

1 **A Data-Driven Network Model for Traffic Volume Prediction at Signalized Intersections**

2 Rezaur Rahman (Corresponding Author)

3 Ph.D. Candidate

4 Department of Civil, Environmental, and Construction Engineering

5 University of Central Florida

6 Email: rezaur.rahman@knights.ucf.edu

7
8 Jiechao Zhang

9 Ph.D. Candidate

10 Department of Civil, Environmental, and Construction Engineering

11 University of Central Florida

12 Email: zlyj123@Knights.ucf.edu

13
14 Sudipta Dey Tirtha

15 Ph.D. Candidate

16 Department of Civil, Environmental, and Construction Engineering

17 University of Central Florida

18 Email: sudiptadeytirtha2018@knights.ucf.edu

19
20 Tanmoy Bhowmik, Ph.D.

21 Post-Doctoral Associate

22 Department of Civil, Environmental, and Construction Engineering

23 University of Central Florida

24 Email: tanmoy78@Knights.ucf.edu

25
26 Md Istiak Jahan

27 Graduate Research Assistant

28 Department of Civil, Environmental & Construction Engineering,

29 University of Central Florida

30 Email: mdistiakjahan@knights.ucf.edu

31
32 Naveen Eluru, Ph.D.

33 Professor

34 Department of Civil, Environmental, and Construction Engineering

35 University of Central Florida

36 Email: Naveen.Eluru@ucf.edu

37
38 Samiul Hasan, Ph.D.

39 Assistant Professor

40 Department of Civil, Environmental, and Construction Engineering

41 University of Central Florida

42 Email: samiul.hasan@ucf.edu

43

44

45

ABSTRACT

Network-wide traffic prediction at the level of an intersection can benefit transportation systems management and operations. However, traditional traffic modeling approaches relying on mathematical or simulation-based models are either less useful or require higher computational time in predicting high fidelity traffic volumes. In addition, these frameworks need to be modified to ingest large-scale data (such as automated traffic signal performance measures) available from intersections. To overcome these challenges, in this study, a data-driven method based on a deep learning architecture has been developed for network-wide intersection-level traffic prediction. The study has tested two deep learning architectures: Graph Convolutional LSTM (*GCN-LSTM*) and Graph Convolutional Encoder Decoder LSTM (*GCN-Encoder-Decoder*) model to predict intersection-level hourly traffic movement volumes over multiple time steps (e.g., 4-hour sequence). Such deep learning architectures capture the spatiotemporal cross correlation among network wide traffic features while learning the patterns in traffic movement volumes. To test the model performances, we have fused data from multiple sources such as travel demand data, built environment characteristics etc. We have extracted 1-year (2016) of traffic movement volume data from Seminole County's automated traffic signal performance measure (ATSPM) database. Experiment results show that the developed *GCN-LSTM* model outperforms all the other baseline models. The absolute difference between actual and predicted volumes are quite low ($GEH < 5$); for right turn, through and left turn movement RMSE values are 4.02, 59.37, and 2.47, respectively. The R^2 score of the model is 0.98, which indicates that the model can capture the spatiotemporal variations of traffic movement volumes very well.

Index Term – Network model, artificial intelligence, traffic prediction, ATSPM

1. Introduction

Network-wide traffic prediction can benefit transportation systems management and operations including measuring traffic signal performance, optimizing signal timing plans, and managing incidents. Typically, traditional mathematical or simulation-based network modeling approaches have been applied to estimate traffic flows (Peeta and Ziliaskopoulos, 2001; Yi-chang et al., 2011). Although these approaches produce reasonable solutions for traffic prediction problems, the complexity and computation time required to implement such models make them less suitable for real world applications such as real-time link flow or intersection-level traffic volume prediction. In addition, these frameworks need to be modified to ingest large-scale data (such as automated traffic signal performance measures) available from intersections.

Ubiquitous use of sensing technologies such as probe vehicles, roadway detectors, and social sensors has created an opportunity to overcome these challenges and implement deployable modeling approaches to predict traffic at a higher resolution such as the level of an intersection (i.e., direction wise and movement wise). However, large-scale network level traffic forecasting is more challenging due to higher computational complexity because of network size. For this purpose, a robust prediction model is required with the ability to capture spatial correlation of traffic among adjacent roads and learn driver route choice behavior from high resolution data.

In recent years, researchers are exploring different data-driven approaches for traffic prediction. However, existing data-driven approaches have several limitations such as, they predict only short-term traffic states (speed, flow, travel time) (Billings and Jiann-Shiou, 2006; Chen et al., 2021; Deshpande and Bajaj, 2016; Lee, 2009; Vlahogianni et al., 2014; Wu et al., 2004; Yu et al., 2016)

for one or multiple segments of highways, but not at the scale of a network (Gu et al., 2019; Rahman and Hasan, 2018; Saroj et al., 2021; Song et al., 2018). Moreover, these approaches do not consider features related to travel demand and land use when predicting future traffic. As such, these approaches consider traffic prediction as a simple time series problem and predict traffic state for a shorter time horizon (e.g., next 5 to 60 mins).

To overcome the limitations of existing data-driven methods, we develop a new modeling framework for intersection-level movement volume prediction for a large-scale network. We develop a graph convolution based deep learning model to predict traffic movement volume by capturing the correlation between demand features and traffic flow patterns of a transportation network. To test the model, we have collected Automated Traffic Signal Performance Measure (ATSPM) data and zonal-level travel demand data for Seminole County, Florida. This study has made several contributions:

- It develops a data pipeline incorporating extensive data assessment approaches to extract and process traffic movement volumes from ATSPM data;
- It develops a new method for intersection-level traffic prediction considering the correlation between travel demand and traffic flow inside a transportation network; and
- It provides empirical evidence on the performance of a deep learning-based model for traffic prediction using real-world large-scale traffic signal performance data.

2. Literature Review

Understanding traffic evolution and congestion propagation for an entire road network rather than a single road will be more helpful for traffic managers in transportation planning and proactive decision making (Zhan et al., 2017; Zhang et al., 2011). However, large-scale network level traffic forecasting is more challenging due to higher computational complexity incurred by the network size and topology, which requires a robust prediction model, with the ability (i) to capture the spatial correlation of traffic in interconnected roads, and (ii) to predict traffic for a long-term to reflect congestion propagation. But traditional traffic prediction models (Mahmoud et al., 2022, 2021; Qiao et al., 2013; Qu et al., 2020; Rahman and Hasan, 2021) only consider temporal variations of traffic state variables, thus cannot deal with high dimensionality of the data to learn spatial correlation.

Convolutional LSTM methods are the initial attempt to model the spatial and temporal correlation among the traffic states for network level traffic prediction. A few studies (Bao et al., 2019; Guo and Zhang, 2020; Hao et al., 2019; Ma et al., 2017) have implemented the convolutional LSTM model for network level traffic speed, travel demand and crash risk prediction. Although this model outperforms existing state of art data-driven model, it does not consider stochastic traffic flow dynamics (i.e., flow propagation) while extracting spatial correlation among network traffic. Recently, graph convolution neural network (Guo et al., 2021; Kipf and Welling, 2016; Wu et al., 2020; Zhang et al., 2019; Zhou et al., 2020) has been emerging as a new approach to overcome the limitation of convolutional neural networks in traffic prediction problem. Graph convolutional neural network approaches utilize the concepts of graph theory along with deep neural network architectures to model the stochastic traffic dynamics inside a network. These approaches aim at learning the interactions between roadways in the traffic network to forecast network-wide traffic states. However, the application of such a neural network architecture hardly exists for a large-

scale transportation network. A few studies have utilized the concept of graph convolution to represent traffic network as a generalized graph for traffic state prediction (Bogaerts et al., 2020; Cui et al., 2020a, 2020b; Guo et al., 2021; Li et al., 2021; Y. Li et al., 2018; Z. Li et al., 2018; Zhang et al., 2021; Zhao et al., 2020). These studies, however, have focused on learning network-wide short-term correlations among traffic states (e.g., speed) to predict future states 5 to 60 mins ahead of time.

Data-driven methods for intersection level traffic prediction mostly involve traffic flow (Alajali et al., 2018), movement volume prediction (Ghanim and Shaaban, 2019; Li et al., 2020; Mahmoud et al., 2021) and traffic signal queue length prediction (Chang and Su, 1995; Lee et al., 2019; Rahman and Hasan, 2021). These studies are highly data intensive; previous studies either used a simulation-based or a hybrid approach to develop intersection-level traffic prediction models. For instance, Chang and Su (Chang and Su, 1995) developed a data-driven neural network model for predicting queue length at short time step (3s). They used the data from simulation experiments to train the model for queue prediction. Lee et al. (Lee et al., 2019) developed a deep learning model for queue length estimation. They relied on traffic simulations to generate the training data and used real-world driving data from the Federal Highway Administration's Next Generation Simulation (NGSIM) program to test the approach. However, one limitation is that these approaches are based on isolated intersections without considering the coordination among multiple intersections.

Recently, Alajali et al. (Alajali et al., 2018) applied gradient boosted decision tree based model to predict intersection traffic volume for large scale network which covers intersections at central business district (CBD) area of Melbourne, Australia. This study proposed an online and offline training approach to deal with the limitation in computation power for large scale data. However, the proposed method is limited to aggregate traffic volume prediction at intersection level rather than traffic movement volumes. A recent study by Li et al. (Li et al., 2020) proposed a deep learning method to predict intersection level traffic movement volume. This study utilizes Convolutional LSTM model to capture spatiotemporal dependency among network-wide traffic states considering traffic network as an image. Thereby fails to capture the stochastic traffic flow dynamics of the network. Moreover, the proposed approach does not consider travel demand features, thus limited to only short-term traffic movement prediction (i.e., 5-15 min ahead of current time).

In summary, we find that most of previous studies adopted deep learning models without considering directional traffic volume. Moreover, capturing the spatiotemporal dependency of traffic for interconnected roadway segments can improve the model performance compared to traditional times series-based approaches. However, still there exists several research gaps in application of deep learning methods for intersection-level traffic movement volume prediction. First, these deep learning methods have not been tested for network scale intersection level traffic movement volume prediction, rather applied over isolated intersections or at a corridor level. Second, although graph theoretic approach has been applied for detector level or segment wise traffic state prediction, no study has considered traffic dynamics and tested its influence on model accuracy. Third, previous studies focused on short-term or cycle-level traffic prediction, thus, do not consider any demand related features to account for long-term demand variations. However, for predicting traffic over a long-term period (>1 hr.) we need to add the information on travel demand variation over different periods (peak hour, off peak hours etc.).

3. Problem Formulation

We present a data-driven approach of learning intersection-level traffic movement patterns of a transportation network given that information on travel demand and corresponding traffic movement volumes are available. Instead of estimating traffic movement patterns using traditional traffic assignment models, here we implement the idea of learning those movement patterns from large-scale training data. Adopting the concept of graph convolution, we develop a deep learning model to capture the cross correlation among spatiotemporal traffic features to predict traffic movement over a long-term sequence.

