J. Electrical Systems 20-9s (2024): 1597-1610

[1] Mahmud Un Nabi

[2] M Ashraful Amin

[3] Amin Ahsan Ali

[4] AKM Mahbubur Rahman

Traffic Forecasting using Modified Unified Spatio-Temporal Graph Convolutional Network for Developing City: Dhaka, Bangladesh (A Case Study)

Journal of Electrical Systems

Abstract: - Deep learning models for traffic forecasting gained a lot of success in recent years. Important application in traffic domain is to predict traffic congestion after certain time window based on historical data. While most of the deep learning models are evaluated using well-known traffic dataset containing vehicle speed collected using loop detectors, those model performances were not being tested on the generated traffic dataset from google maps containing traffic density information. We demonstrate the effectiveness of Unified Spatio-temporal Graph Convolutional Network in forecasting traffic congestion based on the traffic data of a developing countries like Bangladesh which is collected from google maps. We have quantified the traffic fluctuation pattern of any road of Dhaka dataset by introducing a single metric (coefficient of variation of traffic density fluctuation) which can explain the traffic congestion fluctuation pattern within a certain time window. We have also analyzed the whole traffic network of Dhaka using centrality measures (betweenness centrality) of Graph Theory. Based on the coefficient of variation of traffic density fluctuation and betweenness centrality of each road, we built clusters of roads. Based on those clusters, we proposed modification of USTGCN for generating better prediction. Finally, the prediction results are compared with the base USTGCN framework and we have explained the factor behind model performance degradation in terms of sparsity of the datasets with which the USTGCN models are trained on.

Keywords: Traffic Forecasting for developing countries, Unified Spatio-Temporal Graph Convolutional Network, Machine Learning.

I. INTRODUCTION

Traffic congestion is one of the most growing concerns for developing countries like Bangladesh. In order to build an Intelligent Transportation System and redesigning the city structure, it is important to know the traffic patterns and factors behind the exhibited patterns across whole traffic network so that the concerned planners can easily design and allocate the resources accordingly. It is also important to know about the regular traffic congestion pattern ahead of time for the daily commuters so that, they can plan their journey accordingly. Normally, most of the daily commuter plan to travel through the shortest path to reach their destination which incurs low travel time. As a result, it becomes really important to know the overall traffic congestion pattern of those shortest path across different timestamp. Traffic congestion patterns across different timestamps of each road are affected by the structural condition of the road and the other related factors created by humans. For example, in countries like Bangladesh, it is a quite normal scenario to see oversized vehicle travelling through the narrow roads. As a result, the overall traffic congestion pattern gets affected for this human made factor. The effect of the human made factors also gets propagated to the neighboring roads which are connected with that narrow road. Forecasting traffic pattern becomes the important problem to be solved immediately in order to let the citizen plan their daily commutes effectively and let the city planners come up with the better traffic distribution plan to decrease the probability of the congestion and deadlocks.

Traffic network in any cities contains a lot of interconnected roads with different characteristics which exhibits spatial and temporal dependencies among the neighborhood roads. While most of the developed countries deployed advanced deep learning techniques based on Graph Neural Network, the developing countries struggles to deploy one of the models just because the existing data collection methods that are being used in the developed countries are pretty costly for developing countries like Bangladesh. Hence, we need to adopt the data collection method that is scalable and less costly compared to the existing methods. A recent work [1] introduced a novel traffic congestion data collection method using Google Maps data which is scalable and cost-effective for the developing countries like Bangladesh.

The collected dataset can be used to train one of the SOTA traffic forecasting model based on Graph Convolutional Network to generate traffic congestion prediction. Among the SOTA models, USTGCN [2] is one of them. The mentioned SOTA model can model the spatial and temporal dependencies easily by leveraging the road network

[1, 2, 3, 4] Center for Computational & Data Sciences, Independent University Bangladesh

structures and temporal features of every road in order to generate prediction for future timestamps. Generating accurate prediction is very important which indicates the successful deployment of the predictive model. In order to improve the existing performance of the USTGCN, several meta information can be used from the training data in order to modify the existing USTGCN architecture for better traffic congestion forecasting. Meta information such as (betweenness) centrality measures for each road of the traffic road network is one of them. This measures the importance of every road in a network in terms of betweenness (see VI-2).

Along with that, the general traffic congestion patterns of each road is also important. It can be measured using the statistical concepts like co-efficient of variation. This kind of meta information can be leveraged to find the roads which exhibit similar characteristics (betweenness centrality and co- efficient of variation of traffic congestion fluctuation) and form clusters. The existing USTGCN models can be modified using the cluster information so that the congestion prediction values generated from the encodings of modified USTGCN gets improved.

In summary, our key contributions are:

- We have analyzed the traffic network of Dhaka and identified two kinds of road based on their width observed from the Google Maps.
- We have also introduced a single comparable scalar (co- efficient of variation of traffic jam fluctuation) for each road which is introduced as a proxy for traffic jam-length fluctuation.
- We have measured the similarity of roads and formed multiple clusters based on their topological (betweenness centrality) and temporal characteristic (co-efficient of variation of traffic jam length).
- Based on the road cluster information, we have proposed a general modification scheme for existing USTGCN.
- We have compared the prediction performance of the modified USTGCN with the existing USTGCN and pointed out the best performing region for modified USTGCN.
- We have also showed that sparse training data can hamper the prediction performance of all variants of USTGCN.
- Finally, for future work, we have also suggested further modification of aggregation scheme of USTGCN by ditching symmetric normalization and embracing the notion of principal neighborhood aggregation inspired by this work in order to achieve better traffic feature prediction.
-

II. Background And Context

Traffic forecasting is one of the real-world problems which is considered to be pretty complex task. Several deep learning methods have been proposed to solve this task. The real-world datasets for traffic forecasting contain information about the roads and their corresponding traffic conditions (e.g., traffic density, vehicle speed etc.) with the timestamp information. We can generate a network of roads based on the connectivity of the roads described in the dataset. This network can express the accessibility of each road residing in the entire road network along with traffic features (traffic congestion length) with corresponding timestamp. More specifically, we can define each node of network as a road and the edge between every node can be defined as the connection between those roads. We can use this dataset to perform any complex downstream machine learning task like traffic forecasting. There is plethora of ways to collect and process traffic forecasting dataset. All of those has their own advantages and disadvantages. In this work, we have studied and implemented the existing data collection method [1] for traffic forecasting which is scalable, cost-effective and efficient compared to other data collection methods for the developing countries like Bangladesh. The collected dataset is further processed in such a way that we can construct a network based on the connectivity of the roads. Every road contains its own features along with the corresponding timestamp. Each feature represents the traffic congestion length with its corresponding timestamp. With this kind of network dataset, we can apply models which can be used to generate spatial and temporal encodings based on the timestamped feature of each node. Graph Neural Network can be used to encode the spatial encoding of the road as this kind of neural network requires spatial connectivity information (adjacency matrix). Since our dataset contains traffic feature of several timestamps, we need some special variant of graph neural network which can leverage both spatial and temporal feature of each node to generate the encodings. The generated encodings can be used to generate traffic congestion prediction.

