CS 224W Final Project Report
Spatial-Temporal Model for Traffic Forecasting on Road Network

Hua Shao
Dept. of Civil and Environmental Engineering
Stanford University
huashao@stanford.edu

Veronica Peng
Dept. of Computer Science
Stanford University
tpeng24@stanford.edu

1 Introduction

Traffic forecasting is the prediction of traffic volume and speed on the road for a particular time. Accurate traffic flow prediction is critical for developing an intelligent transportation system. It is applied in a wide range of areas including traffic management during peak hours, urban system planning, and route optimization in navigation. In recent years, the need for efficient and accurate traffic predictions grows rapidly as the number of vehicles in big cities increases. For San Francisco in particular, the vehicle traffic entering the city increases by 27% from 2010 to 2018. (SFMTA, 2018) Our project thus predicts the future 1-hour traffic speeds in the San Francisco Bay Area given historic traffic speeds and the underlying road networks.

A challenge for traffic speed prediction is to capture the spatial and temporal dynamics of traffic together. Spatial correlation means the traffic speed is influenced by topological road structure. Intuitively, the traffic speed of a particular road segment is often influenced by traffic from the roads leading to the current road segment. We capture the spatial correlation by modeling the road network as a graph. For the temporal dynamics, the traffic speed on a particular road can vary over time. For instance, the traffic speed may become slower than usual in rush hour and accidents. We capture the temporal dynamics by using historical traffic speed data with recurrent neural networks. Therefore, in this project, we will address the following challenges: 1. Efficient spatial representation for city transportation network given mobile app navigation data; 2. Time series prediction with non-linear temporal dynamics.

2 Related Works

The existing approaches to traffic forecasting can be categorized into two divisions: model-driven approaches and data-driven approaches. Model-based approaches represent the relationship between traffic volume and speed with comprehensive system modeling based on prior knowledge. However, it is often hard to capture all the complex factors influencing the traffic and it requires significant computation time to construct the model.

Data-driven methods instead infer the variation in traffic based on the statistical regularity of data. Among the data-driven methods, deep neural network models capture spatial and temporal dynamics well and yield the best accuracy in prediction compared to other approaches. The deep learning approaches mostly share a similar solution: represent the spatial dependency with the graph and then capture the temporal dependency with the recurrent neural network or its variation. The differences come from their graph construction methods and assembly mechanics of spatial and temporal models:

Li et al. (2018) proposes Diffusion Convolutional Recurrent Neural Network (DCRNN), which uses the random walk to model the diffusion of traffic on the road network, and applies Gated Recurrent Units (GRU), a variation of Recurrent Neural Network (RNN) to capture the temporal characteristics. The paper breaks ground in traffic forecasting because past models used to represent the spatial dependency with Euclidean distance but DCRNN captures the spatial correlation with a topological structure for the first time, significantly increasing the prediction accuracy. Despite the high accuracy, the DCRNN model suffers substantially in computation time even with a small graph with speed data at about 300 sensor points on a highway, which is significantly less complicated than the real-world road network. It is thus hard to scale the model to make speed predictions for real road network.

The above approach uses speed data collected from highway sensors. Thus, the model represents the sensors as nodes and road segments between sensors as edges. However, sensors are usually available only on the highways. For normal roads within the city, speed data is collected through users' navigation data from the phone when they travel on the road. In this case, the speed data is the feature of a road segment instead of a sensor point. The difference in data often leads to a different graph construction approach. For instance, Zhao, et al. (2019) use the taxi speed data from Luohu District of Shenzhen, China. This paper constructs the road network differently from the other deep learning models by representing the road segments as nodes and road junctions as nodes. After the construction of the graph, the paper proposes a similar approach: Temporal Graph Convolution Network (T-GCN) model, which applied Graph Convolutional Network (GCN) to capture the network spatial dependency and Gated Recurrent Unit (GRU) to capture the temporal dependency. Similar to Li et al. (2018), the model suffers in computation time.

In order to improve the computation efficiency, Yu, et al. (2018) proposes the Dynamic Spatio Temporal Graph Convolutional Neural Networks (DST-GCNN) model, which uses Graph Convolutional Neural Network (GCNN) to capture the topological spatial correlation and Gated Linear Units (GLU) to build temporal CNN layers. Unlike traditional CNN that can only work in Euclidean space, GCNN is able to capture topological structure in non-Euclidean domains. Furthermore, GCNN is able to achieve linear computational complexity and constant learning complexity as standard CNNs. The Gated temporal CNN layers also achieve better computation efficiency by reducing ordered operations of traditional recurrent models so that the computation can be parallelized. However, with the GCNN approach, the graph, and thus the traffic speed, are undirected. In contrast, most real-world road networks are directed since they have two-way traffic. The traffic speed in both directions can vary a lot at times. For instance, at peak time on weekday afternoons, the traffic speed from San Francisco to San Jose is much slower than that from San Jose to San Francisco since a lot of people returning home from the city. However, since the graph is undirected, the speed difference in different directions is not reflected.

To contribute to the research in traffic forecasting, our team would like to expand the state-of-the-art techniques of traffic forecasting to make it applicable to predict real-world road network while keeping the directional characteristic of speed.

3 Dataset

3.1 Dataset Overview

We use the free Speed Data published by Uber Movement. The dataset provides the average speed of Uber drivers on a given road segment for each hour of each day in June 2019. Each row of the data includes when the speed is recorded, the road segment on which the speed is recorded, road junction ID at both ends of the road segment, average speed, and its standard deviation. The given speed is the average speed for all Uber vehicles passing the road segment within the one-hour time slot. The speed is only provided when at least 5 unique drivers pass the road segment within that hour, making the speed data more representative.