To implement the model, we represent the transportation network as a graph where each node indicates an intersection, and the edges indicate the shortest path distance between two intersections. Let, $\mathcal{G}(v, \mathcal{E}, A)$ is an undirected graph, where v denotes the set of nodes (i.e., intersection) and $\mathcal{E}$ denotes the set of links between nodes (i, j) . A represents the connectivity between nodes as a weighted adjacency matrix, where weights are based on the distance between any two nodes (i, j) , defined as follows:

$$A(i, j) = \begin{cases} d_{i,j} & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where, $d_{i,j}$ denotes the travel distance between the origin i and the destination j nodes. The connectivity inside an adjacency matrix detects which neighboring nodes (j) will be influenced by the traffic condition at a given node (i). Moreover, in a timeseries problem the existing traffic condition at a given node (i) will also influence its future traffic condition, which means each node is temporally self-influenced. This is represented by adding an identity matrix (I) with the adjacency matrix which ensures that nodes are self-accessible,

$$\bar{A} = A + I \quad (2)$$

The proposed data-driven traffic prediction model aims to *learn* intersection-level traffic movement volume of a transportation network over multiple future time-steps based on capturing the spatiotemporal correlation among traffic features at different intersections. In other words, the framework captures how traffic at a given intersection influences the traffic condition at other intersection based on the distance between two intersections. Thus, we feed the model with information of two aspects: (i) a weighted adjacency matrix ($\bar{A}$) indicating the connectivity and distance between intersections and (ii) node level features ($\mathbf{X}_t$) representing traffic demand and traffic state variations.

The traffic prediction problem aims to learn a function $\mathcal{F}(\cdot)$ that maps l instances of input sequence $([\mathbf{X}_{t-l}, \mathbf{X}_{t-l+1}, \dots, \mathbf{X}_t])$ to predict intersection-level traffic movement volume $(\mathbf{F}_{t+1}, \mathbf{F}_{t+2}, \dots, \mathbf{F}_{t+p})$ for the entire network. We define the problem as follows:

$$\mathcal{F}([\mathbf{X}_{t-l}, \mathbf{X}_{t-l+1}, \dots, \mathbf{X}_t]; \mathcal{G}(v, \mathcal{E}, \bar{A})) = [\mathbf{F}_{t+1}, \mathbf{F}_{t+2}, \dots, \mathbf{F}_{t+p}] \quad (3)$$

where, $l(= 0, 1, 2, \dots, l)$ and $p(= 1, 2, 3, \dots, p)$ indicates the input and output sequence; $\mathbf{F}_{t+p} \in \mathbf{R}^{N \times 12}$ indicate the traffic movement volumes for the entire network at time $t + p$. Each row of the matrix $\mathbf{F}_{t+p}$ indicates all the possible movements (i.e., left, through, and right) at each approach (e.g., four

approaches) of an intersection. Description for all the notations associated with the model development is included in Table 1.

Table1

Description of the Notation associated with the model development

Notation	Description
$\mathcal{G}$	Transportation network
$\mathcal{V}$	Set of nodes in $\mathcal{G}$ with size of $ \mathcal{V} = N$
$\mathcal{E}$	Set of links in $\mathcal{G}$ with size of $ \mathcal{E} = E$
$A \in \mathbf{R}^{N \times N}$	Weighted adjacency matrix of $\mathcal{G}$, defined by Equation (1)
$I \in \mathbf{R}^{N \times N}$	Identity matrix
$\bar{A} \in \mathbf{R}^{N \times N}$	Neighborhood matrix defined by Equation (2)
$\bar{D} \in \mathbf{R}^{N \times N}$	Degree matrix of $\mathcal{G}$, a diagonal matrix where diagonal elements (i, i) indicate the number of links connected with a node
d_{ij}	distance between nodes i and j
l	Input time sequence length (0,1, l)
c	Number of input features
$X_t \in \mathbf{R}^{N \times c}$	Contains all the traffic features associated with each node (i) of the network
g	Graph Convolutional filter to learn the congestion propagation inside the network
$f(.)$	Activation function
$W_{gc} \in \mathbf{R}^{N \times N}$	Learnable parameters for the convolution filter
h	Indicates the outputs from different layers of the proposed neural network architecture
p	Prediction horizon (1, p)
$F_{t+p} \in \mathbf{R}^{N \times 12}$	Intersection level traffic movement volumes for entire network

5

6 **4. Methodology**

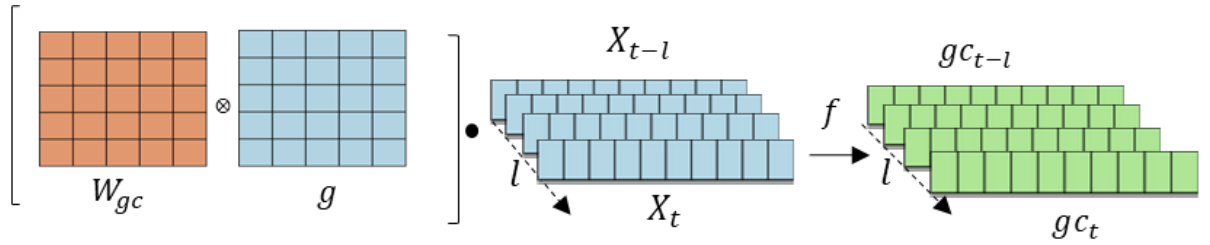
7 *4.1 Graph Convolution for Spatial Dependency Modeling*

8 In a transportation network, the nodes are interconnected so traffic condition at a given node will
9 impact the traffic in the neighboring nodes. This is because of the propagation of traffic flow from

the origin nodes towards the neighboring nodes. Graph convolutional neural network can capture these intrinsic dynamics of the network (i.e., flow propagation), which means it can capture the spatial cross correlation among the neighboring nodes based on their position inside the network (Fig. 1(a)). In a graph convolution approach, we can represent the traffic flow propagation process using a convolutional filter. To derive the convolutional filter, the adjacency matrix is decomposed into its eigenvalues to represent the structural properties of a graph such as strength of a node (e.g. node level features), shortest path between two nodes etc. Such representation while feed into deep learning model suffers from exploding or vanishing gradient problem due to sparsity in eigen values' distribution. To overcome this exploding or vanishing gradient problem, Kipf and Welling (Kipf and Welling, 2016) proposed a normalization technique to represent a graph and its intrinsic dynamics known as spectral graph. Based on spectral graph representation method the convolution operation to capture the spatial correlation can be defined as follows,

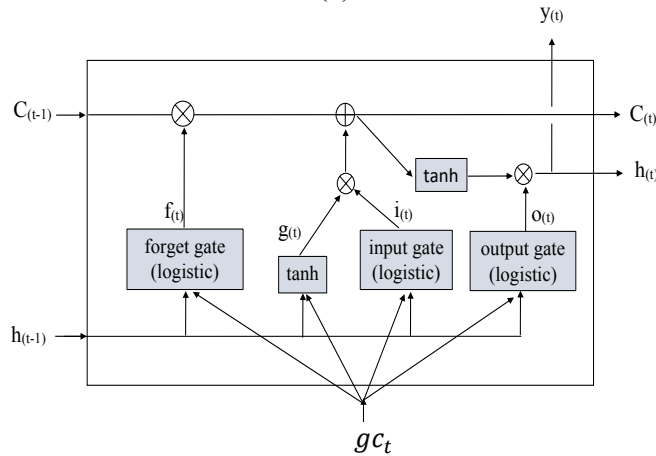
$$gc_t = (W_{gc} \odot \bar{D}^{-\frac{1}{2}} \bar{A} \bar{D}^{-\frac{1}{2}}) X_t \quad (4)$$

where, gc_t indicates the convoluted feature matrix and W_{gc} indicates the parameters for the convolution filter. $\bar{A}$ and $\bar{D}$ indicates the adjacency matrix and degree matrix respectively and the expression $\bar{D}^{-\frac{1}{2}} \bar{A} \bar{D}^{-\frac{1}{2}}$ indicates the symmetrically normalized adjacency matrix. X_t indicates node level features, representing traffic demand variations. The convoluted feature matrix gc_t represents the traffic flow propagation (i.e. network's intrinsic dynamics) inside the network.



g Graph convolutional filter
 W_{gc} Parameters for the convolution filter
 X_t Contains input features for all the nodes
 f Non-linear function
 l Input time sequence length
 p Output sequence length
 $\otimes$ Element wise multiplication

(a)



(b)

Fig. 1 Model Architectures (a) Graph Convolutional Neural Network (b) Long Short-Term Memory Neural Network

4.2 LSTM Model for Temporal Dependency Modeling

In traffic prediction problems, an LSTM (Hochreiter and Uergen Schmidhuber, 1997; Ma et al., 2015; Rahman and Hasan, 2018) model is applied to capture the temporal correlation among traffic features. The basic difference between an LSTM model and other neural network models is that in a simple neural network model, a hidden state is stored as single vector or matrix whereas in a LSTM model the hidden state consists of two vectors a short-term state (h_t) and a long-term state (c_t) (Fig. 1(b)). To capture the short-term correlation (i.e., hourly pattern) between two consecutive timesteps the short-term state (h_t) is recursively fed into each of the consecutive LSTM units (i.e., over the length of input sequences). Moreover, LSTM stores the long-term information (such as period or seasonal traffic variations) as it passes over multiple time steps. At each time step (t) the hidden states (h_t, c_t) are continuously updated by four fully connected neural network layers known as input $i(t)$, forget $f(t)$, cell $g(t)$ and output $o(t)$ layers. In the final time step the hidden states (h_t, c_t) are fed into the output layer to get the final output ($y_{(t)} = h_t$).

4.3 Modeling Frameworks

We proposed two different graph convolution based neural network architectures to model the traffic movement volumes over multiple temporal sequences: Graph Convolutional LSTM and Graph Convolutional Encoder Decoder LSTM. In the following sections, the proposed modeling framework is described,

Graph Convolutional LSTM (GCN-LSTM): In the graph convolutional LSTM architecture, we stack the graph convolution and LSTM layer to capture both spatial and temporal dependency of traffic features. The model takes intersection level traffic features (X_t) such as hourly volume, zonal level trip attraction and production, characteristics of the built environment as inputs and outputs the predicted traffic movement volumes. Fig. 2 shows different components of the model as it unrolls over multiple time steps (i.e., length of input sequences). At each time step, the model performs a graph convolution operation over the input data (G_t) and feeds the output (gc_t) into the LSTM model. After taking the input from the graph convolution (GC) layer, it updates the hidden states (h_t, c_t). In this architecture, the short-term state (h_t) at the final time step (t) is linearly transformed using a fully connected layer to get the predicted (y_t) traffic movement volumes over multiple sequences.