Unified spatio-Temporal Graph Convolutional Network [2] is one of the SOTA GNN architectural framework which has demonstrated competitive performance predicting traffic congestion based on the real-world traffic dataset like PEMSD7 and PEMSD8. Since our dataset is collected from Google maps, we can assume that our dataset is different form the dataset like PEMSD7 and PEMSD8 in terms of feature variance. Moreover, the feature and the level of sparsity of our dataset in not the same as PEMSD7 and PEMSD8 since these two datasets contain vehicle speed with corresponding timestamp collected via loop detectors [3] placed on the different intersection of the roads, while our dataset contains traffic congestion length with corresponding timestamp collected from google maps.

In this work, we experimented with the existing USTGCN [2] architecture and modified USTGCN based on some meta information of each road ingested from the dataset. The meta information of each road consists of co-efficient of variation of the traffic congestion fluctuation within a time window and the betweenness centrality of each road in the road network. Betweenness centrality was used to measure the influence of an arbitrary road over the network. More detailed discussion can be found in section VI-2. We have conducted traffic fore- casting based on the several variants of our modified USTGCN architecture and demonstrated the best results that we have achieved using our approach and compared with the results of existing USTGCN [2]. We have also showed that the degree of sparsity of training set hampers the overall performance of traffic forecasting using any variant of USTGCN.

III. Related Works

Most of the traffic forecasting related research has been done based on the dataset collected from the roads of developed countries. Dataset like BJER4 dataset built by the authors of this paper [4] by collecting traffic speed data using 39000 loop detector sensors from 12 roads in China, was used for traffic forecasting using Spatio-temporal graph convolutional network [4] California Traffic Department also provided traffic speed related dataset [3] which was built by collecting data from huge number of sensors. Another traffic dataset that was built by collecting traffic data using the vehicle GPS sensors by the author of this paper [5], which includes traffic information of 1250 arterial streets of Chicago.

For the developed countries, the described methods of collecting traffic information sounds trivial while for the developing countries it sounds infeasible. Because, to cover the whole traffic network, a lot of sensors, loop-detectors have to be deployed throughout the city of developing countries which is quite expensive. Moreover, if we want to scale out our network of loop detectors, it will also incur a lot of money which is quite infeasible for the developing countries like Bangladesh. For the developing countries like Bangladesh, we need a method of collecting traffic dataset which is feasible to apply taking account of the cost and scalability. One of the ways it can be done by using the google maps data by periodically monitoring the associated traffic color for certain time window. This kind of method has been proposed recently by the author of this paper [1] The authors collected traffic congestion information encoded in color provided by Google with 30 seconds interval for six months (November 2019-April 2020) using the Google Traffic Layer API. From the dataset they have collected, they have used one month (November 2019) traffic data from their whole dataset for developing statistical (HA, ARIMA) and machine learning (SVR, SVR-Graph) predictive models to generate predictions of traffic congestion for future timestamps. They have shown the efficacy of their models by analyzing the predicted traffic congestion patterns for weekdays and weekends. They have also included their analysis of how the history of traffic congestion length affects the capability of their predictive models.

Modelling traffic congestion based on the Google Maps data raises an important question regarding the accuracy of the collected datapoint w.r.t the actual traffic situation. To be more precise, it was not known if the color encoding provided by Google’s Traffic Layer API is accurate compared to the actual situation of traffic congestion in developing countries. To find that out, a study [6] was conducted based on the traffic data of urban streets of a city of Ecuador collected from the Traffic Layer of Google Maps. The study shows that, the color- coded information provided by Google Traffic Layer have a reasonable trend with the ground truth (actual traffic situation) speeds. They have also showed that the traffic color coding provided by the Google Traffic Layer can relate between LOS and the average speed of ground truth.

In recent years, several deep learning methods were pro- posed for traffic forecasting related task. In order to encode spatial and temporal features, many deep-learning methods were proposed based on Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNN is used for encoding spatial features while the RNN is used for encoding temporal features which are time dependent. However, RNN like architectures are widely known to be difficult to train and computationally heavy because there is a lot of parameters to train. To mitigate the training cost with lesser parameters, authors of this [4] paper proposed a model based on Graph Convolutional Network [7] which has much less amount of parameters compared to the existing CNN and RNN based models. This method was introduced to encode both spatial and temporal feature of traffic road network represented as a Graph, where each node represents roads containing feature vectors with timestamp information and edges represents the connection between each road. Other several Graph Convolutional Network oriented approaches were proposed for traffic forecasting problem. Unified Spatio-Temporal Graph Convolutional Network (USTCGN) [2] is one of them. The motivation behind USTGCN is to generate traffic feature embedding in a unified way by directly propagating traffic feature information across different timestamps for each road (node) using Graph Convolutional Operator (Spectral Graph Convolution). As a result, the proposed model can observe traffic pattern of current timestamps along with the previous timestamps to generate future predictions. The authors of UST- GCN tested their model performance on the dataset introduced by the authors of this [3] paper. Although, the performance of USTGCN was not evaluated on the dataset introduced by this paper [1] which was collected and pre-processed from the Google maps data. Moreover, the characteristics of collected dataset from Google Maps using the method mentioned in this paper [1] were not used for any traffic forecasting related work yet, which are leveraged in this work.

IV. Preliminaries

4.1 Traffic Graph Network

Given a road network expressed as a graph $G = (V, E, A)$ where V represents a set of roads, a set of connection between each road expressed as E and the adjacency matrix A, which represents the structure of the road network based on the connectivity, the goal is to predict the traffic feature of each node $v \in V$ on the next n timestamps $(t + 1, \ldots, t + n)$ by modelling the traffic feature of previous timestamps $(1\ to\ t'thtimestamps)$. Here, we can define traffic feature matrix of t'th times- tamp as $X(t) \in \mathbb{R}$. So, the problem can be formulated as, given a sequence of traffic feature matrix of total n timestamps $(X^{(1)}, X^{(2)}, X^{(3)}, \cdots, X^{(t)})$, predict the next n traffic feature matrix sequence $(x^{t+1}, X^{t+2}, \cdots, X^{t+n})$. We can construct input feature matrix which will be used in predictive modelling by vertically concatenating the matrices contained in the traffic feature matrix sequence, so that the dimension of the input feature matrix will be $(N\ T\ \times\ d)$, where N is the number of total nodes and T is the number of total timestamps included.

4.2 Node Embedding

The basic of Graph Neural Network is, it generates node embeddings $z_u, \forall u \in V$ by utilizing message passing framework applied on the set of node features defined as $x \in \mathbb{R}^{|v| \times n}$. The idea of message passing network can be formalized as follows:

$$z_u^{(k)} = f(z_v^{(k-1)}, agg^{(k-1)}(z_v^{(k-1)}, \forall v \in N(u))) \quad (1)$$

where $f\ (.)$ and $agg^{(k)}(.)$ are differentiable functions, $N\ (u)$ represents the set of neighbors of node $u \in V$, z_k^u is the feature embedding of the node $u \in V$ after $k'th$ aggregation and z_v^k is the feature embedding of the node neighbor of node $u \in V$. After $k'th$ aggregation, every node feature will inherit the embedding of $k'th$ neighborhood updated by function $f\ (.)$and aggregated by $agg^k(.)$. In general, graph neural network's main motivation is to aggregate and propagate spatial information using node features with the help of graph adjacency matrix. The final embeddings can be used to conduct various machine learning task like classification, regression etc. There is a plethora of node embedding methods for creating effective embeddings for downstream tasks. In this project, we have used the GCN [7] method for creating node embeddings.

