Traffic Forecasting using Temporal Line Graph Convolutional Network: Case Study

Abdelrahman Ramadan
EE Department
Qatar University
Doha, Qatar
a.ramadan@qu.edu.qa

Ahmed Elbery
School of Computing
Queen's University
Kingston, Ontario, Canada
a.elbery@queensu.ca

Nizar Zorba
EE Department
Qatar University
Doha, Qatar
nizarz@qu.edu.qa

Hossam S. Hassanein
School of Computing
Queen's University
Kingston, Ontario, Canada
hossam@cs.queensu.ca

***Abstract*—Traffic forecasting is imperative to Intelligent Transportation Systems (ITS), and it has always been considered as a challenging research topic, due to the complex topological structure of the urban road network and the temporal stochastic nature of dynamic change. Popular sports events attract vast numbers of spectators travelling to the event, which will have a substantial effect on ITS, showing peaks on the network that can collapse a smart city's ITS. In this paper, we tackle traffic forecasting and use the Doha network in Qatar and the FIFA World Cup 2022 (FWC 2022) event as a case study. We propose a novel technique for embedding road network graphs into a Temporal-Graph Convolutional Network. The embedding process includes a modification to the graph weights based on graph theory and the properties of the line graph. Extensive simulations are carried out on a real-world calibrated dataset from Doha's road network. Our Temporal Line Graph Convolutional Network (TLGCN) proposal shows outstanding performance when compared to state-of-the-art techniques, not only for huge special events but also for the regular daily traffic.**

***Index Terms*—Traffic Forecasting, Line Graphs, Temporal Line Graph Convolutional Network, TLGCN, T-GCN, Spatiotemporal Dependence.**

I. Introduction

Transportation plays a vital role in everyone's daily lives. A recent report [1] states that commuters in Qatar, vehicles spend on average an extra 98 hours stuck in congested traffic. The report also estimated an economic loss of 1.9 billion USD, due to additional fuel costs and losses in productivity. Given these consequences, under such circumstances, ITS is of paramount importance to drivers, the private sector, and governments. Spatiotemporal forecasting serves a variety of applications, from autonomous vehicle operations to energy and smart grid optimization, to logistics and supply chain management. Widely used transport services, such as flow control (i.e., traffic signal timing and ramp metering), route planning, and navigation, rely heavily on high-quality traffic assessment, where multi-scale traffic forecasting is the foundation of urban traffic control and guidance.

Traffic forecasting in itself is challenging, attributable to the complicated spatiotemporal dependencies and inherent long-term forecasting difficulties. Traffic time series show significant temporal dynamics. Combining recurring patterns (such as rush-hours congestion) with non-recurring events, like accidents, makes long-term prediction complicated.

Authors in [2] proposed the Temporal-Graph Convolutional Network (T-GCN), a Neural Network (NN) based traffic forecasting model for traffic speed but with reduced accuracy as the prediction horizon is expanded. They use Gated Recurrent Units (GRU) [3] to capture temporal dependencies between graph nodes, and Graph Convolutional Network (GCN) to capture spatial dependencies of the network complex topological structure. Nonetheless, the T-GCN model in [2],(was based on the notion of GCN [4], later enhanced by [5], and adapted in [6]) works with vertex-centric graphs, meaning that, it can only be trained to predict the features on the vertices of a graph. However, in real road networks, most of the vehicles' mobility features (such as speed, vehicle, travel time, etc.) are associated with the road links. Thus a unified approach to convert the road network graph (which is edge-centric) into a vertex-centric graph is essential. T-GCN showed promising results when it was trained on real-world datasets. However, the scheme is not robust enough to deal with extreme use cases such as large sports events. When adapted to our case study (the 2022 FIFA World Cup, to be held in Doha, Qatar), the T-GCN performed poorly due to the special features in our case. After each match, an immense number of vehicles will leave stadiums to different destinations, causing severe network congestion. It is a challenging use case, and its features can be encountered in several countries and events (national sports events, large concerts, etc.). This anticipated traffic flow is a special characteristic, that results in distinct traffic conditions, such as the existence of links that are either empty or heavily crowded for a long time. Additionally, this traffic pattern results in different routing techniques [7], [8], where drivers may take longer routes as detours for congested roads.