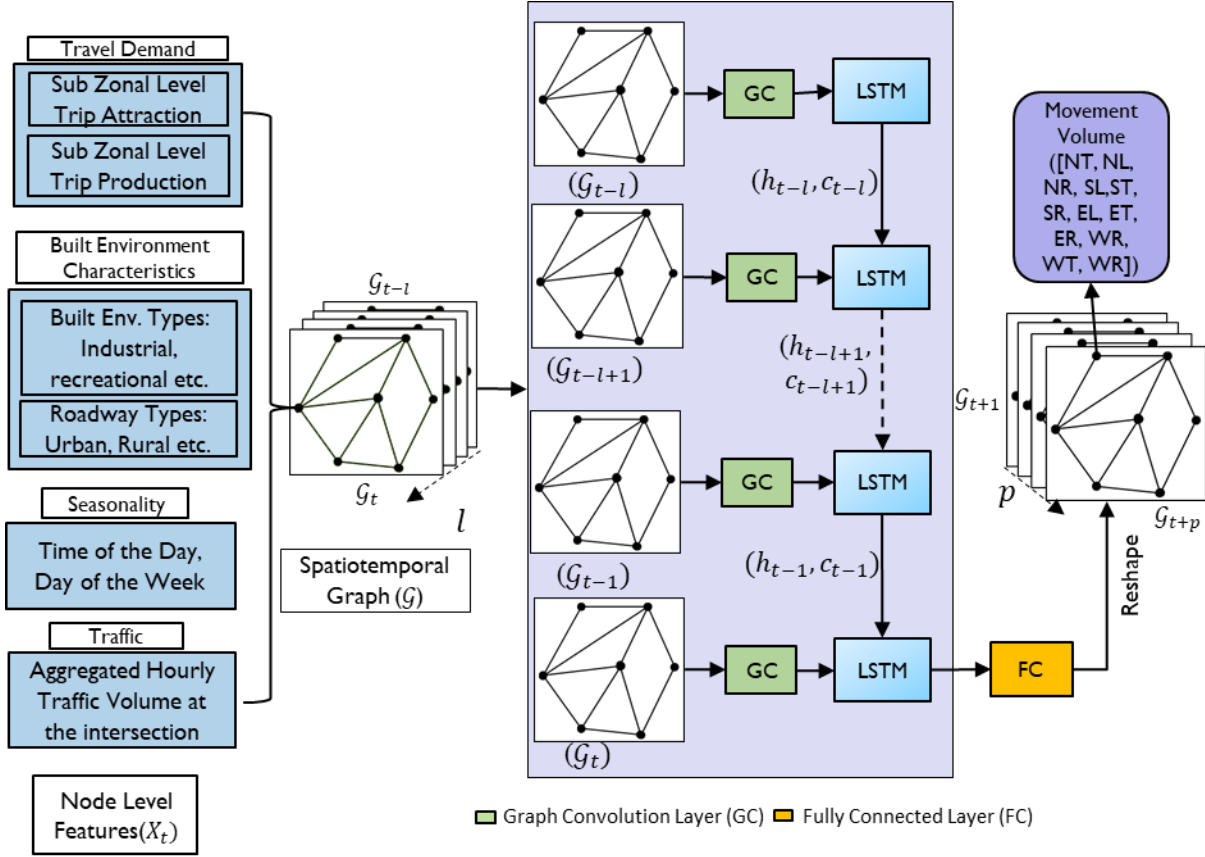


Fig. 2 Modeling framework and information flow diagram for different components for Graph Convolutional LSTM

Graph Convolutional Encoder Decoder LSTM (GCN-Encoder-Decoder): In this architecture, the outputs from the graph convolutional layer are fed into an *encoder-decoder LSTM* architecture. Similar to the previous architecture, the encoder LSTM captures the short-term and long-term dependencies of traffic features as it unrolls over multiple time step, thus it updates the hidden states (h_t, c_t) . However, to get the final outputs over multiple temporal sequences, we use a decoder LSTM (Fig. 3). At the first-time step, the decoder LSTM takes the hidden states (h_t, c_t) and input state (gc_t) from the final encoder LSTM step (t) . In this architecture, we modify the traditional encoder decoder LSTM to pass the input state (gc_t) at time step (t) along with hidden state of encoder LSTM. We encode this input as a vector which indicates the starting point of future time series prediction. In the following time steps, it recursively takes the predicted traffic volumes (F_{t+1}) and hidden states (h_{t+1}, c_{t+1}) from the previous step to predict the traffic volumes in the next step (F_{t+2}) . The decoder LSTM iteratively performs this operation to generate the whole output sequence.

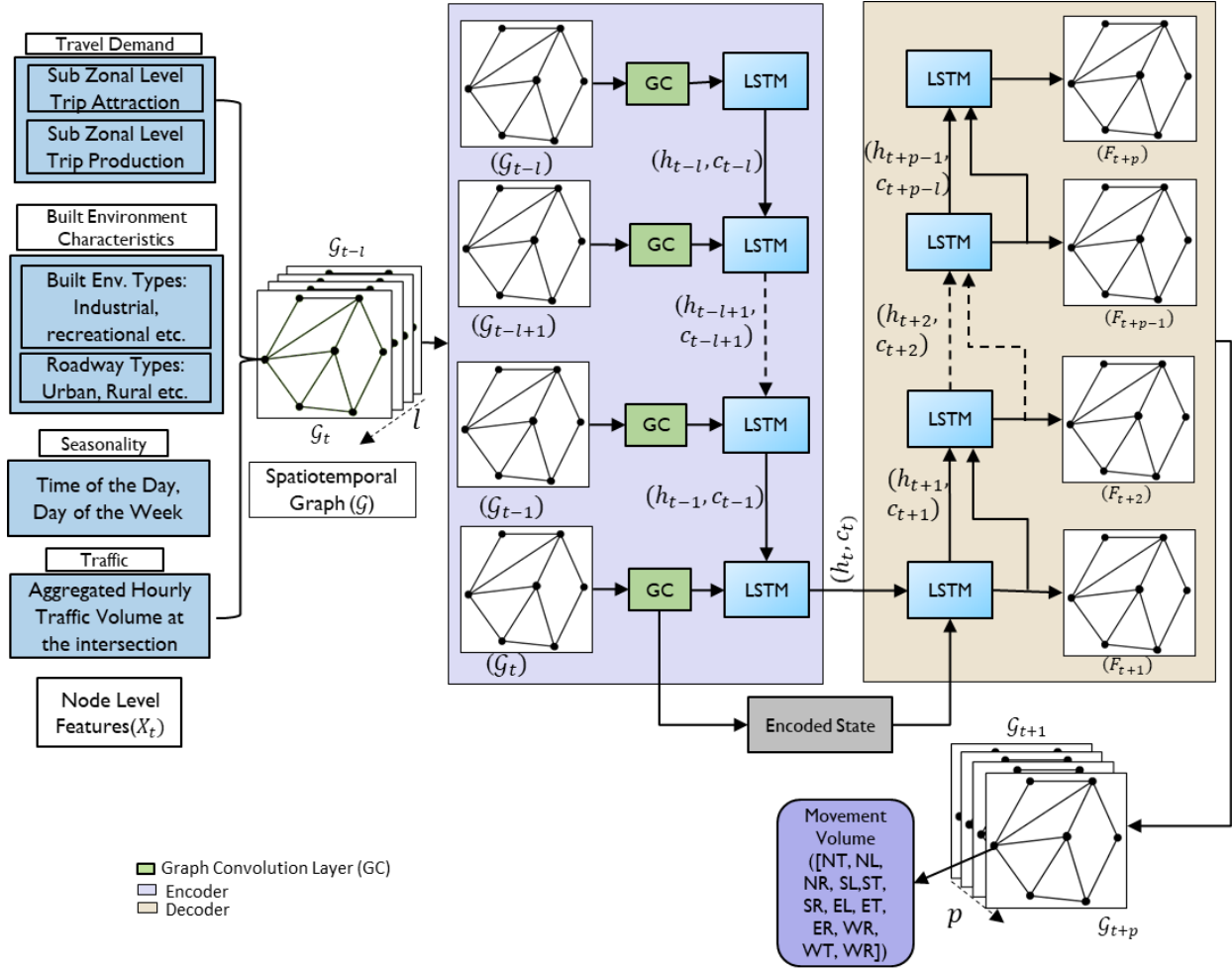


Fig. 3 Modeling framework and information flow diagram for different components of Graph Convolutional Encoder Decoder LSTM

5. Data Collection and Processing

5.1 Traffic Movement Volumes

In Seminole County, majority of the signalized intersections are equipped with advanced traffic signal controllers on the arterials and each signal provides Automated Traffic Signal Performance Measures (ATSPM) (Fig. 4), which includes traffic movement volumes, signal timing, traffic queue volume etc. We extracted high-resolution event-based signal data for Seminole county from January 1, 2016 to December 31, 2016; in total we have collected data for 253 intersections. From the raw dataset, we mainly extracted the traffic movement volumes for different direction and movement types.

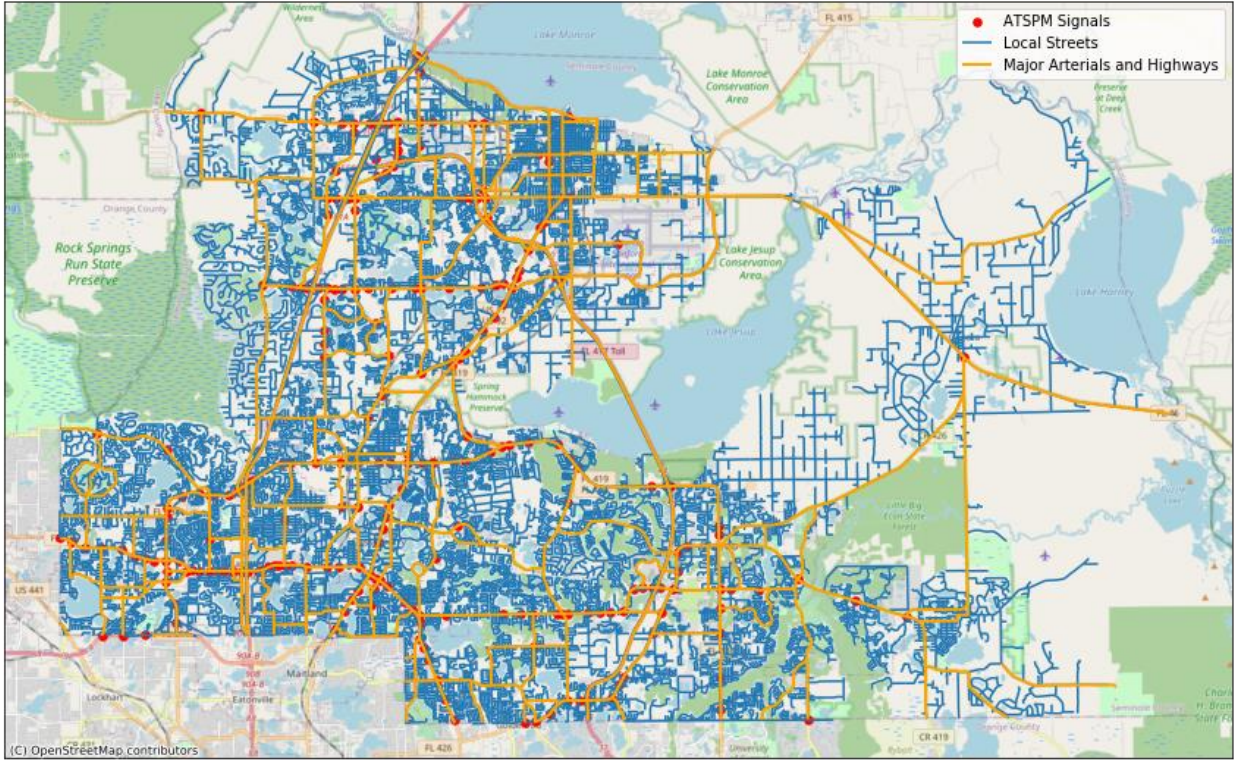


Fig. 4 The study area (Open Street Map)