4.3 C. Graph Neural Network

The aggregation rule for a given graph G and the corresponding adjacency matrix A can be formulated as follows,

$$Z^{(k)} = D^{-1/2} A^{D^{-1/2}} X^{(k-1)} W^{(k-1)} \quad (2)$$

where $Z^{(k)}$ is defined as an embedding matrix containing the embedding of all the nodes in graph at $k'th$ aggregation. A is an adjacency matrix of the graph, $X^{(k-1)} \in \mathbb{R}^{|v| \times n}$ is feature matrix containing input feature generated after $(k - 1)$'th aggregation of all nodes in a graph. Here $D \in \mathbb{R}^{|v| \times |v|}$ is a degree matrix in which the diagonal entries contain the node degrees and the rest of the entries are zero.

The matrix $D^{-1/2}$ is responsible fore normalizing the feature aggregation to eliminate the influence of nodes with higher degree. It is also true that, the inverse diagonal degree matrix will also eliminate the ability to encode node neighborhood structure information which can be essential for many down- stream tasks. In that case, we won't be able to leverage the neighborhood structure information which can be used for traffic density regression task. Here, the nodes with high degree of connectivity in a traffic network may have higher influence on the other nodes other than nodes with low degree of connectivity. But it is also true that, this single topological node level metric like node degree doesn't always depict the influence of node over other nodes in a network. There can be other non-topological factors involved like traffic fluctuation pattern, weather condition etc.

1) Spatio-Temporal Embedding of Node Features:

In regression task, specifically for time series prediction task, people have used many variations of recurrent neural network architectures which encodes the time series information of previous timestamps and build inference model to generate prediction for the next timestamps. However, in Graph domain, only spatial level information can be formulated using the adjacency matrix and feature matrix for storing the feature of all nodes in a Graph. Therefore, only spatial feature aggregation is possible. In any road network, roads that are connected with the neighboring roads exhibits traffic pattern which can be explained by modelling the dependencies between the neighboring roads traffic patterns across the different timestamps. The degree of connectivity of each road in a network plays a vital role in influencing other roads traffic pattern. In order to encode temporal features along with spatial features, USTGCN [2] introduced Cross-Spacetime edges. The idea is to encode both spatial feature and temporal features of the road network of the previous timestamps in unified way. This method can capture traffic pattern of target nodes from different timestamps and aggregate the neighboring nodes traffic pattern to encode both spatial and temporal dependencies. The idea of Cross-Spacetime edges and the USTGCN aggregation method are described in section IV-C2 and IV-C3 consecutively.

2) Cross-Spacetime Edges:

USTGCN encodes complex spatio-temporal relationship from different timestamps by constructing a lower triangular adjacency matrix, which can be defined where the diagonal entries of $A_{st}[i, i] = A$ (adjacency matrix of the road network), lower triangular entries, $A_{st}[i, j] = \tilde{A}$, where $i < j$ and $\tilde{A} = (A + I)$. Here, I is an identity matrix. Rest of the entries of A_{st} can be defined as $A_{st}[t:]$, which aggregates the spatio-temporal encoding of the network from $1\ to\ (t - 1)$ timestamps. Each entry in the lower triangular part of A_{st} aggregates the feature of 1-hop neighborhood only.

In general, the lower triangular submatrix is responsible for aggregating the traffic feature of neighboring nodes and target nodes from the previous timestamps whereas, the diagonal submatrix is responsible for aggregating traffic feature of the neighboring nodes only.

3) USTGCN Aggregation Methods:

In USTGCN, by constructing a lower triangular adjacency matrix A_{st} and D_{st} where $D_{st}[i,i] = \sum_{k=1}^{NT} A_{st}[i,k]$ and an initial feature matrix $x_{self}^{l} \in \mathbb{R}^{(NT\times d)}$ (feature matrix from the previous layer), where the input feature matrix contains the stacked up node features of last previous T timestamps, graph convolution is performed at each convolutional layer l, parameterized by a broadcasted weight matrix $w_{temp}^{l} \in \mathbb{R}^{(NT\times d)}$ which learns the importance of different timestamps as follows,

$$X_{st}^{l} = D_{st}^{-1/2} A_{st} D_{st}^{-1/2} \left(X_{self}^{l} \cdot w_{temp}^{l}\right) \quad (3)$$

The learned spatio-temporal embedding matrix at layer l, X_{st}^{l} is concatenated with X_{self}^{l} and aggregated using a final shared learnable weight matrix $w_{final}^{l} \in \mathbb{R}^{(2d\times d)}$ followed by applying non-linearity which will learn the importance of self-representation of the current timestamps as well as aggregated spatio-temporal embeddings as follows,

$$X_{self}^{l+1} = Re\,LU\left(w_{final}^{l}\left(X_{self}^{l} \parallel X_{st}^{l}\right)^{T}\right)^{T} \quad (4)$$

The aggregation methods that we have presented here are applied in every convolutional layer of USTGCN for each timestamp. The embedding for each timestamp is concatenated and aggregated by another weight matrix W_F as follows,

$$Z_F = W_F \cdot \left(z_E^{\langle 1\rangle} \| \ldots \| Z_E^{\langle T\rangle}\right) \quad (5)$$

The final learned embeddings Z_F are passed through a neural network for the specific downstream regression task. In this work, we tried to modify the aggregation functions of UST- GCN in order to encode similar embeddings for the roads that have similar characteristics of traffic density across different aggregated timestamps which may lead to the improvement in the accuracy on sparse dataset pre-processed from Google maps data.

In USTGCN, the broadcasted weight matrix $w_{temp}^{l} \in \mathbb{R}^{(NT\times d)}$, is constructed by stacking same learnable weight matrix, $W_b^{l} \in \mathbb{R}^{(n\times d)}, T$ times, which means all node feature vectors across the different timestamps are aggregated by same learnable weight parameters by constructing W_{temp}^{l} as follows,

$$W_{temp}^{l} = \left(W_b^{l} \oplus \ldots \oplus W_b^{l}\right) \quad (6)$$

In the equation above, $\oplus$ represents vertical concatenation. USTGCN learns the parameter of W_b^{l} only during training via backpropagation. It is possible to construct W_{temp}^{l} in a different way by initializing W_b^{l} differently. The way of initializing W_b^{l} may include some sort of node-level properties (betweenness centrality, average co-efficient of variation of traffic congestion fluctuation for each road) from the training set to let the model recognize the nodes/roads with similar characteristics which we have explored in this work.