In this paper, we introduce a new scheme that converts edge-centric network graph into line graph, in which, vertices represent the edges in the original road network. Our novel scheme is an upgraded T-GCN that can understand the network in a better way, consequently, can account for extreme cases, by considering a Temporal Line Graph Convolutional Network (TLGCN). More importantly, we introduce a modification to the weights that considers both the distance proximity between links, as well as the link connectivity represented by its degree in the transformed line graph. The

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

objective of using the link connectivity is to preserve the dynamics and topological structure of the original graph. We further validate our approach by applying TLGCN to predict traffic conditions on a real-world calibrated dataset from Doha's road network. This traffic forecasting will help the governing authorities make better-informed decisions in real-time on route planning and traffic control, during this large event. Since there is no dataset for this future event, we resorted to simulating this event in the Doha road network using a microscopic traffic simulator; the INTEGRATION software [9]. The software accounts for all road network parameters including traffic signals, queuing, and navigation. Therefore, it generates traffic conditions close to the real world.

The remainder of the paper is organized as follows. Section II is the related work. Section III describes the problem of traffic prediction on road network graphs including the conversion to line graph and the different weight manipulation methods to preserve the original graph properties. Section IV presents our case study on the Doha road network, the simulation methodology we utilized, the generated dataset, and the prediction results. Finally, the conclusions are discussed in Section V.

II. Related Work

Due to the importance of traffic prediction in practical systems, several research works have been proposed in the literature. Outstanding results depend on the prediction technique used, where real graph-structured data is tackled in [4], spectral graph convolutional networks are analyzed in [10], and semi-learning through generalized transductive inference is presented in [11].

Special consideration is given to deep learning. For instance, the Diffusion Convolutional Recurrent Neural Network scheme in [5], can model traffic flow through the diffusion process on a directed graph. It is able to capture spatial dependencies by generating a graph through a bidirectional-random-walk-based approach. While being able to capture temporal dependencies through an architecture of encoder-decoder with importance sampling (in order to enhance long-term forecasting), this model is strictly dependent on the notion of edge-centric graphs.

An alternative architecture is presented in [12], which combines GRU, Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) to capture spectral-spatial-temporal features of geographical imagery. Instead of using a traditional convolutional window, the authors proposed a dilated convolutional window, which allows the exponentially enlarging field of view while not suffering from a multiplicative increase in the number of parameters; however, their model is adapted only to Euclidean structures.

Transductive node classification is used in [13] instead of inductive node classification, as an effective approximation method, showing the traffic speed forecasting through a specialized aggregator. Whereas, an alternative to exploring all representation of sub-spaces, they added a soft gate to assign normalized importance to each head. The scheme drawback is that the attention mechanism that needs to calculate the very complex spatial dependency weight between the nodes.

In [6], the authors proposed a novel model; High-Order Graph Convolutional Recurrent Neural Network, which addresses a drawback of vanilla GCN, as it is unable to learn to mix relationships between the class of neighbourhood. This is possible through k-hop graph convolution. Their graph convolutional-LSTM only needs to learn spatial and temporal dependencies of neighbouring nodes in a traffic network based on one step historical data. Nevertheless, this algorithm is an extension of Gated Convolutional Recurrent Network (GCRN) [14], which infer the use of gates like input and forget gates that are akin to LSTM or reset and update gates in GRU, with their associated drawbacks.

III. Traffic Prediction on Road Network Graphs

A. Road Network as a Directed Graph

In TLGCN, the road network is represented by a weighted directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V} = \{i : i = 1, 2, ..., n\}$ is a set of n vertices (i.e., intersections or topological change points), and $\mathcal{E}$ is a set of e directed edges (i.e., road links), i.e., $\mathcal{E} = \{e_{ij} : i, j \in \mathcal{V}\}$, where e_{ij} is the road segment from vertex i to vertex j. Moreover, the edge weights $\mathcal{W} = \{w_{ij}\}$, where $w_{ij} \geq 0$ for each $e_{ij} \in \mathcal{E}$ and zero otherwise. This weighted nxn adjacency matrix, $\mathcal{W}$, represents the graph structure. At any time step, the traffic flow parameters (i.e., speed, density, etc.) are denoted as $X^t \in \mathbb{R}^{e \times p}$, where p represents the number of edge features. Thus, the time series of the graph features is $X \in \mathbb{R}^{e \times p \times T}$, where T represents the number of observation time steps.