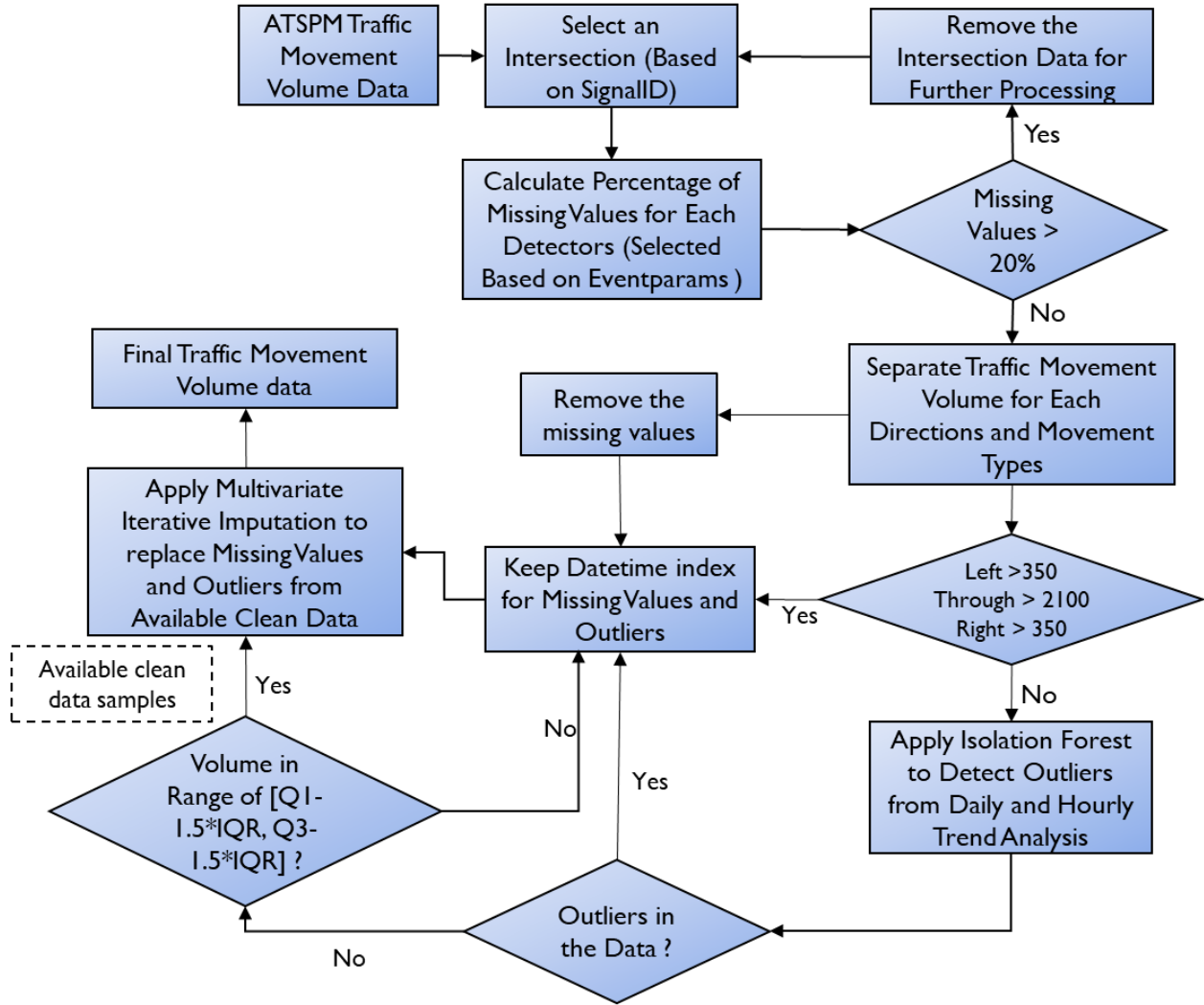
The raw data collected from traffic detectors are subjected to errors. Several factors such as detector's malfunctioning, false encoding during storing the data into the server, overlapping of multiple entries, duplicate entries, bad weather conditions etc. can cause errors. Therefore, before proceeding to any data analysis, we need an extensive data cleaning and quality checking. Fig. 5 shows the framework for the data processing steps.

To check the quality of the data, we followed several steps starting with checking the percentage of missing data for different detectors. We consider the detectors having a higher percentage of missing values as unreliable detectors. Moreover, data imputation is not feasible for the detectors with too many missing values as it will produce unrealistic data distribution. Considering this issue, we retain the detectors having missing values less than 20% of total data samples. We apply three techniques to check the quality of the data and detecting outliers.

First, we compare vehicle per hour for each movement types (i.e., left, through, right) with Federal Highway Administration (FHWA) guidelines on capacities at signalized intersection. According to the FHWA guidelines for left and right turn movement the capacities vary from 150-350 veh/hr, while for through movement the capacity varies from 1600-2100 veh/hr depending on number of available lanes (FHWA, 2004). We observe the distribution of the hourly turning movement volume for different movement types. Almost all the data samples, except a few (less than 0.1%), have hourly volume less than capacity. We consider the samples with values greater than capacity as outliers.

Second, we apply isolation forest algorithm (Liu et al., 2012) to detect outlier based on temporal pattern of traffic volume for each movement types. The algorithm learns traffic pattern with respect

1 to variations in hours of the day and days of the week and isolates the outliers which show unusual
2 pattern.



Q1: First quartile, Q3: Third quartile, IQR: Interquartile Range ($Q3 - Q1$)

Fig. 5 Data processing framework

Third, we check if the turning movement volume remains within the range between $(Q1 - 1.5 * IQR)$ and $(Q3 + 1.5 * IQR)$, where $Q1$ and $Q3$ indicates the first and third quartile and IQR indicates the interquartile range ($Q3 - Q1$).

Finally, we use a technique known as multivariate iterative imputation (van Buuren and Groothuis-Oudshoorn, 2011) adapting Bayesian ridge regression as an estimator to impute the missing values and outliers. To fit the estimator, we use time of the day (hour), day of the week (day), and volume with missing values as inputs. For each imputation the algorithm takes a sample from gaussian posterior of the fitted estimator. We use Python scikit learn (Pedregosa et al., 2011) library to implement the algorithm. The details about the data imputation algorithm are provided in reference (van Buuren and Groothuis-Oudshoorn, 2011). Fig. 6 shows the distribution of through movement volumes before and after data processing. After cleaning the data, we have in total 196

intersections in our final data set. Moreover, we only keep the data in between 6 am (Early Morning) and 12am (Midnight), most of cases from 12 am to 6 am the traffic activity within this region is either zero or nearly zero. We have also removed the data for special events and holidays such as hurricane evacuation period (September 28, 2016 – October 10, 2016), thanksgiving holiday etc.

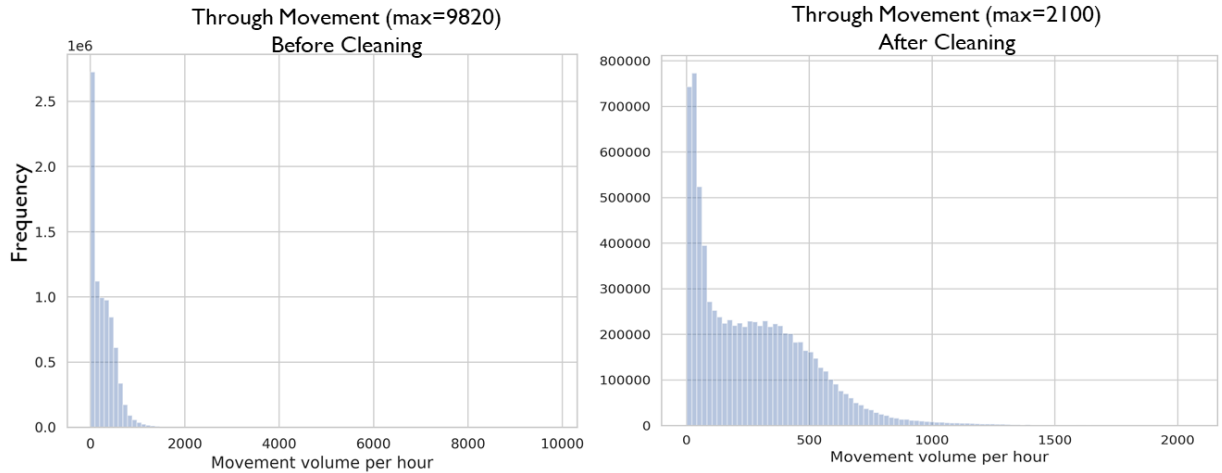


Fig. 6 Distribution of data samples based on vehicles per hour per lane

5.2 Travel Demand Features

To generate the travel demand for the study area, we employ Central Florida Regional Planning Model (CFRPM 6.1) which outputs average weekday trip production and trip attraction at the Traffic Analysis Zone (TAZ) level by trip purpose and special generators. Our study area (Seminole County, Florida) includes 230 TAZs, so, we can aggregate the weekday trips to estimate total production and total attraction for each of these 230 zones. However, each TAZ has multiple intersections within its boundary. Hence, to link this demand information with operational characteristics of an intersection, we need to partition these TAZs into a finer spatial resolution.

Considering this issue, we develop a technique to partition the TAZs into multiple subzones to generate demand at a finer resolution.

We considered multiple approaches for creating the sub-zones. We determine the number of subzones based on spatial distribution (196 intersections sparsely distributed over 230 TAZs) of intersections. This will ensure an adequate representation for each TAZ while extracting the spatial features. We also considered employability of the approach by transportation agencies; hence we adopt a stable process to divide the TAZs. Based on these considerations, we divide each TAZ into 5 subzones which ensures an adequate representation of each TAZ.

We use Geographic Information System (GIS) software to create the subzones for each TAZ. The TAZ shape file is a polygon shape file in GIS which is usually constructed by joining multiple points in the perimeter. To identify the points to be connected, we followed the following process:

- First, we convert the polygon TAZ shapefile to point shape file using GIS. For example, if a given TAZ shape file contains total 100 points in the perimeter, we alter the TAZ shape file to the corresponding 100 points.

- Second, we divide the number of points into 5 groups. If the number is not divisible by 5, then we assign all the remaining points in the last group. For example, with 100 points, we will have 5 groups of 20 points each. If we have 101 points, then the last groups will have 21 points. Also, it is important to note that the last point of a group will be the first point of the subsequent group. So, for 100 points, the groups will be: 1-20; 20-40; 40-60, 60-80 and 80-100.
- Finally, within each group, we identify the first and last point and connect both these points with the centroid, thus converting one single TAZ to 5 sub-TAZs.

In total we create 1150 subzones from 230 TAZs. Fig. 7 shows the TAZs and generated subzones for the study area. From the CFRPM model we estimate average weekday trip attraction and production at a subzone. Afterwards, we expand the daily demand into hourly demand using hourly distribution factors provided by CFRPM (see (FDOT, 2014) for more details).

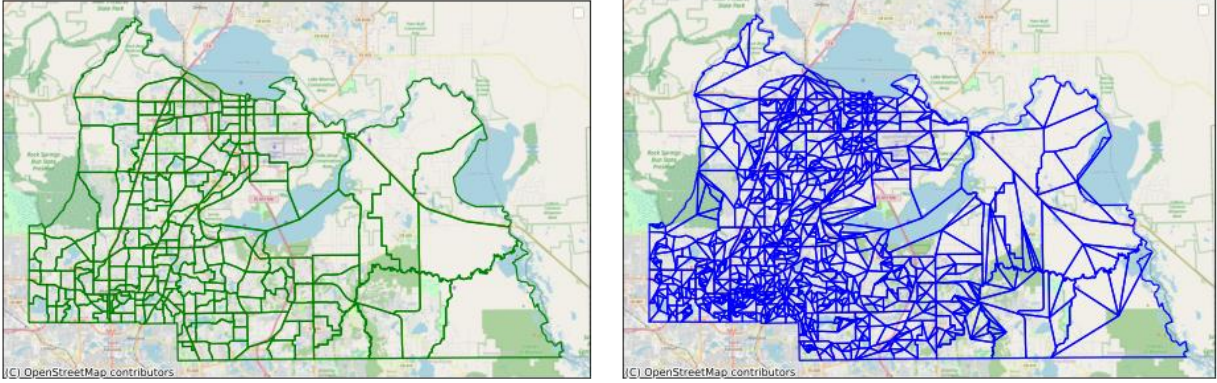


Fig. 7 Traffic Analysis Zones and Subzones for the Analysis

We have also extracted exogenous variables including built environment and land-use characteristics for each subzone. Built environment characteristics variables are processed from NAVSTREET data and include number of restaurants, shopping centers, business centers, entertainment establishments and educational institutions. Land-use characteristics are processed using high resolution parcel level land-use data sourced from Florida Department of Revenue. Each parcel is assigned a unique ID (Parcel ID) linking it with equivalent parcel level attribute information such as property/feature value, land value, land area in square feet, land-use codes

(DOR-UC), owner name, owner address, physical address, physical zip code, building details and so on contained in the Name-Address-Legal (NAL) file. In this study we consider five land use categories: Residential, Retail/Office, Industrial/Manufacturing, Institutional/Infrastructure, Recreational etc.