V. Dataset Description

5.1 Data Collection Procedure:

Collecting traffic data by deploying sensors, loop detectors are not cost-efficient and manual data collection is time-consuming. Most of the data collecting methods that are being used are not scalable enough. In order to collect the data in a scalable manner with the advantage of flexibility in adjusting data pinging interval and getting data remotely, we have collected traffic data following the automatic data capturing method from Google Maps described in [1]. According to their approach, traffic congestion information of each road segment colored by traffic color code (green, orange, red, dark red) indicating no traffic, moderate traffic, heavy traffic and very heavy traffic respectively are collected and pre-processed using image-processing. The collected dataset includes pixel-wise traffic density information of 65 intersections of Mirpur area. For each inter-section, traffic density information spanning from 6:00 AM to 11:59 PM with an interval of 30 seconds were captured. The entire dataset contains 92 days of traffic density information spanning from 1st November, 2019 to 31st January,2020. This dataset covers only Mirpur area of Dhaka, Bangladesh. After collecting the data, we have converted each adjacent intersections to a road and created an adjacency matrix that represents the connection between the roads. (Described in V-3) Geographically, roads can be classified in many ways based on their spatial characteristics, type of materials used to build the roads, traffic capacity information etc. These characteristics also affects the traffic fluctuation over time. Since, we have captured only traffic density information of roads from Google Maps which varies from time to time, we have classified roads based on the visibility from above in Google maps. We have classified the roads of our entire traffic network into two types. We have used two kinds of zoom level to distinguish between main roads (Tier-1) and branch roads (Tier-2). Roads that are visible from 7964 meters above in Google maps are considered as Tier-1 roads whereas roads which are visible from 3982 meters above are considered as Tier-2.

5.2 Road Characteristics:

According to the definition of road tier described in the previous section, we were able to classify total 76 roads as Tier-1 and 78 roads as Tier-2. There is significant difference between these two types of road w.r.t their road width. We have estimated the road width of different tiers and report the results using confidence intervals.

We calculate the mean (μ) and standard deviation (σ) of road width to calculate the range of road width using confidence interval (CI), which is defined in equation given below:

$$CI = u + \rho \frac{\sigma}{\sqrt{n}} \tag{7}$$

where ρ is confidence level value and n is number of roads in a tier. We report the result of 99.9% confidence level value.

Table I: Tier-1 Road Widths Distribution Properties

Mean (m)	4.25
Confidence Interval (m)	0.10
Upper limit (m)	4.35
Lower limit (m)	4.15

Table II: Tier-2 Road Widths Distribution Properties

Mean (m)	3.55
Confidence Interval (m)	0.05
Upper limit (m)	3.60
Lower limit (m)	3.51

Traffic congestion on all roads cannot be the same as the road width controls the access of differently sized motorized/non-motorized vehicles. Moreover, it is safe to assume that, the topological connection of every road can be different. The number of ingoing and outgoing connection also matters. The traffic fluctuation patterns across each road might not be the same as every road doesn't exhibit same characteristics.

5.3 Data Pre-processing:

The collected data contains 65 intersection segments traffic density information. Information is encoded as traffic jam length measured by the length of traffic color code for every 30 seconds for each intersection. For network analysis and inference, we converted the dataset in a form of a Graph. We can define the whole road network as,

$$G = (V, E) \tag{8}$$

where V is a set of intersections and E is a set of connections(edges) between the adjacent intersections. Here, the connection between two intersections i and j can be defined as,

$$e_{ij} \in E \tag{9}$$

There exist two edges $e_{ij} \in E$ which connects intersection i and j, $e_{jk} \in E$ which connects intersections $j \; and \; k$. We treat them as individual node defined as $v_{e_{ij}} \in V'$ and $v_{e_{jk}} \in V'$ respectively. Here, $v_{e_{ij}}$ means a road that starts from intersection i and ends to intersection j. Two roads $v_{e_{ab}} \in V'$ and $v_{e_{cd}} \in V'$ are adjacent to each other if $b = c$, which means, road $v_{e_{cd}} \in V'$ is accessible from road $v_{e_{ab}} \in V'$.

We further index the roads and build an adjacency matrix that explains the connectivity and accessibility of each road in the network. For the regression task, which we have conducted using Unified Spatio-temporal Graph Convolutional Network described in [2], we convert the traffic density information of all roads into a Graph defined as,

$$G' = (V', E') \tag{10}$$

where each node defined as,

$$v_{e_{ij}} \in V' \tag{11}$$

which corresponds to each road connecting two intersections $i \; and \; j$. Each edge connecting adjacent roads is defined as,

$$e_{v_{e_{ij}} v_{e_{kl}}} \in E' \tag{12}$$

where for two adjacent roads $v_{e_{ij}}$ and $v_{e_{kl}} \subset V'$, we define and edge that goes from road $v_{e_{ij}}$ to $v_{e_{kl}}$ as,

$$e_{v_{e_{ij}} v_{e_{kl}}} = \left(v_{e_{ij}} \rightarrow v_{e_{kl}} \right) \tag{13}$$

which means road $v_{e_{kl}}$ is accessible from $v_{e_{ij}}$. Road $v_{e_{kl}}$ is accessible from road $v_{e_{ij}}$ if $j = k$. We construct an adjacency matrix A' where each entry of A' is defined as,

$$A'_{v_{e_{ij}} v_{e_{kl}}} = \begin{cases} 1 \; if \; j = k \; in \; e_{v_{e_{ij}} v_{e_{kl}}} \in E' \\ 0, \; if \; j \neq k \; in \; e_{v_{e_{ij}} v_{e_{kl}}} \notin E' \end{cases} \tag{14}$$

we have constructed the transformed feature matrix $X_{transformed} \in \mathbb{R}^{(NT \times d)}$ and used it along with A' to learn the weight parameters of our inference model.

5.4 4) Dataset Characteristics:

We have collected data of three months (November 2019, December 2019, January 2020) comprising of 92 days of traffic density information of total 154 roads. We calculated the sparsity of each pre-processed dataset using the following formula,

$$sparsity = 1 - \left(\frac{C_n}{C_{tc}}\right) \quad (15)$$

where C_n is the total count of non-zero datapoints in the dataset and C_{tc} is total datapoints in the dataset.

Table III: Sparsity and Shape of Each Pre-Processed Dataset of Dhaka Road Network

Dataset	Shape	Sparsity
November 2019	(154,64800)	0.756
December 2019	(154,66960)	0.591
January 2020	(154,66960)	0.552

VI. Methodology

Given a feature matrix $X_{transformed}$ and the adjacency matrix A', we have built a unified traffic forecasting model based on [2] which forecasts traffic information up to next 15 minutes, 30 minutes and 45 minutes. The inference model of ours resembles the USTGCN described in [2], in which the same importance is given to the node features across all the timestamps using the broadcasted learnable weight matrix.

But not all the roads traffic congestion behavior might be the same. So, not all the roads exhibit the same traffic congestion characteristics. There are some factors that can be used to explain the roads traffic congestion characteristics as well as find out and cluster the roads with similar characteristics. This type of meta information can be used to construct better representation cluster-wise and use it for any downstream task.