Given a graph $\mathcal{G}$, the traffic prediction model aims at learning a function $f(.)$ mapping $\hat{\tau}$ historical observations to future τ observations, as shown below:

$$[X_{\hat{t}_0}, X_{\hat{t}_0+1}, ..., X_{\hat{t}_0+\hat{\tau}}] \xrightarrow{f(.)} [X_{t_0}, X_{t_0+1}, ..., X_{t_0+\tau}] \quad (1)$$

Since, the T-GCN is a vertex-centric model, where the features X are associated with vertices, the road network graph has to be converted from an edge-centric graph to a vertex-centric graph, where these features are related to the nodes. In the literature, each research work addressed this issue separately based on the case and the dataset. We propose to use the line graph conversion technique to convert the road network edge-centric graph to its corresponding vertex-centric graph, as such, the conversion process would preserve the graph properties. Thus, the next subsections overview line graph and its characteristics, as well as some basic definitions of line graphs that we utilized to preserve the original graph structure and dynamics, in order to improve evaluation metrics of the T-GCN.

B. Line Graph and its Properties

Given a graph $\mathcal{G}$, its line graph $\mathcal{L}(\mathcal{G})$ is another graph, where each edge in $\mathcal{G}$ maps to a vertex in $\mathcal{L}(\mathcal{G})$. Moreover, two nodes of $\mathcal{L}(\mathcal{G})$ are adjacent, if and only if their corresponding edges are incident in $\mathcal{G}$. Fig. 1 illustrates the transformation on a small sample graph.

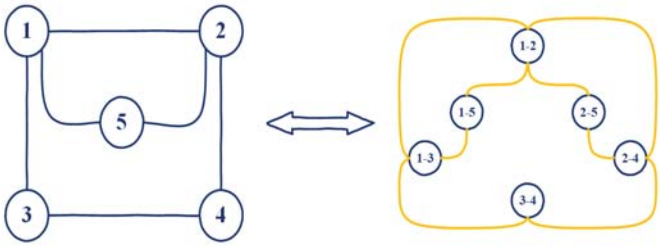

Figure 1: Line Graph of Undirected Graph

If $\mathcal{G}$ is a connected graph, $\mathcal{L}(\mathcal{G})$ is a connected graph. Nonetheless, if $\mathcal{G}$ is disconnected, it does not mean $\mathcal{L}(\mathcal{G})$ cannot be connected. A line graph has a bi-connected component (articulation point) contingent on the condition of an underlying graph having a bridge between the two endpoints, neither of them having a degree of one. In the context of graph theory, any network line graph preserves network properties, such as the small-world property, which allows very short walks between any non-neighbouring vertices as if they are clustered together in the same clique or near-clique [15].

Another generalization of line graphs is the directed line graph. For a directed graph $\hat{\mathcal{G}}$, its line directed graph has a node for each edge, where the directed edge from node i to j and the self-loop from i to i, will form two nodes in the line graph, where the common starting node is i [15].

C. Weighted Network Line Graph

For the line graph, we define the incidence matrix $\mathcal{Z}(G)$, where $\mathcal{Z}_{i\alpha}$ is 1 when the edge α is connected to node i and 0 otherwise, the edges are denoted by α and β where they are incident to vertex i. Thus $\mathcal{L}(\mathcal{G})$ is represented by its $e \times e$ adjacency matrix $\mathcal{C}(\mathcal{G})$ [16], where:

$$\mathcal{C}_{\alpha\beta} = \sum_{i} \mathcal{Z}_{i\alpha}\mathcal{Z}_{i\beta}(1 \ - \delta_{\alpha\beta}) \tag{2}$$