6. Experiments

6.1 Input features and Graph Representation

In this section we discuss about spatiotemporal features extraction technique from the data. We also discuss about the graph representation from geolocation of the intersections.

Input features in the data samples (X_t): We have extracted different types of features to represent the travel demand and built environment characteristics corresponding to each intersection. Since the spatial variations in built environment characteristics are more subtle, we use natural logarithm of the areas (acre) corresponding to different built environment types (Table 2), which scales the subtle changes and make it more prominent. Finally, we merge these variables with hourly trip attraction and trip production. To feed the model with short term travel demand variations, we use aggregated intersection level hourly traffic volume. We also use temporal features such as time of the day and days of the week to capture the seasonality inside the data.

In total we have 11 features; we formulate the traffic data sample as [number of samples (n), input time sequence (l), number of nodes (N), input features (c)]. Since, we have the data from 196 intersections, so the number of nodes, $N = 196$. We select 6-hour input data sequence to predict traffic for next 4 hour, so input time sequence length $l = 6$ and prediction horizon length, $p = 4$. In total we have $c = 11$ input features: days of the week (i.e. Saturday, Sunday etc.), time of the day as hour (1 to 24 hr), intersection specific aggregated traffic volume, hourly trip attraction, hourly trip production, proportion of highway among total roadways of a zone. We have 5706 data samples from a one-year period (2016). Finally, the input data and target data samples have the shape as [5706, 6, 196, 11] and [5706, 4, 196, 12], respectively.

Table 2. Description of Input Features

Variables	Description
t_d	Days of the week (i.e. Saturday, Sunday etc.)
t_{hr}	Time of the day as hour
q_t	Intersection specific aggregated traffic volume at time t
T_A	Hourly Trip Attraction
T_P	Hourly Trip Production
HW_{prop}	Proportion of Highway among total roadways of a zone
B_{ind}	Ln(industrial)
B_{rec}	Ln(recreational)
B_{ret}	Ln(retail/office)
B_{inst}	Ln(institutional)
B_{res}	Ln(residential)

Graph Representation: We follow several steps to construct the graph. First, we map the signals' locations into Open Street Map. Second, considering the signalized intersections as nodes we connect each of them with their neighboring nodes. To find the neighboring nodes, we select an

origin node and find the shortest path from the origin node to all the other nodes. The nearest node on each of these shortest paths is the neighboring node to the origin. Finally, after constructing the network, we represent the network using adjacency matrix (see equation 1 and 2). We use the travel distance as weight for the graph adjacency matrix. We also perform gaussian transformation on the weighted graph adjacency matrix,

$$A(i, j) = \begin{cases} \exp\left(-\frac{d_{i,j}^2}{\sigma^2}\right), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where, σ indicate the standard deviation of distances.

6.2 Baseline Models

We implement three baseline models to compare the performance of the proposed graph convolution-based models.

Historical Average (HA): In the Historical average method we compute intersection specific the average traffic volume for each direction (i.e., Left, Through, Right) and each hour (i.e., 1 to 24 hour). We estimate this average based on the training data. So, if we want to forecast traffic for a given intersection at 2pm then we will take the average of all the traffic at 2pm from the historical data.

LSTM: In the *LSTM* model, we use single *LSTM* layer to predict traffic for next 4 hours. In the hidden layers we assign 196 (number of nodes) hidden neurons. The output layer is a fully connected layer with tanh activation function.

Convolutional LSTM (Conv-LSTM): In the Convolutional LSTM (*Conv-LSTM*) model we stacked a convolution layer with LSTM layer. Convolutional layer use convolution filter to extract the spatial correlation among traffic features in among neighboring intersections. We experiment with different size of the kernel (k) and find that the model performs best for a kernel size of 3. The output from the convolutional layer is feed into the LSTM layer to capture temporal correlation among traffic features while predicting traffic flow over a temporal sequence.

6.3 Model Training

We use 80% of the data for training (learning the parameters), 10% for validation (tuning hyper-parameters), and rest 10% of the data is used for testing (checking performances) the model. Based on the validation accuracy we tune the hyper-parameters such as learning rate, types of activation functions (i.e., tanh, sigmoid etc.), maximum number of iterations. Once the final model parameters are fixed, we test it on the test data set.

We train the model using mean squared error (MSE) as the loss function. At each iteration, the model estimates the MSE for the estimated traffic movement volume ($\hat{F}_{t+p}^{ij}$) and the actual movement (F_{t+p}^{ij}) volume for a given intersection (i), and movement types (m). Afterward, the gradient of the loss function is backpropagated to adjust the weights to reduce loss function value.

The loss function can be defined as:

$$L_m = \text{Loss}(F_{t+p}^{mi}, \hat{F}_{t+p}^{mi}) \quad (6)$$

$$MSE = \sum_{p=1}^p \frac{1}{N} \sum_{i=1}^N \frac{1}{M} \sum_{m=1}^M (F_{t+p}^{im} - \hat{F}_{t+p}^{im}) \quad (7)$$

where, $Loss(.)$ is the function to estimate the error between the actual (F_{t+p}^{im}) and estimated values ($\hat{F}_{t+p}^{im}$). i and m denotes the intersections (i.e., nodes) and movement types (i.e., NL (northbound left), NT (northbound through), NR (northbound right) etc.).

We implement the model using pytorch library ("PyTorch," 2016) and train it in Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-62-generic x86_64) supported by a cluster of four NVIDIA RTX 2080Ti 11 GB GPUs. While training the model, we track the training and validation loss values to check whether the model is overfitting or not. From the loss values, we find that it takes about 100 epochs with a learning rate of 0.001 for the *GCN-LSTM* model to converge (i.e., similar train and validation loss value) (Fig. 8). Moreover, after 100 epochs, the value of the loss function for the validation data gradually starts increasing, indicating that the model starts to overfit. We use Adaptive Moment Estimation (ADAM) to train the model. Compared to other optimizers such as Adaptive Gradient (AdaGrad), Root Mean Square Propagation (RMSProp) etc., ADAM optimizer gives more stable solutions, which means MSE values for train and validation data almost remain same after convergence.

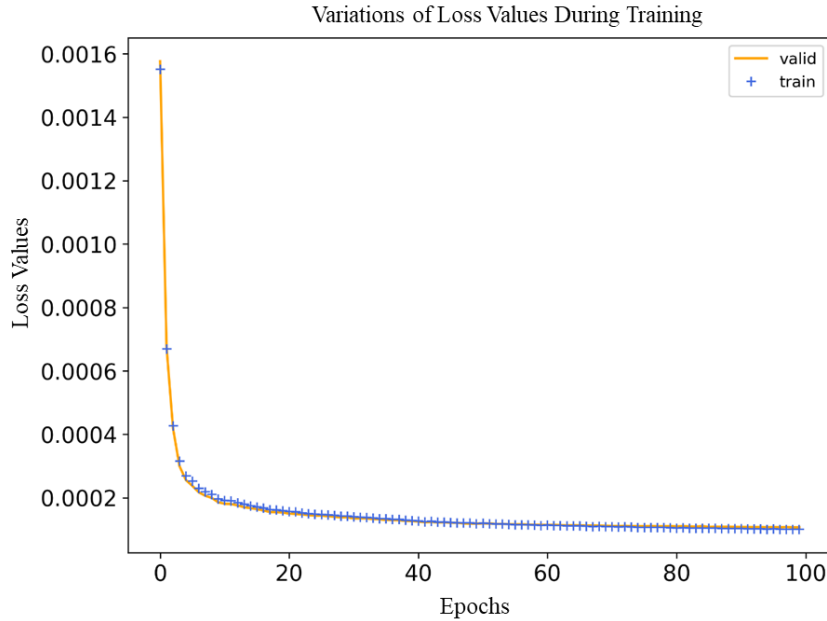


Fig. 8 Variations of training and validation loss values *GCN-LSTM* model

We also compare the training time for different models. Compared to other models, the *GCN-LSTM* model takes less training time. The reason is that for other models it takes more than 100 epochs to converge to a stable solution. Moreover, on average the *GCN-LSTM* takes less time per epoch (6.16 seconds) compared to the *Conv-LSTM* (8.19 seconds) and the *GCN-Encoder-Decoder* (7.19 seconds).

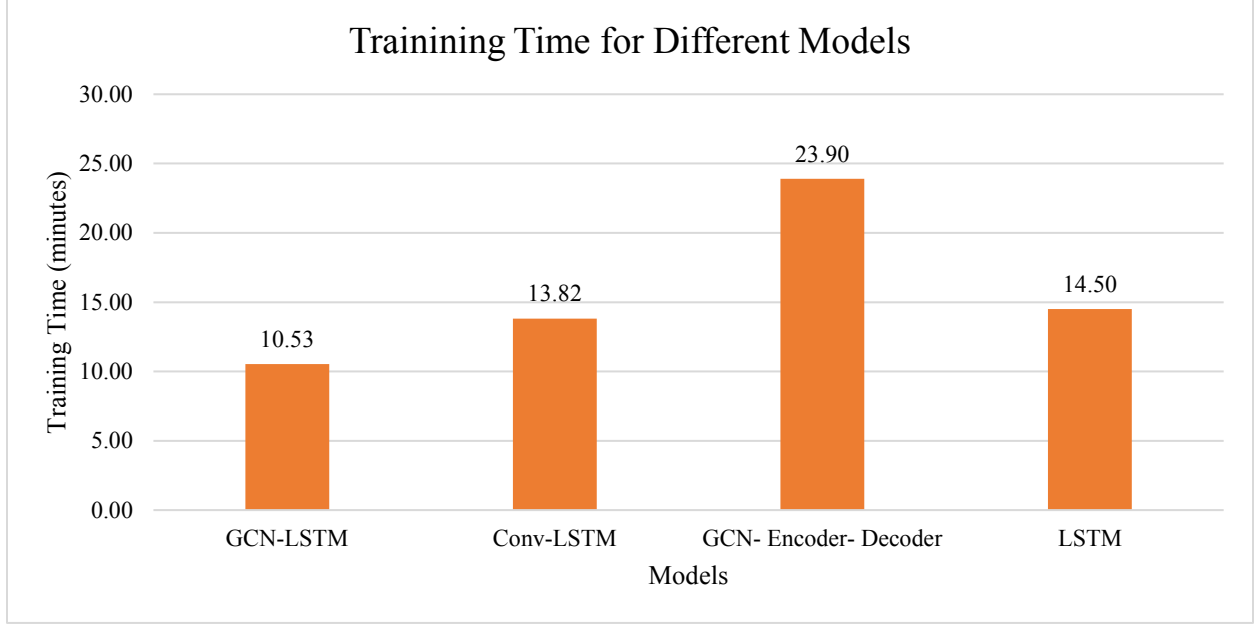


Fig. 9 Overall Training time for different models

6.4 Experiment results

Once the final model is fixed, we test it on the test data set. We calculate Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and GEH (Transport for London, 2010) statistics as performance measures to check the accuracy of the implemented model. Performance metrics are defined as,

$$RMSE = \sqrt{\sum_{p=1}^p \frac{1}{N} \sum_{N=1}^N \frac{1}{M} \sum_{M=1}^M (F_{t+p}^{im} - \hat{F}_{t+p}^{im})^2} \quad (8)$$

$$MAE = \frac{1}{p} \sum_{p=1}^p \frac{1}{N} \sum_{N=1}^N \frac{1}{M} |F_{t+p}^{im} - \hat{F}_{t+p}^{im}| \quad (9)$$

$$GEH = \sqrt{\frac{2(\hat{F}_{t+p}^{im} - F_{t+p}^{im})^2}{\hat{F}_{t+p}^{im} + F_{t+p}^{im}}} \quad (7)$$

where, F_{t+p}^{im} and $\hat{F}_{t+p}^{im}$ indicate actual and predicted movement volumes for each intersection (i) and movement types (m) at time ($t + p$).