In this work, we modified USTGCN framework based on the road clusters to test whether it will improve the forecasting result or not. For clustering similar roads, we have considered two factors, betweenness centrality and co-efficient of variation of traffic density of each road. Betweenness centrality is one of the topological properties of a road(node) in a road network. This property can explain the importance of each node(road) of the road network based on the betweenness (see VI-2). On the other hand, coefficient of variation of traffic density quantifies the traffic fluctuation pattern. Based on these two properties of each road, we have applied K-means clustering algorithm to generate clusters of roads. Every cluster represents a set of roads which exhibits similar representation. Using the cluster information, we can modify the existing USTGCN [2] model so that the feature vectors of roads existing in same cluster are aggregated using the same learned weight vectors across each timestamp which we have dis- cussed in VII. The model performance is evaluated using RMSE and MAE which is reported in experimental setup and results segment (see VIII). Although, we are working on time series dataset, we have tested our model performance using different train-test segments (see Table IV in order to show the contribution of sparsity of training set on the model performance degradation.

The whole dataset contains, the traffic information of November, December of 2019 and January of 2020. We have trained the model by keeping 2 months of data as training set and validated on the rest 1 month of data. So, every month of the whole dataset was used as a validation set. The reason of doing this is to capture the model performance on different training set with different characteristics in order to find out what factors in the training set that are responsible for model performance degradation or improvement.

1) Topological Properties of Mirpur Road Network:

Centrality measures were introduced to measure the popularity or the influence of any node in a social network. The same centrality measures can explain the influence of each road in a road network. The definition of centrality measure based on betweenness was first introduced in [8]. It is based on the shortest path. We chose to calculate betweenness centrality for each road. Betweenness centrality can rank each node based on their topological characteristics.

2) Betweenness Centrality:

Given an arbitrary Graph $G = (V, E)$ where V *is a set of vertices and* E is a set of edges. For each $v \in V$, betweenness centrality can be defined as,

$$c(v) = \frac{\Sigma_{s \neq v \neq t} p_{st}(v)}{p_{st}} \quad (16)$$

where p_{st} is total number of shortest paths starting from node s *to node* t. $p_{st}(v)$ is the number of those paths that goes through v. For scaling the score so that $c \in [0,1]$, we can further normalize the value of $c(v)$ with minmax normalization,

$$c_{norm}(v) = min\,max\big(c(v)\big) \quad (17)$$

where $min\,max\big(c(v)\big)$ is defined as,

$$min\,max\big(c(v)\big) \;=\; \frac{c(v)-\min(c)}{(c)-\min(c)} \tag{18}$$

Nodes with high betweenness centrality score (close to 1) have more influence over the network compared to the nodes which have low betweenness centrality score. In case of our Road network, every node is considered as a road. From the commuter's perspective, they always tend to use the shortest paths in the road network to reach their terminal point(road) from their starting point(road) to minimize the travel time. Roads with high betweenness centrality scores are accessed more often by the commuters than the roads with low betweenness centrality. Because this score signifies the betweenness of each road (node) among overall shortest paths of the network. But it is also true that, a piece of information in a network doesn't always flow through the shortest paths. To be precise, an arbitrary commuter may not consider travelling through the shortest path all the time. There might be a lot of reason behind these decisions of commuters. So, only topological statistic like every node's betweenness centrality of Dhaka Road network isn't enough to analyze the traffic density pattern. In order to analyze the traffic density pattern in our road network, along with betweenness centrality, we have used the notion of coefficient of variation to measure how traffic density fluctuates which is described in the next sub-section.

6.1 Traffic Density Fluctuation Analysis:

We have used betweenness centrality as a topological characteristic of our road network which doesn't depict anything about the traffic density fluctuation of each road. In order to measure the traffic density fluctuation, we have used the coefficient of variation of traffic density for each road. Our scoring methods simply describes for each road, how much the traffic density fluctuates over the time w.r.t the mean of each roads traffic density. Moreover, not all the roads mean of traffic density fluctuation might not be the same. Hence, we chose this method so that we can compare the scores between any pair of roads in the network. This information can be used along with betweenness centrality to cluster the roads with the closest coefficient of variation of traffic density fluctuation. We will use this cluster information and ingest it to USTGCN which will help the model learn different weight matrices for roads with different characteristics (given betweenness centrality and coefficient of variation of traffic density (see next sub-sub-section)).

1) Co-efficient of variation as a measure of traffic density fluctuation:

Given a vector $X_i = \big(a_j, \cdots, a_n\big)$ that contains traffic density information of a road i with total timestamp of n, we can define another vector $Y_i = \big(b_j, \cdots, b_{n-1}\big)$ where $b_j = \big|a_j - a_{j+1}\big|$. Here, $|y_i| = (n-1)$. Now we can define the co-efficient of variation of road $i\ as\ \lambda_i$,

$$\lambda_i = \frac{\sigma_{Y_i}}{u_{Y_i}} \tag{19}$$

where $\{\lambda_i \epsilon \mathbb{R} | 0 \le \lambda_i < \infty\}$, u_{Y_i} ids defined as the mean of vector Y_i, σ_{Y_i} as the standard deviation of vector Y_i. The score defines how much traffic density fluctuates w.r.t the mean. Roads with λ close to 0 are the roads which exhibits low traffic density fluctuation in between n timestamps and vice versa. The only reason of scoring each road based on their traffic density fluctuation is to group the similar kind of roads which exhibits similar fluctuation patten that can be explained by λ. Along with topological characteristics (betweenness centrality) of each road described in VI-2, we cluster all roads based on their co-efficient of variation of traffic density using K-means clustering method.

6.2 Road Cluster Analysis:

In order to mine similarity of roads with similar traffic fluctuation pattern and betweenness centrality, we have conducted K-means clustering on 2-dimensional data of road characteristics (betweenness centrality and co-efficient of variation of traffic density fluctuation). Before clustering, we have calculated average co-efficient of variation of traffic density fluctuation for each road on the training set. Given co-efficient of variation of traffic density fluctuation, λ_i^j for road i on day j, we can calculate average co-efficient of variation of traffic density fluctuation for each road over the entire training set as,

$$\lambda_{i_{avg}} = \frac{\sum_{J-1}^{n} \lambda_i^j}{n} \tag{20}$$

where n is total number of days in training set and $\lambda_{i_{avg}}$ is average co-efficient of variation of road i over the entire training set. So, for all roads r, we can instantiate r number of observations $(l_1, \ldots, l_r)$, where each observation is a 2-dimensional data points comprises of betweenness centrality and co-efficient of variation of traffic density fluctuation for each road of the training segments. We have clustered each road based on these observations using K-means clustering. All the labels of the road after clustering were used to instantiate different weight matrices for aggregation in our modified USTGCN model. These different weight matrices were learned during the training process.

1) Cluster Parameters Selection:

Given a set of observations $S = \{s_1, \ldots, s_n\}$, where each observation is a 2D points containing betweenness centrality score and average co-efficient of variation of traffic density fluctuation which are calculated from the training set, we initialize centroids randomly at the 1st timestep and applied K-means clustering for 300 epochs. We have applied K-means algorithm by instantiating 2, 3 and 4 cluster centroids on these observations of 3 different training sets.