The Whitney isomorphism theorem [17] guarantees that the line graph $\mathcal{L}(\mathcal{G})$ encode graph $\mathcal{G}$'s topology, but it does not guarantee that the dynamics of $\mathcal{G}$ are preserved in $\mathcal{L}(\mathcal{G})$. Using the notion of bidirectional random walker, for an edge e in $\mathcal{G}$ that is connected to a node of degree $O(k)$, the T-GCN will travel over it $O(k^2)$ more frequently in the line graph $\mathcal{L}(\mathcal{G})$. Since a vertex in $\mathcal{G}$ with a degree k contributes to $k(k-1)/2$ edges, it will be assigned a higher importance factor when the GCN traverses the adjacency matrix of $\mathcal{G}$, $\mathcal{C}(\mathcal{G})$. A brilliant, yet a very natural solution to this problem is proposed in [16], by defining a new weighted line adjacency matrix $\mathcal{D}(\mathcal{G})$ of the form:

$$\mathcal{D}_{\alpha\beta} = \sum_{i,\ k_i>1} \left(\frac{\hat{\mathcal{Z}}_{i\alpha}\mathcal{Z}_{i\beta}}{s_i - w_\beta} (1 \ - \delta_{\alpha\beta}) \right) \tag{3}$$

where edges α and β are incident to vertex i have weights w_α, w_β and s_i is the degree of node i. The importance of dividing by strength s_i is to reduce the effect of prominence given to some vertices with high degree, and high strength by a factor of $O(s^{-1})$.

We upgrade this technique to improve the line graph representation in the TLGCN, compared to just assigning weights based only on the distance in T-GCN. This way, we can preserve the original graph structure and dynamics, which should help the GCN to better adjust its weights.

D. Temporal Graph Convolutional Neural Network

a) Spatial Dependency Modeling: In traffic forecasting, acquiring complicated spatial dependence is a central dilemma. CNN can only spatially localize on Euclidean spaces, for instance, pictures. On the other hand, a traffic network is a form of a graph. Thus, CNN cannot comprehend the complex topological structure, hence, its inability to capture spatial dependencies. Graph Convolutional Networks (GCN) is a generalization of CNN and has successfully been adopted in many applications, including document classification, molecular synthesis, and image classification, among others.

In order to discern how GCN works we define graph Laplacian as: $\boldsymbol{\Delta} \ = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\mathcal{W}\boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ is the degree matrix and $\mathcal{W}$ is the weight matrix. The above Laplacian matrix is a symmetrical one which admits Eigen decomposition in the form $\boldsymbol{\Delta} = \boldsymbol{\Phi\Lambda\Phi}^{\mathsf{T}}$, through non-negative eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, which act like frequencies, and orthonormal eigenvectors $\boldsymbol{\Phi} = (\phi_1^{\mathsf{T}}, \ldots, \phi_n^{\mathsf{T}})$, that act as standard Fourier atoms is Hilbert space, it allows Fourier decomposition as $f = \boldsymbol{\Phi}\widehat{\boldsymbol{f}}$, where the graph Fourier transform of f is $\widehat{\boldsymbol{f}} = \boldsymbol{\Phi}^{\mathsf{T}} f$. Therefore, a convolutional operation can be defined similarly in Euclidean situation as $f * g = \boldsymbol{\Phi}\left(\widehat{\boldsymbol{f} \cdot \boldsymbol{g}}\right) = \boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathsf{T}} f \cdot \boldsymbol{\Phi}^{\mathsf{T}} g)$. Henceforth, GCN model is designed as a concatenation of stacked convolutional layers of the form:

$$\widetilde{f}_l = \xi \left(\sum_{l'=1}^{c'} \boldsymbol{\Phi}\widehat{\boldsymbol{G}}_{ll'}\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{f}_{l'} \right), \ l = 1, \ldots, \ c, \tag{4}$$

where the output and the input channels are denoted by c, c', respectively, and ξ is the activation function. The filter is represented by $\widehat{\boldsymbol{G}}_{ll'}$, which is just a diagonal matrix of spectral multipliers. Among the significant disadvantages of this design compared to classical Euclidean CNNs is the elevated computational complexity $O(|\mathcal{E}|^2)$ owing to the expense of calculating the forward and inverse graph Fourier transform, resulting in dense matrix multiplication per layer.