In Table 3, we report the performance of model on the test data. To test the sensitivity of the model over different data samples, we randomly split the data to generate 10 different train, test and validation datasets. Finally, we train 10 different models and report the mean and standard deviation of the estimated performance measures on the test data sets. Based on performance measures, we find that the proposed *GCN-LSTM* model performs best compared to other baseline models. The RMSE and MAE values of the model for through traffic movement are 59.27 and

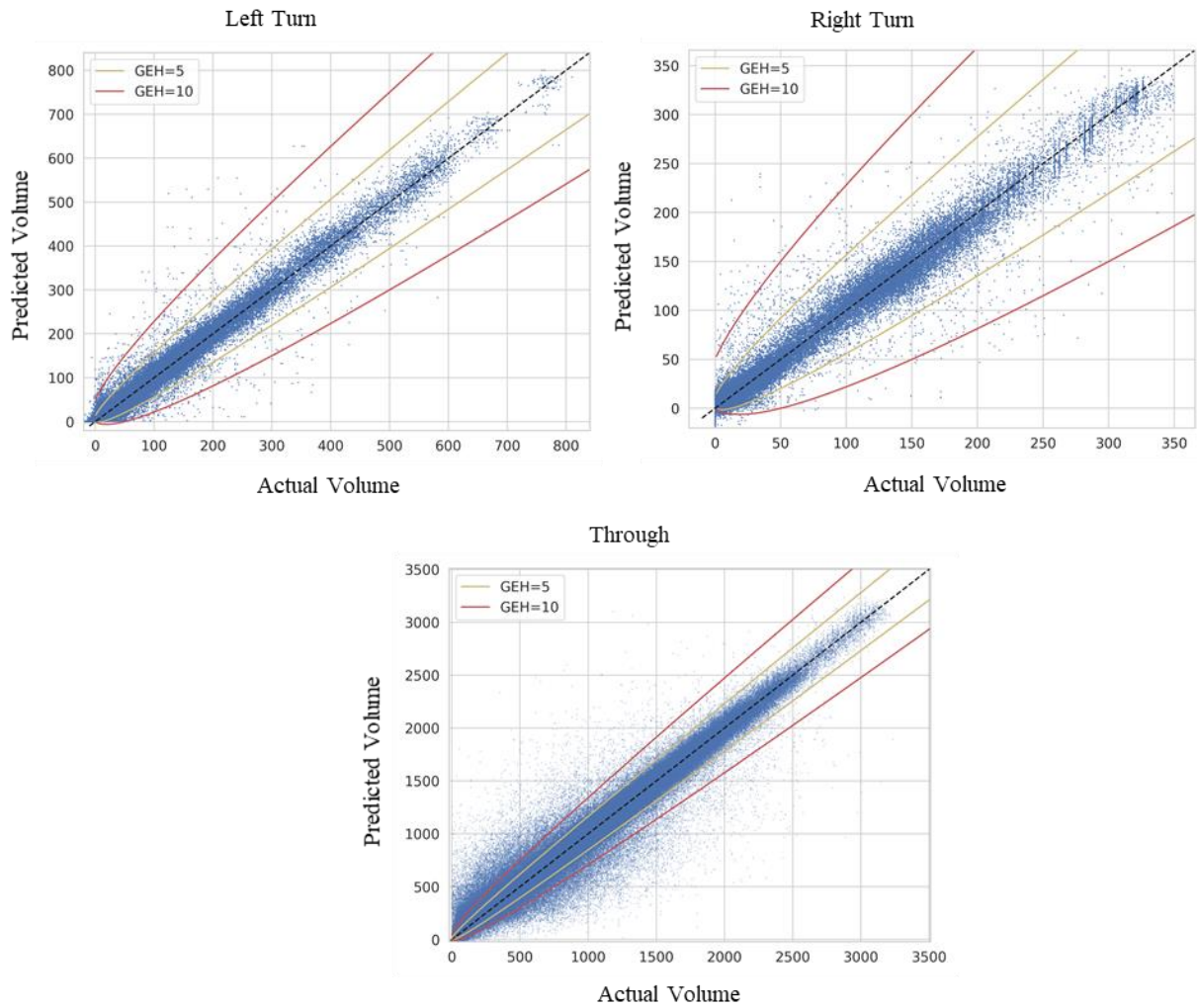
23.072, respectively. However, RMSE and MAE provide aggregate information (average over all the outputs) on the performance of the models, hence, we also estimated R^2 score. As shown in table 3, the R^2 score for all the movement types of the proposed model is above 0.98 indicating that the model can learn the traffic movement patterns very well (Fig. 9 (a)).

Table 3

Comparisons among different models to predict traffic volume over 4-hour sequence for different movement types

Model		RMSE			MAE			R ² Score		
		Left	Through	Right	Left	Through	Right	Left	Through	Right
HA	Mean	7.597	134.369	3.453	1.004	60.72	0.341	0.948	0.927	0.961
	Std	-	-	-	-	-	-	-	-	-
LSTM	Mean	5.768	89.784	3.457	1.143	40.584	0.579	0.966	0.967	0.961
	Std	0.343	6.885	0.191	0.098	3.734	0.098	0.004	0.005	0.004
GCN-Encoder-Decoder	Mean	4.888	82.283	2.855	0.913	32.462	0.488	0.976	0.972	0.973
	Std	0.258	3.309	0.141	0.128	0.899	0.112	0.003	0.002	0.002
Conv-LSTM	Mean	4.177	61.145	2.572	1.289	23.832	0.926	0.982	0.985	0.978
	std	0.114	1.945	0.074	0.115	0.623	0.119	0.001	0.001	0.001
GCN-LSTM	Mean	4.018	59.372	2.458	1.189	23.072	0.830	0.983	0.985	0.980
	std	0.062	0.706	0.065	0.148	0.227	0.155	0.001	0.000	0.001

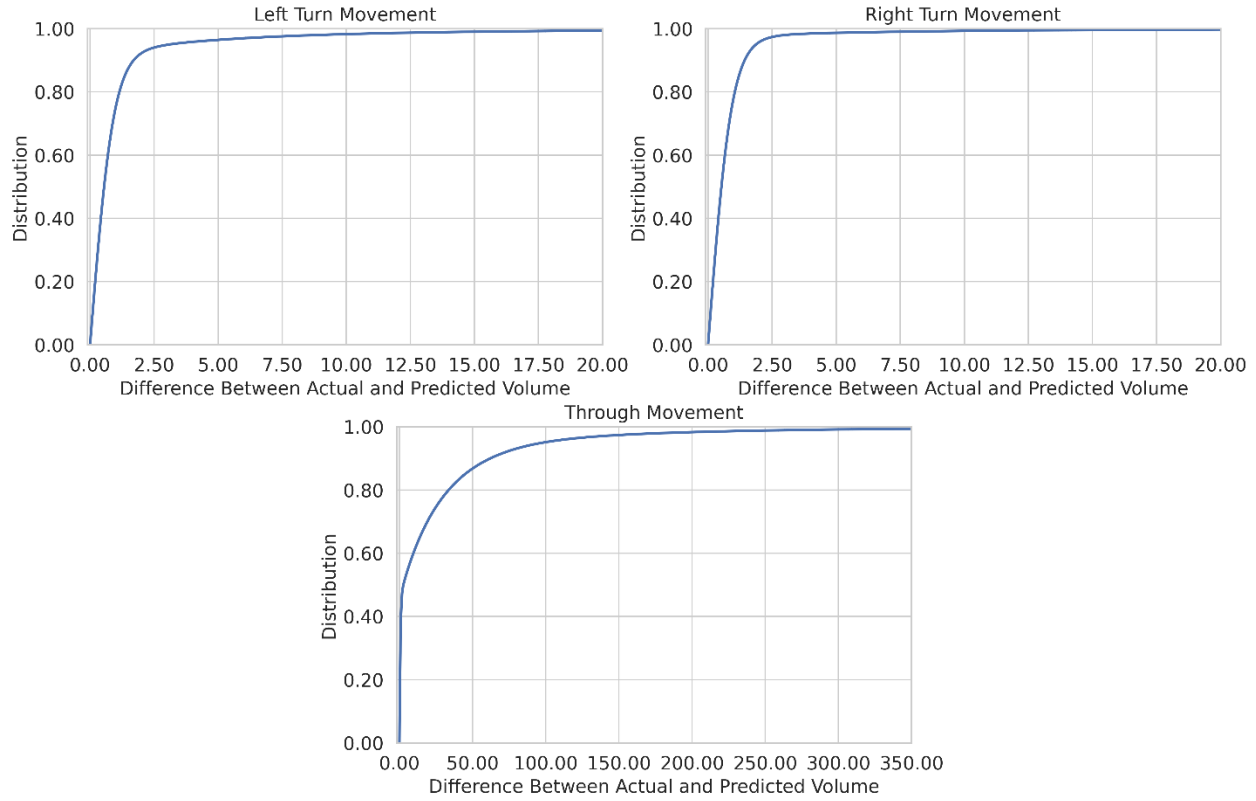
We also estimate the absolute difference between the actual and predicted traffic volume. Fig. 9 (b) shows the cumulative distribution function (CDF) value of absolute errors for all the movement types. For left and right turn movement, more than 95% of the predicted volumes have an absolute error less than 5, whereas for through movement about 90% of the predicted volumes are less than 60. This is because through movement volumes are higher compared to left and right turn movement. To provide a normalized value for accuracy measure we also estimate GEH score. A GEH score of less than 5 is considered as a good match between actual and predicted volumes. According to (Nezamuddin et al., 2011) 85% of the predicted volumes should have a GEH score less than 5, while In our case more than 90 % cases GEH score is less than 5 (Fig. 9 (a)).