For each cluster, we instantiate different learnable weight vectors. The cluster member's (roads) features are aggregated with the same weight parameters during training the model which we have discussed in the next section (see VII).

VII. Proposed Modification Of Node Feature Aggregation In Ustgcn

In this chapter, we will discuss about how we can leverage the cluster information to modify the existing aggregation functions described in USTGCN. Recall from section 2.2.4, USTGCN takes feature matrix of previous layer $X_{self}^{l} \in \mathbb{R}^{(NT \times d)}$, the lower triangular adjacency matrix A_{st} as inputs and creates spatio-temporal embedding using equation 1, parameterized by broadcasted learnable weight matrix $W_{temp}^{l} \in \mathbb{R}^{(NT \times d)}$ which learns the importance of node features across different timestamps at layer l. Here, W_{temp}^{l} is constructed by vertically concatenating same learnable weight parameter $W_{b}^{l} \in \mathbb{R}^{(T \times d)}$ N time using equation 4, where N is the total number of roads in a network. T is the total number of timestamps and d is the number of days that we consider to training USTGCN. Instead of constructing W_{temp}^{l} this way, we construct W_{temp}^{l} by initializing different weight parameters by using the cluster information created based on the meta information (betweenness centrality and co-efficient of variation) ingested from training set.

Given a set of clusters $S = \{s_1, s_2, \cdots s_{n_C}\}$ created based on the betweenness centrality and average co-efficient of variation of each road ingested from the training set, we can instantiate a set of learnable weight vectors $W = \{w_1, w_2, \cdots w_{n_c}\}$ where every element of W corresponds to every element of S. Here, the length of each vector in W is the number of previous days that we consider to train the model on which is in our case 8 days. For example, for the 1st cluster set s_1, the corresponding learnable weight vector is w_1,which means to learn the importance of different timestamps, all the aggregated feature vectors of roads residing in the 1st cluster set aggregated by lower triangular matrix A_{st}, are weighted using the learnable weight vector w_1 to create spatio-temporal embedding X_{st}^{l} at layer l. This procedure is applied for the rest of the cluster sets too.

We store each roads cluster information as key-value pairs denoted by,

$$clusterinfo_{ij} = \langle i, j \rangle \quad (21)$$

where i is the index of the road and j is the index of the cluster in which the $i'th$ road resides. Here, $i \in [0, N)$ and $j \in [1, N_c]$, where the total number of clusters denoted as N_c. We store these key-value pairs in a set denoted as $S_{cluster}$. Here the cardinality of $S_{cluster}$ can be denoted as,

$$|S_{cluster}| = N \quad (22)$$

Where N is the number of roads in the Dhaka road network. Instead of constructing W_{temp}^{l} using equation 4, we construct $w_{clustered}^{l} \in \mathbb{R}^{(N \times d)}$ such that, each row of $w_{clustered}^{l}$ can be denoted as,

$$w_{clustered}^{l}[i:] = W'[S_{cluster}[i]] \quad (23)$$

where $S_{cluster}[i]$ represents the cluster index of road i and $W'[S_{cluster}[i]]$ represents the learnable weight vector stored in W' for road i residing in the cluster indexed as $S_{cluster}[i]$. Finally, we construct W_{temp}^{l} by concatenating $w_{clustered}^{l}$ vertically T times as follows,

$$W_{temp}^{l} = (w_{clustered}^{l} \oplus \ldots \oplus w_{clustered}^{l}) \quad (24)$$

Here, T denotes total number of timestamps. Using the constructed W_{temp}^{l}, the spatio-temporal embedding X_{st}^{l} is learned using equation 4.

In USTGCN, the learned spatio-temporal embedding matrix at layer l, X_{st}^{l} is concatenated with X_{self}^{l} and aggregated using a final shared learnable weight matrix $W_{final}^{l} \in \mathbb{R}^{(2d \times d)}$ followed by applying non-linearity which will learn the importance of self-representation of the current timestamp as well as aggregated spatio-temporal embedding using equation 5.

In this work, we have also modified equation 2 by incorporating two learnable weight matrices instead of W_{final}^{l} as follows,

$$X_{self}^{l+1} = \mathrm{ReLU}\left(\left(W_{final_{self}}^{l} \cdot X_{self}^{l}\right)^{T} || \left(W_{final_{st}}^{l} \cdot X_{st}^{l}\right)^{T}\right)^{T} \quad (25)$$

where $W_{final_{self}}^{l}$ and $W_{final_{st}}^{l}$ are used to encode both feature matrix of previous layer and spatio-temporal embedding separately created at layer l. After generating embeddings for each timestamp, we encode the final embeddings using equation 3. The final embedding of modified USTGCN was passed through a three-layer neural network where we have used Tanhshrink as activation function in between the first two linear layers and ReLU after the last linear layer. Here Tanhshrink function is defined as,

$$\mathrm{Tanhshrink}(x) = x - \tanh(x) \quad (26)$$

VIII. Experimental Setup And Results

8.1 Train-Validation Split:

We split the dataset into training and validation segments, where training segment contains 61 days of traffic density information of 154 roads and validation segment contains 31 days of traffic density information. We trained the model for 200 epochs with early stopping enabled, validated on the validation set and saved the model parameters which results in the best result based on RMSE and MAE measured on validation set. We have trained the models for predicting traffic density up to 15, 30, 45 minutes and report the best result. We have reported only RMSE and MAE of overall validation dataset.

We have generated results for each day of validation set by passing last 7 days data and the previous 1 hour of data (before the hour of the day from which we want to generate prediction) to the model.

8.2 Training Hardware Environment:

Our model was trained on NVIDIA GTX 1060 6 GM GDDR5 GPU with AMD Ryzen 5 3600 6 core, 12 thread CPU and 16 GB of RAM.

8.3 Hyperparameters:

For each model that we have trained, we have chosen Adam optimizer with initial learning rate 0.0001. We stacked 4 layers of our GNN and for the downstream regression task, we have used feed-forward neural network with 3 hidden layers and Tanhshrink function as non-linear activation function. We trained every model for 200 epochs with early stopping enabled. During training, we chose to fix the seed value to 824.

8.4 Model performance evaluation metrics

To assess the model performance, we have chosen two criterion, Root-Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Given the ground truth for $i'th$ data sample y_i, RMSE can be defined as follows,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\widehat{y_i}-y_i)^2}{n}} \quad (27)$$

The lower value of RMSE is desirable since it indicates the predicted values are not far away from the ground truths. This is true for MAE. We can define MAE as follows,

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y_i}| \quad (28)$$

8.5 Results:

We have trained our data on USTGCN and modified (VII) version of USTGCN and reported the performance score based on RMSE and MAE for traffic congestion prediction up to 15 minutes, 30 minutes and 45 minutes. We have also trained our model using our same modified framework just by clustering roads based on their Tier information (see Table I and Table II) denoted as USTGCN (Tier conditioned) in the result tables (see Table IV).