The length of the graph Fourier transform window segment S trades the temporal resolution for frequency resolution. Thus, by increasing the size of S, better resolution in the frequency domain can be achieved, hence more details extracted. Nonetheless, as the window size decreases, the temporal resolution is enhanced. Consequently, any transition in the frequency is proportionally more vivid in terms of graph signals.

A Chebnet filter is presented in [11], which is a CNN structure that has a Chebyshev basis with polynomial filters.

These filters can be computed by applying powers of the graph Laplacian. Accordingly, Eigen decomposition is avoided. Consequently, the computational complexity is reduced from $O(|\mathcal{E}|^2)$ to $O(|\mathcal{E}|)$. The work in [4] limits the polynomial order to 1, hence, a further simplification of the Chebnet, namely Graph Convolutional Network (GCN).

In the TLGCN approach, we use a 2-layer GCN model. However, weighted line graph adjacency matrix is computed based on Eq. (3). So, the propagation rule is formulated as: $f(X, A) = \sigma\left(\hat{A}\ ReLU\left(\hat{A}XW_0\right)W_1\right)$, where X denotes the features matrix, A is the proposed weighted line graph adjacency matrix. $\hat{A} = \boldsymbol{D}^{-\frac{1}{2}}\ \mathcal{W}\boldsymbol{D}^{-\frac{1}{2}}$ where $\mathcal{W} = A + I_N$ denotes a matrix with self-loops structure, W_0 and W_1 represent the weight matrices of the first and the second layers, respectively, and σ is the activation function.

b) Temporal Dynamics Modeling: To model temporal dependency, we use GRU [3], which is a robust variant of RNNs, that helps overcome various drawbacks of vanilla RNNs like vanishing and exploding gradients by using a structure of gated units to memorize long-term information. GRU is faster to train and requires less data for achieving consistent results when compared to LSTM. GRU unit is constructed by only two gates, the reset and the update gates. Intuitively, the update gate defines how many memories need to be stored, and the merge of fresh input with the prior memories is governed by the reset gate.

IV. Case Study: FWC 2022 on Doha Network

Qatar will host the FWC 2022. So, in order to improve the transportation network performance during this event, authorities need to apply better route planning and navigation. Thus, efficient real-time traffic forecasting is essential. In order to improve the prediction accuracy, in our case study, we apply the traffic prediction using the line graph with modified weights based on Eq. (3). Notice that even we consider the FWC 2022 use case, any other massive event would benefit from our proposed technique.

Because we are working with a future event, we do not have real data for the traffic conditions. Therefore, we resorted to using microscopic simulation to generate the road network traffic conditions after the match.

This section describes the simulation methodology we used to build the network, and to and generate the traffic flows.

A. The Simulation Software

For the simulation purpose, we use the INTEGRATION software [9], which is an agent-based microscopic traffic simulator and assignment framework. INTEGRATION is characterized by its modelling accuracy for vehicle mobility and the associated parameters, such as speed, traffic density and travel time. The program traces each car at a resolution step of one deci-second from its origin to its destination. Its accuracy is reasoned to its capability to replicate vehicle longitudinal motion using the Rakha-Pasumarthy-Adjerid (RPA) car-following model [18], which captures vehicle steady-state car-following behaviour using the Van Aerde model, in which, movement from one steady-state to another is constrained by a vehicle dynamics model and safety constraints. Its microscopic nature allows it also to model the vehicle lateral motion that considers the surrounding vehicles and their speeds. INTEGRATION also accounts for traffic lights and their impact on the speed and travel time. The scalability of the INTEGRATION software is a key feature that enables us to model up to 46000 vehicles and track them in the network scenario.

B. Road Network and Traffic Setup

To achieve realistic traffic condition dataset, we developed the Doha road network shown in Fig. 2.