1

2

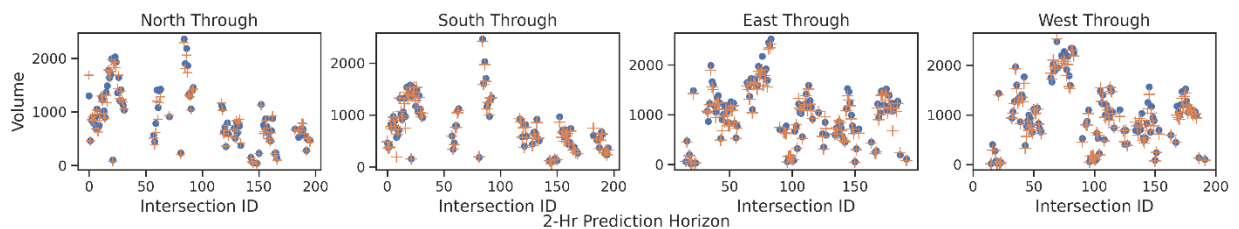
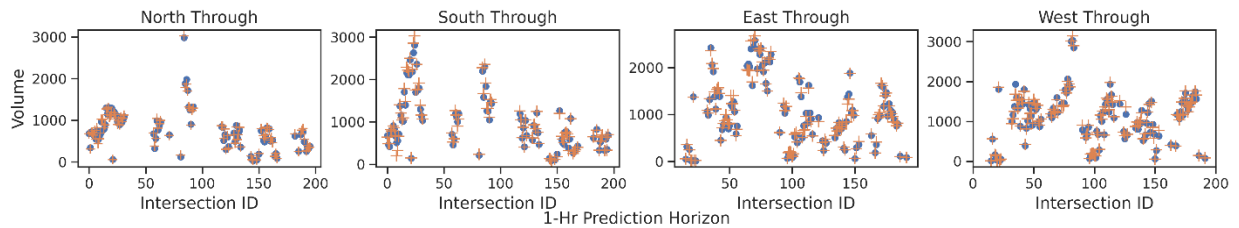
(a) Comparison between actual and predicted traffic volume



(b) Cumulative distributions of absolute different between actual and predicted traffic volume

Fig. 9 Performance of the model in turning and through movement prediction

In Fig. 10 and 11 we show the distribution of through movement types for all the intersection at a given weekday and weekend time periods (i.e. four-hour sequence prediction). For a few intersections we do not have all the movement types, in those cases the model predicts the volume as zero or close to zero (<1). So, the figures include values for a valid movement type of an intersection based on the actual data. From the figures we find that incase of both weekday and weekend the model captures the distribution of traffic movement volumes very well.



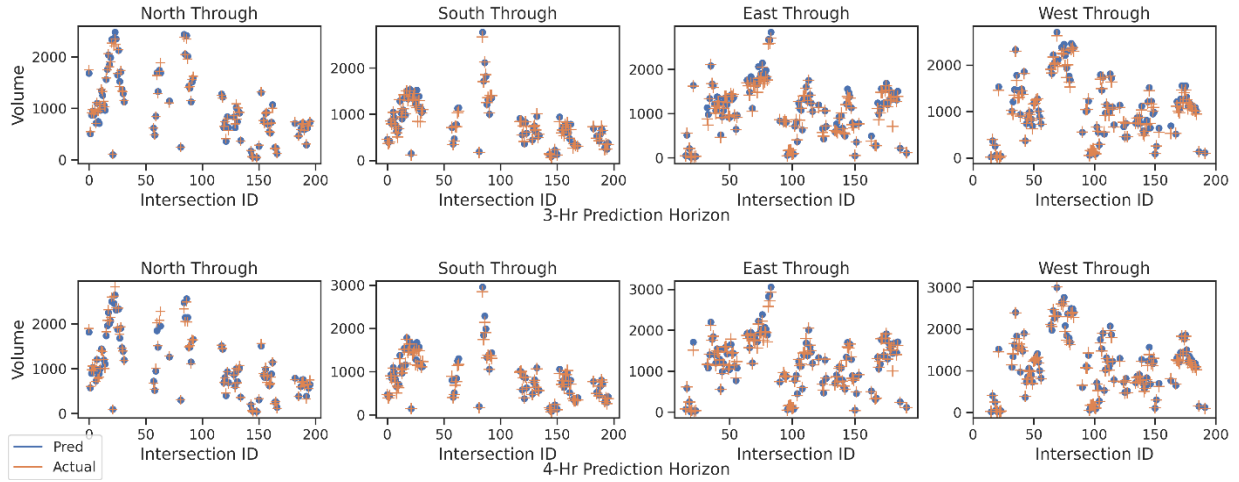


Fig. 10 Predicted through movement volumes for all intersections for a given weekday time period (timestamp 08-10-2016 2 pm – 6pm)

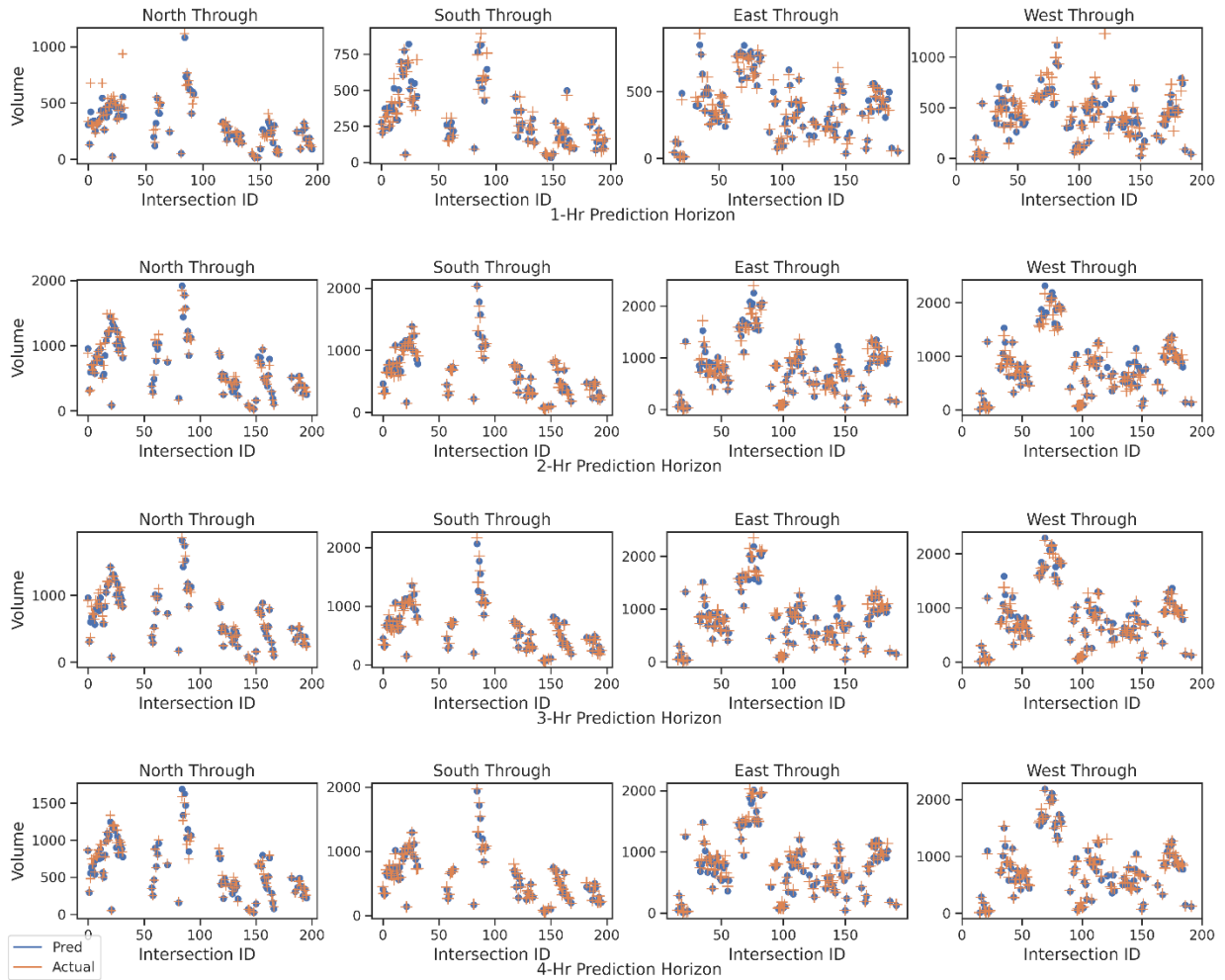


Fig. 11 Predicted through movement volumes for all intersections for a given weekend time period (timestamp 2016-11-13 2 pm – 6pm)

7. Conclusion

Accurate traffic forecasting at the scale of a network is critical to ensure proactive decision making and optimal action plans for traffic operations and management. However, traditional network models involve complex mathematical or simulation-based approaches, which require higher computational time in predicting traffic at the level of an intersection including turning volume. In this study, we develop an alternative data-driven approach adopting the concept of graph convolution. We propose two different modeling frameworks *GCN-LSTM* and *GCN-Encoder Decoder* to predict hourly traffic volume over multiple time steps. From the model results, we find that *GCN-LSTM* model outperforms other baseline models. The overall R^2 value of the model is close to 1 indicating that the model captures the traffic movement volumes very well. Moreover, the absolute difference between actual and predicted volumes are quite low ($GEH < 5$); for right turn, through and left turn movement RMSE values are 4.02, 59.37, and 2.47 respectively. We also compare the efficiency of the model in terms of required training time. Based on the comparison results, the proposed *GCN-LSTM* model takes less training time (10.53 min), being more efficient compared to traditional deep learning models.

Another benefit of the proposed model is that traditional deep learning methods follow a black box approach to select the number of parameters and most of the cases it requires more manual effort. Whereas, in our case we develop the model architecture automatically based on the structure of the network. Even in case of large networks we can automate this process by dividing a large network into small sub networks.

Although this study provides viable solution to model and predict real time network traffic volumes at a higher accuracy, one of the limitations of this study is that we do not use any information on real-time variations of travel demand, rather we use hourly demand which is constant for all the days. High resolution demand data from emerging technologies such as mobile phone sensors or connected vehicles data can be used to overcome this issue. A comparison between traditional demand data and emerging data might give us more insights on model performance.

Moreover, in this study, we only explore the two variants of graph neural networks Graph Convolutional LSTM and Graph Convolutional Encoder Decoder. However, future research should explore more variants of graph neural network such as Graph Attention Networks (Veličković et al., 2018) and GraphSage (Hamilton et al., 2017) based neural networks to improve the overall accuracy of network-wide traffic movement volume prediction.

ACKNOWLEDGMENT

This study was supported by Florida Department of Transportation's District 5 office and the U.S. National Science Foundation through the grant CMMI #1917019. However, the authors are solely responsible for the facts and accuracy of the information presented in the paper.

AUTHOR STATEMENT

The authors confirm contribution to the paper as follows: study conception and design: R. Rahman, S. Hasan; analysis and interpretation of results: R. Rahman, S. Hasan, J. Zhang, T. Bhowmik, S. Dey, I. Jahan, N. Eluru; draft manuscript preparation: R. Rahman, S. Hasan, J. Zhang, T.