We have showed the result of USTGCN, modified USTGCN by using three different kinds of train-test split (see IV). Among these three train-test splits, the results of every variant of USTGCN are affected by the average sparsity of the training set with which the model was trained on is less compared to the other training set (see Table IV).

Using the data of November 2019 and December 2019 as the training set and January 2020 as validation set, we have achieved the best result on the validation set out of our modified USTGCN when predicting up to 15 minutes using 2 clusters (see Table IV). Our modified USTGCN has also achieved best result (using 4 clusters for predicting up to 15 minutes and 45 minutes) on the validation set of November 2019 when using December 2019 and January 2020 as the training set (see table IV). This training set (January 2020, December 2019) has the lowest sparsity than any other training sets too.

In summary, our modified USTGCN performed better than the base USTGCN [2] when the model is trained on less sparse dataset.

IX. Comparison Of Results From Different Modified Ustgcn Models

In this section, we are going to compare the predicted traffic congestion of roads by USTGCN and best performing modified USTGCN by showing the predicted congestion of traffic of randomly chosen weekday and weekend from the validation set for the roads belonging to the different clusters. The base USTGCN doesn't incorporate any cluster information during the training.

Table IV: Validation Result of Different Type of Ustgcn for Different Prediction Length

Datasets	Prediction length (minutes)	Models	RMSE	MAE
	15	USTGCN	20.44	12.38
		USTGCN (2 tier)	20.53	12.55
		USTGCN (2 cluster)	20.54	12.51
		USTGCN (3 cluster)	20.53	12.47
Train:		USTGCN (4 cluster)	20.50	12.57
	30	USTGCN	22.90	13.93
		USTGCN (2 tier)	23.01	14.09
		USTGCN (2 cluster)	22.95	14.22
Validation:		USTGCN (3 cluster)	22.98	14.00

		USTGCN (4 cluster)	22.85	13.91
		USTGCN	24.57	15.13
	45	USTGCN (2 tier)	24.64	15.05
		USTGCN (2 cluster)	24.71	15.21
		USTGCN (3 cluster)	24.48	15.04
		USTGCN (4 cluster)	24.51	15.03
		USTGCN	23.43	14.89
	15	USTGCN (2 tier)	22.39	14.09
		USTGCN (2 cluster)	22.37	14.01
Train:		USTGCN (3 cluster)	22.51	14.15
		USTGCN (4 cluster)	22.45	14.16
Nov 19,	30	USTGCN	24.92	15.82
		USTGCN (2 tier)	25.22	15.86
		USTGCN (2 cluster)	25.13	15.82
		USTGCN (3 cluster)	25.07	15.80
Validation:		USTGCN (4 cluster)	24.98	15.88
		USTGCN	26.58	16.88
	45	USTGCN (2 tier)	26.78	16.80
		USTGCN (2 cluster)	26.68	16.89
		USTGCN (3 cluster)	26.65	16.90
		USTGCN (4 cluster)	26.70	16.84
		USTGCN	19.05	10.17
	15	USTGCN (2 tier)	19.07	10.10
		USTGCN (2 cluster)	19.08	10.05
		USTGCN (3 cluster)	19.11	10.30
Train:		USTGCN (4 cluster)	18.97	10.01
		USTGCN	21.04	11.34
	30	USTGCN (2 tier)	21.20	11.34
		USTGCN (2 cluster)	21.12	11.61
Validation:		USTGCN (3 cluster)	21.13	11.64
		USTGCN (4 cluster)	21.18	11.31
		USTGCN	22.36	12.45
	45	USTGCN (2 tier)	22.45	12.69
		USTGCN (2 cluster)	22.44	12.39
		USTGCN (3 cluster)	22.46	12.38
		USTGCN (4 cluster)	22.30	12.28

9.1 Prediction Comparison of a weekend:

Figure 1. Prediction result based on modified USTGCN (2 clusters) of road number 15 on one of the weekends of January,2020. (Training set: November 2019, December 2019). Here, tier-0 means road number 15 belongs to cluster number 1

Figure 2. Prediction result based on USTGCN of road number 15 on one of the weekends of January,2020. (Training set: November 2019, December 2019). Here, tier-0 means road number 15 belongs to cluster number 1. But in USTGCN the clustering was not performed

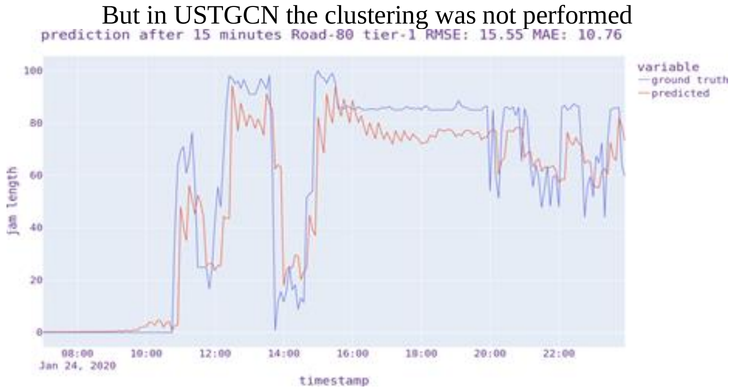

Figure 3. Prediction result based on modified USTGCN of road number 80 on one of the weekends of January,2020. (Training set: November 2019, December 2019). Here, tier-1 means road number 80 belongs to cluster number 2.

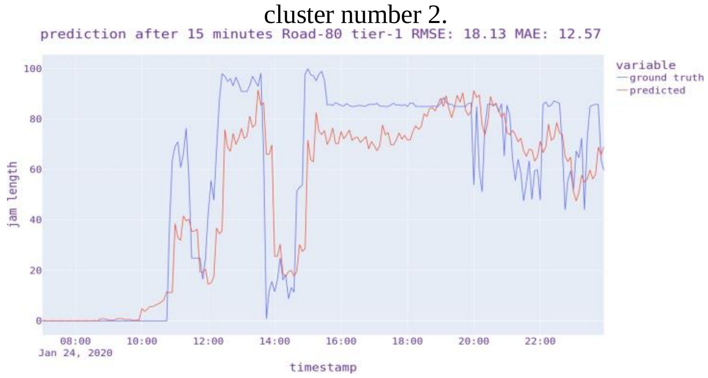

Figure 4. Prediction result based on USTGCN of road number 80 on one of the weekends of January,2020. (Training set: November 2019, December 2019). Here, tier-1 means road number 80 belongs to cluster number 2. But in USTGCN the clustering was not performed

