averaged weight matrix for the graph network. When comparing the state transition from congestion condition to free-flow or vice versa, it is possible to determine the traffic states in the future. Last but not least, We can apply more robust prediction methods to consider external factors that determine the traffic prediction (e.g. the rainy weather, pandemic, or unwanted events), which is still an open problem area where some scientists performed research [31].

REFERENCES

[1] K. Shaaban, M. H. M. Shamim, and K. Abdur-Rouf, "Visible light communication for intelligent transportation systems: A review of the latest technologies," *J. Traffic Transp. Eng. English Ed.*, vol. 8, no. 4, pp. 483–492, Aug. 2021, doi: 10.1016/j.jtte.2021.04.005.

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

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

[4] 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.

[5] 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, p. 818, Apr. 2017.

[6] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

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

[8] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[9] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis and future directions," 2020, *arXiv:2004.08555*.

[10] 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, doi: 10.1109/TITS.2019.2950416.

[11] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015, doi: 10.1109/TITS.2014.2345663.

[12] B. Yu, M. Li, J. Zhang, and Z. Zhu, "3D graph convolutional networks with temporal graphs: A spatial information free framework for traffic forecasting," 2019, *arXiv:1903.00919*.

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

[14] B. Hussain, M. K. Afzal, S. Ahmad, and A. M. Mostafa, "Intelligent traffic flow prediction using optimized GRU model," *IEEE Access*, vol. 9, pp. 100736–100746, 2021, doi: 10.1109/ACCESS.2021.3097141.

[15] G. Liu, H. Bao, and B. Han, "A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis," *Math. Problems Eng.*, vol. 2018, pp. 1–10, Jul. 2018, doi: 10.1155/2018/5105709.

[16] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intell. Transport Syst.*, vol. 11, no. 9, pp. 531–536, Nov. 2017.

[17] 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, doi: 10.1109/TITS.2019.2935152.

[18] H. Yang, X. Zhang, Z. Li, and J. Cui, "Region-level traffic prediction based on temporal multi-spatial dependence graph convolutional network from GPS data," *Remote Sens.*, vol. 14, no. 2, p. 303, Jan. 2022, doi: 10.3390/rs14020303.

[19] 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, doi: 10.1007/s42488-020-00037-9.

[20] J. Li, J. Yang, L. Gao, L. Wei, and F. Mao, "Dynamic spatial–temporal graph convolutional GRU network for traffic forecasting," in *Proc. 6th Int. Conf. Syst., Control Commun. (ICSCC)*. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 19–24, doi: 10.1145/3510362.3510366.

[21] W. Jiang, Y. Xiao, Y. Liu, Q. Liu, and Z. Li, "Bi-GRCN: A spatio-temporal traffic flow prediction model based on graph neural network," *J. Adv. Transp.*, vol. 2022, pp. 1–12, Feb. 2022, doi: 10.1155/2022/5221362.

[22] 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, doi: 10.1609/aaai.v33i01.3301485.

[23] K. Guo, Y. Hu, Z. Qian, H. Liu, K. Zhang, Y. Sun, J. Gao, and B. Yin, "Optimized graph convolution recurrent neural network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1138–1149, Feb. 2021, doi: 10.1109/tits.2019.2963722.

[24] Y. Huang, Y. Weng, S. Yu, and X. Chen, "Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Communications/13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 1–16.

[25] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, *arXiv:1506.05163*.

[26] H. Wang, E. S. L. Ho, H. P. H. Shum, and Z. Zhu, "Spatio-temporal manifold learning for human motions via long-horizon modeling," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 1, pp. 216–227, Jan. 2021.

[27] A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction," *Neural Netw.*, vol. 145, pp. 233–247, Jan. 2022, doi: 10.1016/j.neunet.2021.10.021.

[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–13.

[29] Z. Zhou and X. Li, "Graph convolution: A high-order and adaptive approach," 2017, *arXiv:1706.09916*.

[30] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial–temporal attention approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4909–4918, Aug. 2021, doi: 10.1109/TITS.2020.2983651.

[31] R. W. Liu, M. Liang, J. Nie, Y. Yuan, Z. Xiong, H. Yu, and N. Guizani, "STMGCN: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multigraph convolutional network," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7977–7987, Nov. 2022, doi: 10.1109/TII.2022.3165886.

[32] M. Liang, R. W. Liu, Y. Zhan, H. Li, F. Zhu, and F.-Y. Wang, "Fine-grained vessel traffic flow prediction with a spatio-temporal multigraph convolutional network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23694–23707, Dec. 2022, doi: 10.1109/TITS.2022.3199160.

[33] B. Yu, Y. Lee, and K. Sohn, "Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN)," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 189–204, May 2020, doi: 10.1016/j.trc.2020.02.013.

[34] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," 2019, *arXiv:1906.00121*.

[35] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proc. AAAI*, 2020, vol. 34, no. 1, pp. 1234–1241, doi: 10.1609/aaai.v34i01.5477.