To build this network, data is collected from different sources. The first is the road network Geographic Information System (GIS) Shapefile, where Doha city shapefile is used to generate the network nodes and links. The second is the OpenStreetMap (OSM) website. The intersection data from OSM is used to extract intersection traffic control information including the traffic control methods (stop sign, yield sign, or traffic signals). The number of phases for each traffic signal and its timing information are obtained based on field observation. The third source is Google and ArcGIS which are utilized for validating road attributes of the road links, including number of lanes, one-way streets, and speed limits. The generated simulation road network has 301 road links, 169 nodes, and 11 traffic lights.

The road network graph $\mathcal{G}$ is constructed and line graph $L(\mathcal{G})$ is created. The resulting line graph has an adjacency matrix of size 301×301 and the weights are adjusted using Eq. (3), which encodes edge weights as the proximity between the link midpoints measured by the road network distance, in addition to the effect of the strength of the nodes of the original graph to reserve dynamics of a random walker moving around the graph.

In regards to the traffic setup, we generated the time-dependent Origin-Destination (O-D) traffic demand matrices every 15 minutes using the car counts collected from OSM. The car counts are converted to O-D traffic rates (car flow rate from origin to destination) using the maximum likelihood estimation.

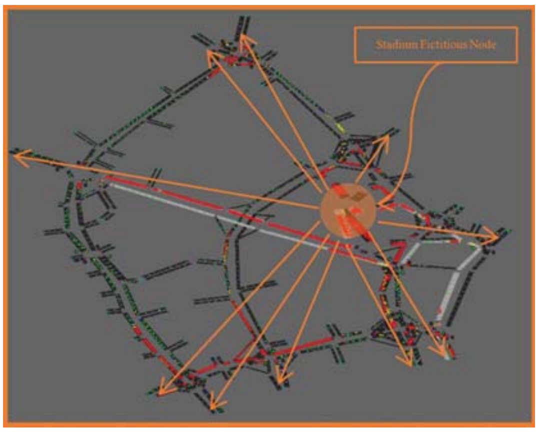

Figure 2: Doha Network and sport Event Case Study

To model the FWC 2022 events, the red area in Fig. 2 is used as a stadium from which, vehicles will depart to different network destinations after the match (as shown by the yellow arrows). This traffic is distributed over 10 network destination points.

We use two main cases for the traffic; the calibrated traffic case which represents the daily regular traffic conditions, and the stadium case, in which, we assume there is a match and vehicles leave the stadium towards their destinations, directly after the match, producing distinct traffic patterns.

1) Calibrated Traffic Scenario: This scenario represents the daily traffic condition in the Doha network. As aforementioned, we use the vehicle count from the OSM website to create the traffic demands between each O-D pair of nodes. The total number of vehicles generated for this case based on the traffic calibration is about 18,000. For each traffic flow, the inter-departure interval between vehicles is computed assuming exponential distribution.

2) The Stadium Case Study: For this use case, we add a fictitious node, as shown in Fig. 2, that represents the stadium along with additional five edges, making the line graph adjacency matrix of size 306×306. In this case, because of the high traffic from the stadium, the calibrated traffic is approximately 18000 vehicles and the stadium traffic generates approximately 28000 vehicles. Thus the total number of cars is approximately 46000 vehicles, which results in a very

Table I: Impact of weight modifications (TLGCN vs. T-GCN)