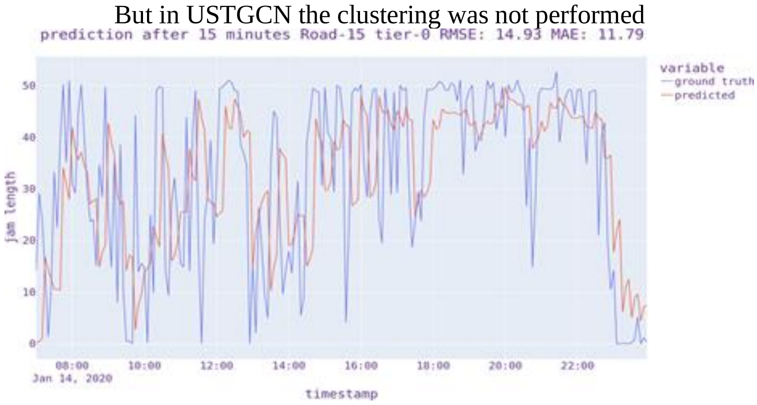

Figure 5. Prediction result based on modified USTGCN (2 clusters) of road number 15 on one of the weekends of January,2020. (Training set: November 2019, December 2019). Here, tier-0 means road number 15 belongs to cluster number 1

Figure 6. Prediction result based on USTGCN of road number 15 on a weekday of January,2020. (Training set: November 2019, December 2019). Here, tier- 0 means road number 15 belongs to cluster number 1. But in USTGCN the clustering was not performed

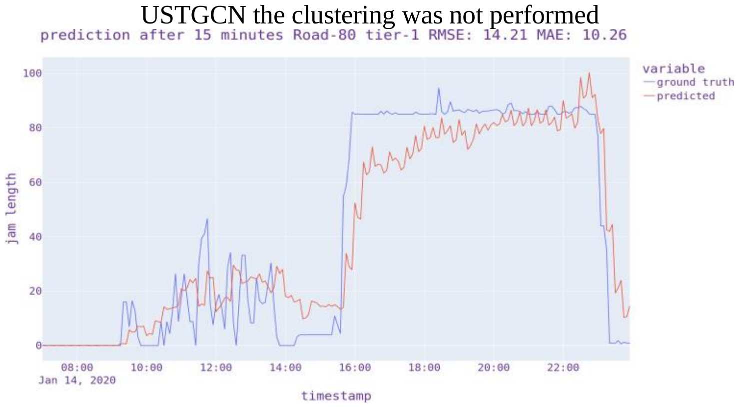

Figure 7. Prediction result based on USTGCN of road number 80 on one of the weekday of January,2020. (Training set: November 2019, December 2019). Here, tier-1 means road number 80 belongs to cluster number 2. But in USTGCN the clustering was not performed

9.2 Prediction comparison of a weekday:

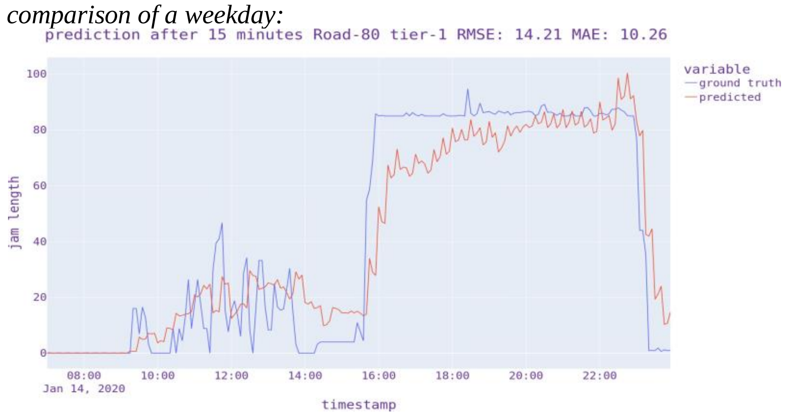

Figure 8. Prediction result based on USTGCN of road number 80 on one of the weekday of January,2020. (Training set: November 2019, December 2019). Here, tier-1 means road number 80 belongs to cluster number 2. But in USTGCN the clustering was not performed

X. SUMMARY

In this work, we have developed a model based on the USTGCN for traffic forecasting using the dataset collected from Google Maps data. We have introduced a new method in order to cluster the roads with similar characteristics (see VI-B) extracted from the training set. Our modified USTGCN leverages the cluster information (see VI) and generates separate embeddings for each cluster (see VII). We have showed that our modified USTGCN performs better than USTGCN for prediction up to 15 minutes while using November 2019 and December 2019 data as training set and January 2020 data as validation set (see VIII-E) and compared the prediction results of roads belonging to the different clusters with the existing USTGCN. We have concluded that, our best perform- ing modified USTGCN is less sensitive to the outliers when generating predictions for weekdays and weekends compared to the existing USTGCN (see IX). We have also showed that, the sparsity in the training data hampers the performance of all variants of USTGCN (see table IV).

XI. FUTURE WORK

In this work, we have proposed a modified USTGCN for traffic forecasting using the dataset of Mirpur area of Bangladesh collected from Google Maps. For each layer of Graph Convolution, we have used single Spectral Convolutional operator described in [7]. This kind of operators aggregate normalized features from the neighborhood of target node to generate embeddings. In traffic forecasting domain, it is important to discriminate between different traffic features while aggregating the neighborhood feature information to increase the model expressiveness and incorporate the inherent traffic network structure during training the GNN model. In the most recent work [9], the authors demonstrated the failures of different aggregator functions and proposes multiple aggregators with degree-scalers which generalize the sum aggregator to capture and exploit the structure of the input graph. The idea of using different aggregators can be extended to be used in USTGCN framework in order to increase the robustness and expressiveness of the model. If we can do this, we will be one step closer in reasoning behind the model generated predictions.

XII. ACKNOWLEDGMENTS

This project has been jointly sponsored by Independent University, Bangladesh and the ICT Division of the Bangladesh Government.

REFERENCES

[1] M. A. Pramanik, M. M. Rahman, A. S. M. I. Anam, A. A. Ali, M. A. Amin, and A. K. M. M. Rahman, “Modeling traffic congestion in developing countries using google maps data,” in Advances in Information and Communication, K. Arai, Ed. Cham: Springer International Publishing, 2021, pp. 513–531.

[2] A. Roy, K. K. Roy, A. A. Ali, M. A. Amin, and A. K. M. M. Rahman, “Unified spatio-temporal modeling for traffic forecasting using graph neural network,” CoRR, vol. abs/2104.12518, 2021.

[3] C. Chen, K. Petty, and A. Skabardonis, “Freeway performance mea- surement system: Mining loop detector data,” Transportation Research Record, vol. 1748, 01 2000.

[4] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting,” CoRR, vol. abs/1709.04875, 2017.

[5] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, “Trafficgan: Network- scale deep traffic prediction with generative adversarial nets,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 1, pp. 219–230, 2021.

[6] Y. García Ramírez, “Developing a traffic congestion model based on google traffic data: A case study in ecuador,” 05 2020.

[7] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” CoRR, vol. abs/1609.02907, 2016.

[8] L. C. Freeman, “A set of measures of centrality based on betweenness,” Sociometry, vol. 40, no. 1, pp. 35–41, 1977. [Online]. Available: http://www.jstor.org/stable/3033543

[9] G. Corso, L. Cavalleri, D. Beaini, P. Lio`, and P. Velickovic, “Principal neighbourhood aggregation for graph nets,” CoRR, vol. abs/2004.05718, 2020. [Online]. Available: https://arxiv.org/abs/2004.05718