[36] 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, doi: 10.1109/TITS.2021.3072743.

BASHARAT HUSSAIN received the M.S. degree in computer science from International Islamic University (IIUI), Islamabad, Pakistan. He is currently pursuing the Ph.D. degree in computer science with COMSATS University Islamabad, Wah Campus. He is a senior IT specialist with 23 years of experience in developing software applications for multinational companies in Europe and USA. He has professional experience in software design and architecture using various languages, including C++, C#, ASP.NET Core, Java, Matlab, and Python. He has extensive academic experience in teaching and research with major universities in Pakistan. He is also a Senior Lecturer with the Faculty of Computing, Riphah International University, Islamabad, Pakistan. His current research interests include applying deep learning and federated learning to solve ongoing problems.

MUHAMMAD KHALIL AFZAL (Senior Member, IEEE) received the M.C.S. and M.S. degrees in computer science from the COMSATS Institute of Information Technology, Wah Campus, Pakistan, in 2004 and 2007, respectively, and the Ph.D. degree from the Department of Information and Communication Engineering, Yeungnam University, South Korea, in December 2014. From January 2008 to November 2009, he was a Lecturer with Bahauddin Zakariya University, Multan, Pakistan; and King Khalid University, Abha, Saudi Arabia, from December 2009 to June 2011. Currently, he is an Associate Professor (tenured) with the Department of Computer Science, COMSATS, Wah Cantt Campus, Pakistan. He received the fully funded scholarship for the master's and Ph.D. studies. He was a recipient of research projects, funded by the Higher Education Commission (HEC), Pakistan; the National Grassroots ICT Research Initiative, Ignite, Pakistan; the European Union's Program Horizon 2020; and the British Council U.K. His research interests include wireless sensor networks, ad hoc networks, data-driven intelligence in wireless networks, smart cities, 5G, and the IoT. He was a recipient of the Best Paper Awards of ACM 7th International Conference on Computing Communication and Networking Technologies and in International Conference on Green and Human Information Technology (ICGHIT-2018). He was a Reviewer of IEEE Access, *Computers and Electrical Engineering* (Elsevier), *Journal of Network and Computer Applications* (Elsevier), *FGCS*, and IEEE Transactions on Vehicular Technology. He has served as a Guest Editor for IEEE Internet of Things Journal, *Computer Communications* (Elsevier), *IEEE Communications Magazine*, *Transactions for Emerging Telecommunications Technologies* (ETT), *Future Generation Computer Systems* (Elsevier), IEEE Access, *Journal of Ambient Intelligence and Humanized Computing* (Springer), and *Sensors* (MDPI).

SHERAZ ANJUM received the first M.Sc. degree in electronics from Quaid-E-Azam University, Islamabad, Pakistan, in 1999, the second M.Sc. degree in computer engineering from UET, Taxila, Pakistan, in 2005, and the Ph.D. (Eng.) degree in microelectronics from the Institute of Microelectronics, GUCAS, Beijing, China, in 2008. Currently, he is a Professor and the Head of the Department of Computer Science, COMSATS University Islamabad, Wah Cantt Campus, Pakistan. He has around 24 years of teaching, research, and administrative experience. His research interests include the design and analysis of networks on chip architectures and algorithms, multi-processor heterogeneous computing, the Internet of Things, intelligent transportation systems, wired and wireless networks, and computer architectures. He is serving as a reviewer for many international journals.

IMRAN RAO received the Ph.D. degree from The University of Melbourne, Australia. He is the Director of IT with Blue Brackets Technologies Pvt. Ltd., Australia and Pakistan. He is a senior IT professional with vast academic experience in teaching and research with prestigious universities. His current research interests include finding solutions to issues related using deep learning and quantum computing.

BYUNG-SEO KIM (Senior Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Inchon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Yuguang Fang. From 1997 to 1999, he was with Motorola Korea Ltd., Paju, South Korea, as a Computer Integrated Manufacturing (CIM) Engineer in advanced technology research and development (ATR&D). From January 2005 to August 2007, he was with Motorola Inc., Schaumburg Illinois, as a Senior Software Engineer in networks and enterprises. His research focuses in Motorola Inc. were designing protocol and network architecture of wireless broadband mission critical communications. From 2012 to 2014, he was the Chairman with the Department of Software and Communications Engineering, Hongik University, South Korea, where he is currently a Professor. His work has appeared in around 167 publications and 22 patents. His research interests include the design and development of efficient wireless/wired networks, including link-adaptable/cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, mobile edge computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks.

He served as a member for Sejong-City Construction Review Committee and Ansan-City Design Advisory Board. He served as the General Chair for 3rd IWWCN 2017; and a TPC Member for IEEE VTC 2014-Spring, EAI FUTURE2016, and ICGHIC 2016 2019 Conferences. He served as a Guest Editor for special issues of *International Journal of Distributed Sensor Networks* (SAGE), IEEE Access, and *Journal of the Institute of Electrics and Information Engineers*. He is an Associative Editor of IEEE Access.