Prediction Horizon	Metric	Calibrated Doha Network			Stadium Case Study		
		T-GCN	TLGCN with $\frac{1}{k_i-1}$	*TLGCN with* $\frac{1}{s_i-w_\beta}$	*T-GCN*	TLGCN with $\frac{1}{k_i-1}$	*TLGCN with* $\frac{1}{s_i-w_\beta}$
1 Period	Accuracy	0.8202	0.852	**0.92**	0.8162	0.8462	**0.8834**
	R^2	0.3741	0.5761	**0.86**	0.5037	0.6524	**0.8621**
	Variance	0.3747	0.577	**0.8605**	0.5038	0.653	**0.86**
	RMSE	13.341	10.979	**6.0058**	13.715	11.478	**8.17**
	MAE	8.5936	5.9617	**2.7812**	8.3699	5.1042	**4.2**
2 Periods	Accuracy	0.8062	0.8378	**0.9039**	0.8059	0.8272	**0.8975**
	R^2	0.2785	0.4943	**0.7964**	0.4631	0.5745	**0.7957**
	Variance	0.2788	0.4956	**0.796**	0.4648	0.5932	**0.8**
	RMSE	14.3728	12.0321	**7.22**	14.449	12.8626	**9.98**
	MAE	9.2973	6.5245	**3.48**	8.6529	6.3209	**5.39**
3 Periods	Accuracy	0.8038	0.8287	**0.893**	0.7989	0.8196	**0.8378**
	R^2	0.258	0.4338	**0.7442**	0.4271	0.5392	**0.7365**
	Variance	0.26	0.4345	**0.75**	0.4396	0.5403	**0.7363**
	RMSE	14.555	12.714	**8.05**	14.957	13.414	**11.36**
	MAE	9.2242	7.0601	**4.06**	8.8764	6.4024	**6.46**

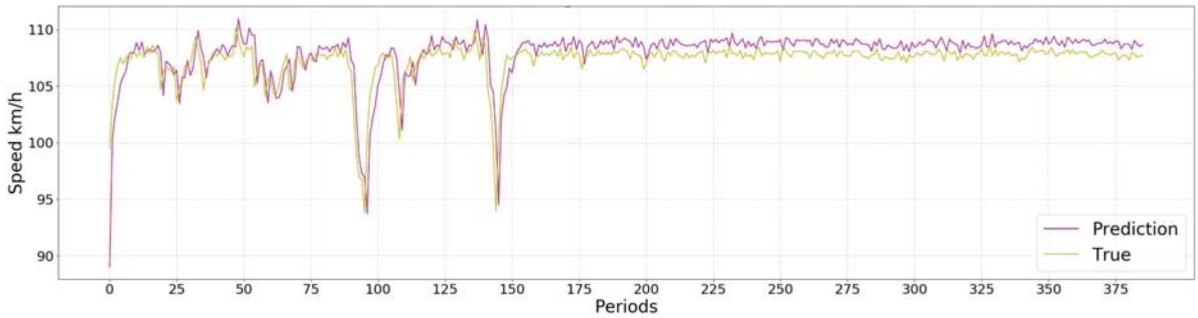

Figure 3: Sample high speed link prediction

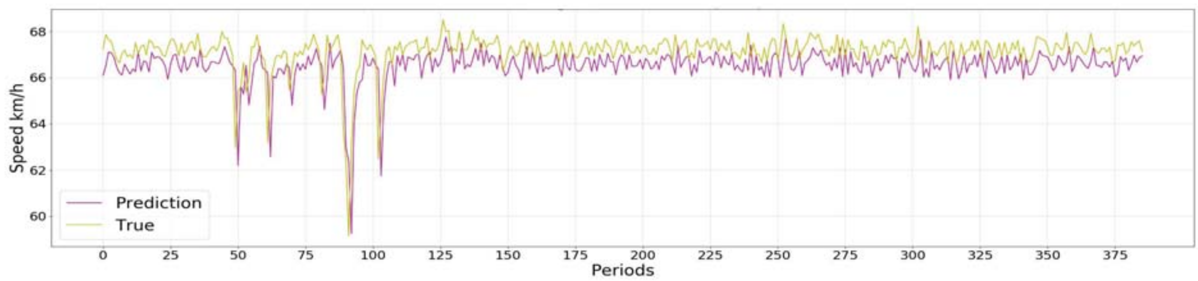

Figure 4: Sample low speed link prediction

congested network.

C. Simulation Output

The generated network and the calibrated traffic are used to generate the dataset over the graph network. The output of the simulation is a link-temporal-speed matrix of the form $S_t \in \mathbb{R}^{n \times e}$, where e is the number of edges in the network graph, and n is the number of time steps. S_t is the average speed over the edge at time i. This speed matrix represents the feature input X_τ to be used for model training and testing.