Bhowmik, S. Dey, N. Eluru; funding acquisition: S. Hasan, N. Eluru. All authors reviewed the results and approved the final version of the manuscript.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

References

- Alajali, W., Zhou, W., Wen, S., Wang, Y., 2018. Intersection traffic prediction using decision tree models. *Symmetry (Basel)*. 10, 1–16. <https://doi.org/10.3390/sym10090386>
- Bao, J., Liu, P., Ukkusuri, S. V., 2019. A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data. *Accid. Anal. Prev.* 122, 239–254. <https://doi.org/10.1016/j.aap.2018.10.015>
- Billings, D., Jiann-Shiou, Y., 2006. Application of the ARIMA Models to Urban Roadway Travel Time Prediction-A Case Study. *Systems, Man and Cybernetics, 2006. SMC'06. IEEE Int. Conf.* 2529–2534.
- Bogaerts, T., Masegosa, A.D., Angarita-Zapata, J.S., Onieva, E., Hellinckx, P., 2020. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transp. Res. Part C Emerg. Technol.* 112, 62–77. <https://doi.org/10.1016/j.trc.2020.01.010>
- Chang, G.L., Su, C.C., 1995. Predicting intersection queue with neural network models. *Transp. Res. Part C* 3, 175–191. [https://doi.org/10.1016/0968-090X\(95\)00005-4](https://doi.org/10.1016/0968-090X(95)00005-4)
- Chen, L., Vonu, P., Konstantinos, T., 2021. Short - Term Prediction of Demand for Ride - Hailing Services : A Deep Learning Approach. *J. Big Data Anal. Transp.* 3, 175–195. <https://doi.org/10.1007/s42421-021-00041-4>
- Cui, Z., Henrickson, K., Ke, R., Wang, Y., 2020a. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *IEEE Trans. Intell. Transp. Syst.* 21, 4883–4894. <https://doi.org/10.1109/TITS.2019.2950416>
- Cui, Z., Lin, L., Pu, Z., Wang, Y., 2020b. Graph Markov network for traffic forecasting with missing data. *Transp. Res. Part C Emerg. Technol.* 117, 102671. <https://doi.org/10.1016/j.trc.2020.102671>
- Deshpande, M., Bajaj, P.R., 2016. Performance analysis of support vector machine for traffic flow prediction. *2016 Int. Conf. Glob. Trends Signal Process. Inf. Comput. Commun.* 126–129. <https://doi.org/10.1109/ICGTSPICCC.2016.7955283>
- FDOT, 2014. Central Florida Regional Planning Model (CFRPM) Version 6.1 [WWW Document]. URL https://www.fsutmsonline.net/index.php?/model_pages/comments/cfrpm_61_documentation
- FHWA, 2004. Signalized Intersections: Information Guide [WWW Document]. URL

- 1 <https://www.fhwa.dot.gov/publications/research/safety/04091/12.cfm>
- 2 Ghanim, M.S., Shaaban, K., 2019. Estimating Turning Movements at Signalized Intersections
3 Using Artificial Neural Networks. *IEEE Trans. Intell. Transp. Syst.* 20, 1828–1836.
4 <https://doi.org/10.1109/TITS.2018.2842147>
- 5 Gu, Y., Lu, W., Qin, L., Li, M., Shao, Z., 2019. Short-term prediction of lane-level traffic
6 speeds: A fusion deep learning model. *Transp. Res. Part C Emerg. Technol.* 106, 1–16.
7 <https://doi.org/10.1016/j.trc.2019.07.003>
- 8 Guo, G., Zhang, T., 2020. A residual spatio-temporal architecture for travel demand forecasting.
9 *Transp. Res. Part C Emerg. Technol.* 115, 102639.
10 <https://doi.org/10.1016/j.trc.2020.102639>
- 11 Guo, K., Hu, Y., Qian, Z., Liu, H., Zhang, K., Sun, Y., Gao, J., Yin, B., 2021. Optimized Graph
12 Convolution Recurrent Neural Network for Traffic Prediction. *IEEE Trans. Intell. Transp.*
13 *Syst.* 22, 1138–1149. <https://doi.org/10.1109/TITS.2019.2963722>
- 14 Hamilton, W.L., Ying, R., Leskovec, J., 2017. Inductive representation learning on large graphs.
15 *Adv. Neural Inf. Process. Syst.* 2017-Decem, 1025–1035.
- 16 Hao, S., Lee, D.H., Zhao, D., 2019. Sequence to sequence learning with attention mechanism for
17 short-term passenger flow prediction in large-scale metro system. *Transp. Res. Part C*
18 *Emerg. Technol.* 107, 287–300. <https://doi.org/10.1016/j.trc.2019.08.005>
- 19 Hochreiter, S., Unger Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Comput.* 9,
20 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- 21 Kipf, T.N., Welling, M., 2016. Semi-Supervised Classification with Graph Convolutional
22 Networks. 5th Int. Conf. Learn. Represent. ICLR 2017- Conf. Track Proc.
- 23 Lee, S., Xie, K., Ngoduy, D., Keyvan-Ekbatani, M., 2019. An advanced deep learning approach
24 to real-time estimation of lane-based queue lengths at a signalized junction. *Transp. Res.*
25 *Part C Emerg. Technol.* 109, 117–136. <https://doi.org/10.1016/j.trc.2019.10.011>
- 26 Lee, Y.L.Y., 2009. Freeway travel time forecast using artificial neural networks with cluster
27 method. 2009 12th Int. Conf. Inf. Fusion 1331–1338.
- 28 Li, G., Knoop, V.L., van Lint, H., 2021. Multistep traffic forecasting by dynamic graph
29 convolution: Interpretations of real-time spatial correlations. *Transp. Res. Part C Emerg.*
30 *Technol.* 128, 103185. <https://doi.org/10.1016/j.trc.2021.103185>
- 31 Li, W., Ban, X.J., Zheng, J., Liu, H.X., Gong, C., Li, Y., 2020. Real-Time Movement-Based
32 Traffic Volume Prediction at Signalized Intersections. *J. Transp. Eng. Part A Syst.* 146,
33 04020081. <https://doi.org/10.1061/JTEPBS.0000384>
- 34 Li, Y., Yu, R., Shahabi, C., Liu, Y., 2018. Diffusion convolutional recurrent neural network:
35 Data-driven traffic forecasting. 6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track
36 Proc. 1–16.
- 37 Li, Z., Wang, C., Emrich, C.T., Guo, D., 2018. A novel approach to leveraging social media for
38 rapid flood mapping: a case study of the 2015 South Carolina floods. *Cartogr. Geogr. Inf.*

Sci. 45, 97–110.

Liu, F.T., Ting, K.M., Zhou, Z.H., 2012. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data* 6, 1–44. <https://doi.org/10.1145/2133360.2133363>

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Yong, Wang, Yunpeng, 2017. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors (Switzerland)* 17, 1–16. <https://doi.org/10.3390/s17040818>

Ma, X., Tao, Z., Wang, Yinhai, Yu, H., Wang, Yunpeng, 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* 54, 187–197. <https://doi.org/10.1016/j.trc.2015.03.014>

Mahmoud, N., Abdel-aty, M., Cai, Q., Yuan, J., 2021. Predicting cycle-level traffic movements at signalized intersections using machine learning models. *Transp. Res. Part C* 124, 102930. <https://doi.org/10.1016/j.trc.2020.102930>

Mahmoud, N., Abdel-Aty, M., Cai, Q., Yuan, J., 2022. Estimating cycle-level real-time traffic movements at signalized intersections. *J. Intell. Transp. Syst. Technol. Planning, Oper.* 26, 400–419. <https://doi.org/10.1080/15472450.2021.1890072>

Nezamuddin, N., Jiang, N., Zhang, T., Waller, S.T., 2011. Traffic operations and safety benefits of active traffic strategies on txdot freeways 7.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.

Peeta, S., Ziliaskopoulos, A.K., 2001. Foundations of Dynamic Traffic Assignment : The Past , the Present and the Future. *Netw. Spat. Econ.* 1, 233–265.

PyTorch [WWW Document], 2016. URL <https://pytorch.org/>

Qiao, W., Haghani, A., Hamed, M., 2013. A Nonparametric Model for Short-Term Travel Time Prediction Using Bluetooth Data. *J. Intell. Transp. Syst.* 17, 165–175. <https://doi.org/10.1080/15472450.2012.748555>

Qu, W., Li, J., Yang, L., Li, D., Liu, S., Zhao, Q., Qi, Y., 2020. Short-term intersection traffic flow forecasting. *Sustain.* 12. <https://doi.org/10.3390/su12198158>

Rahman, R., Hasan, S., 2021. Real-time signal queue length prediction using long short-term memory neural network. *Neural Comput. Appl.* 33, 3311–3324. <https://doi.org/10.1007/s00521-020-05196-9>

Rahman, R., Hasan, S., 2018. Short-Term Traffic Speed Prediction for Freeways During Hurricane Evacuation: A Deep Learning Approach. 2018 21st Int. Conf. Intell. Transp. Syst. 2018-Novem, 1291–1296. <https://doi.org/10.1109/ITSC.2018.8569443>

Saroj, A.J., Guin, A., Hunter, M., 2021. Deep LSTM Recurrent Neural Networks for Arterial Traffic Volume Data Imputation. *J. Big Data Anal. Transp.* 3, 95–108. <https://doi.org/10.1007/s42421-021-00043-2>

1 Song, Z., Guo, Y., Wu, Y., Ma, J., 2018. Short-term traffic speed prediction under different data
2 collection time intervals using a SARIMA-SDGM hybrid prediction model. PLoS One 14,
3 1–19. <https://doi.org/10.1371/journal.pone.0218626>

4 Transport for London, 2010. Traffic Modelling Guidelines v3 [WWW Document]. URL
5 <https://kamanakom.files.wordpress.com/2012/11/traffic-modelling-guidelines.pdf>

6 van Buuren, S., Groothuis-Oudshoorn, K., 2011. mice: Multivariate Imputation by Chained
7 Equations in R. J. Stat. Software, Artic. 45, 1–67. <https://doi.org/10.18637/jss.v045.i03>

8 Veličković, P., Casanova, A., Liò, P., Cucurull, G., Romero, A., Bengio, Y., 2018. Graph
9 attention networks. 6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc. 1–12.
10 https://doi.org/10.1007/978-3-031-01587-8_7

11 Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014. Short-term traffic forecasting: Where we
12 are and where we’re going. Transp. Res. Part C Emerg. Technol. 43, 3–19.
13 <https://doi.org/10.1016/j.trc.2014.01.005>

14 Wu, C., Wei, C., Su, D., Chang, M., Ho, J., 2004. Travel time prediction with support vector
15 regression. Proc. 2003 IEEE Int. Conf. Intell. Transp. Syst. 2, 1438–1442.
16 <https://doi.org/10.1109/ITSC.2003.1252721>

17 Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S., 2020. A Comprehensive Survey on
18 Graph Neural Networks. IEEE Trans. Neural Networks Learn. Syst. XX, 1–21.
19 <https://doi.org/10.1109/tnnls.2020.2978386>

20 Yi-chang, C., Bottom, J.A., Mahut, M., Paz, A., Ramachandran, B., Waller, S.T., Hicks, J.E.,
21 2011. Dynamic Traffic Assignment: A Primer. Transp. Res. Circ.
22 <https://doi.org/10.1016/j.trd.2016.06.003>

23 Yu, B., Song, X., Guan, F., Yang, Z., Yao, B., 2016. k-Nearest Neighbor Model for Multiple-
24 Time-Step Prediction of Short-Term Traffic Condition. J. Transp. Eng. 142, 04016018.
25 [https://doi.org/10.1061/\(ASCE\)TE.1943-5436.0000816](https://doi.org/10.1061/(ASCE)TE.1943-5436.0000816)

26 Zhan, X., Zheng, Y., Member, S., Yi, X., Ukkusuri, S. V, 2017. Citywide Traffic Volume
27 Estimation Using Trajectory Data 29, 272–285.

28 Zhang, J., Wang, F.Y., Wang, K., Lin, W.H., Xu, X., Chen, C., 2011. Data-driven intelligent
29 transportation systems: A survey. IEEE Trans. Intell. Transp. Syst. 12, 1624–1639.
30 <https://doi.org/10.1109/TITS.2011.2158001>

31 Zhang, S., Tong, H., Xu, J., Maciejewski, R., 2019. Graph convolutional networks: a
32 comprehensive review. Comput. Soc. Networks 6. <https://doi.org/10.1186/s40649-019-0069-y>

34 Zhang, Z., Lin, X., Li, M., Wang, Y., 2021. A customized deep learning approach to integrate
35 network-scale online traffic data imputation and prediction. Transp. Res. Part C Emerg.
36 Technol. 132, 103372. <https://doi.org/10.1016/j.trc.2021.103372>

37 Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H., 2020. T-GCN: A
38 Temporal Graph Convolutional Network for Traffic Prediction. IEEE Trans. Intell. Transp.
39 Syst. 21, 3848–3858. <https://doi.org/10.1109/TITS.2019.2935152>

1 Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph
2 neural networks: A review of methods and applications. *AI Open* 1, 57–81.
3 <https://doi.org/10.1016/j.aiopen.2021.01.001>

4