D. Prediction Results

Doha's datasets are partitioned for training (80%) and testing (20%). Table I compares the results for our proposed scheme, and their benchmarking to state-of-the-art technique. It shows how our proposal outperforms for all considered cases. The table only features the results of the Doha network in both cases with the finalized weight type of $\frac{1}{s_i - w_\beta}$. Thus, through preserving the topological and dynamical structure of the original graph $\mathcal{G}$, TLGCN reached an accuracy of 88.34%, 89.75%, and 83.78% for the three prediction horizons respectively, compared to 84.42%, 82.61%, and 81.61% for the original T-GCN distance-based weights.

Table I also shows a significant improvement in R^2 showing that the weight modification results in enabling the model to capture the changes in link speeds, which is important for many applications such as prediction based navigation.

Fig. 3 and Fig. 4 show the performance of TLGCN on two sample links from the calibrated Doha network dataset road links. Each of them has different characteristics. Fig. 3 features a high-speed link, which naturally has a low variation of speeds over time, where most of the traffic flow is close to the maximum road speed. Fig. 4, represents a hot road link with frequent abrupt drops in speed due to bursts of traffic on this specific link. Using the $\frac{1}{s_i - w_\beta}$ weight, the TLGCN was able to predict quite accurately in both cases with different rates. The two figures also show that the TLGCN is able to capture most rates of variations in the speed, which is reflected by the high R^2 value in Table I.

V. Conclusion

In this paper, we introduce a novel approach for traffic prediction that is suitable for large events like sports, musical or any other event that triggers massive vehicle departures from a specific location in time. It allows, through mathematical manipulation of weights, to deal with adjacency matrices based on edge-centric structured graphs, while ensuring the dynamics of the original graph is preserved as well as its topological structure with the transformation. Extensive experimentation shows that our proposed approach helps improve the T-GCN comprehension of the network structure, consequently performs better when adapted to our case study. The results were accomplished with much less training time and faster convergence. Besides, our proposed TLGCN model can be extended to more general Spatiotemporal community structure based graphs, such as emerging social networks, neural links, while preserving the topological and dynamical properties of these graphs.

Acknowledgment

This work was made possible by NPRP grant NPRP 9-185-2-096 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] QMIC. Qatar Traffic Report 2018. https://www.qmic.com/selections/qmic-launches-the-qatar-traffic-report-for-2018/. Accessed: Oct., 2019.

[2] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[3] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

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

[6] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *arXiv preprint arXiv:1802.07007*, 2018.

[7] A. Elbery, H. Hassanein, N. Zorba, and H. Rakha, "VANET-based Smart Navigation for Vehicle Crowds: FIFA World Cup 2022 Case Study," in *2019 IEEE Global Communications Conference: Selected Areas in Communications: Internet of Things (Globecom2019 SAC IoT)*, Waikoloa, USA, Dec. 2019.

[8] A. Elbery, H. S. Hassanein, N. Zorba, and H. Rakha, "VANET-Based Smart Navigation For Emergency and Special Events," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Sep. 2019.

[9] H. A. Rakha, K. Ahn, and K. Moran, "Integration framework for modeling eco-routing strategies: Logic and preliminary results," *International Journal of Transportation Science and Technology*, vol. 1, no. 3, 2012.

[10] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[11] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[12] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 924–935, 2018.

[13] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[14] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 362–373.

[15] F. Harary, "Graph theory addison-wesley reading ma usa," 1969.

[16] T. S. Evans and R. Lambiotte, "Line graphs of weighted networks for overlapping communities," *The European Physical Journal B*, vol. 77, no. 2, pp. 265–272, 2010.

[17] H. Whitney, "Congruent graphs and the connectivity of graphs and the connectivity of graphs," *Amer. J. Math. v54*, pp. 150–168.

[18] H. A. Rakha, P. Pasumarthy, and S. Adjerid, "A simplified behavioral vehicle longitudinal motion model," *Transportation Letters: The International Journal of Transportation Research*, vol. 1, no. 2, pp. 95–110, 2009.