Compared to the datasets used by prior papers, the Uber Movement Speed Dataset provides speed for a more comprehensive road network. Instead of just having speed data from highway sensors, we now have a massive dataset with 878,236 road segments, including many normal city streets. However, the dataset doesn't provide the length of each road segment, which is given in the datasets used by prior papers. Thus, it is hard to find the topological distance between road segments to increase prediction accuracy.

3.2 Data processing

We processed the time series data into input and label by applying the method of "Sliding Window". The basic idea is to group the data into sequences as inputs and store their corresponding labels. A lookback period is set to determine how far back we need to consider as the inputs. The sequence length or lookback period is the number of data points in history that the model will use to make the prediction. The label will be the next data series in time. In this project, lookback values of 3 and 10 are used to extract different number of node features.

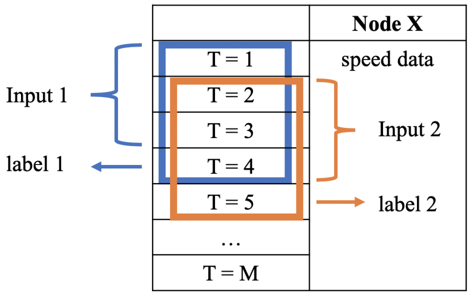

Figure 1: Illustration of Processed Data

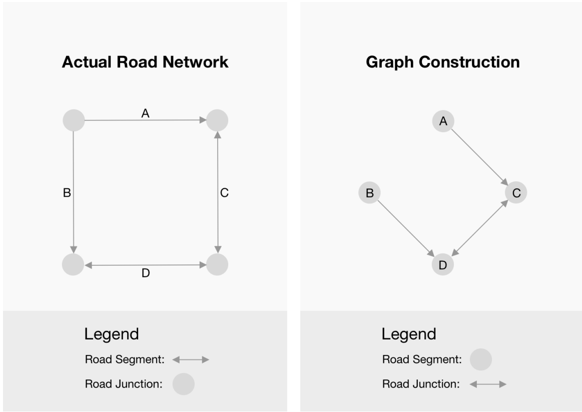

Figure 2: Graph Construction

In this project, 2019 June hourly speed data, a total of 720 time-series records are processed as inputs and labels. The historical data are normalized by its Max and Min value to reach better prediction results. The predicted results are scaled back in the evaluation step to calculate the evaluation metrics. The dimension of input is [number of time series record - lookback, number of nodes, lookback], the label has dimension [number of nodes, 1]. Then the data is split into training and testing data with ratio 9:1. In this project, we will only split the data into the train/test in the time domain. In the spatial domain, the full graph will be used to represent the topological structure. Although the amount of training set is not very large, the algorithm could still provide reasonable results.

4 Road Network

4.1 Graph Construction

Road Network G. A directed unweighted graph $G = (V, E)$ is adopted to describe the topological structure of road network. In this graph, we define each road segment as a node. The set of road segments is defined as $V = v_1, v_2, \ldots v_N$, where N is the number of nodes. Edges are defined as the connectivity between road segments. Intuitively, there would be an edge from a road segment to another road segment if traffic can flow from the former to the latter. With this graph construction, we keep the directional characteristic of traffic speed.

Figure 2 illustrates our graph construction method with a small road network. On the left is the real-world road network, and on the right is the graph we constructed for this road network. We see road segments A and B are one-ways, and C and D have two-way traffic. Since A is a one way, the traffic can only flow from A to C, but the traffic from C cannot flow into A, so there is only an edge from A to C, but no edge from C to A. The same graph construction rule applies to other road segments.

Since Uber Movement only presents the speed for road segments within a particular hour if more than 5 cars pass the road within a period of time, a large number of road segments don't have speed data available at a certain time. We originally used 0 to represent the speed when the data is unavailable in an hour. However, the representation is erroneous as no speed record doesn't mean there is no traffic flow from upstream, and considering the vulnerability of Graph Neural Network to noise in graph data, this misrepresentation could reduce the accuracy of prediction results. Zhao et al. 2019 conducted the perturbation analysis and found out that the GCN model could predict poorly at the peak values with more noise data into their model. Therefore, we finally construct the road network graph only with road segments with speed data at all times in the final report in June, 2019.

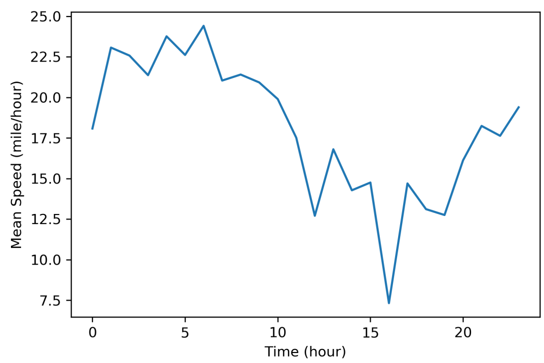

Figure 3: A road with speed record throughout a day. We see the speed is low in the afternoon rush hour, and high in the early morning, and late night.

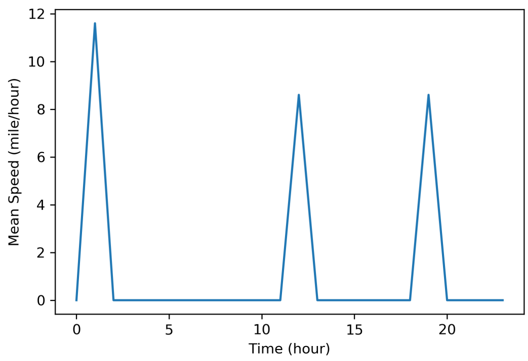

Figure 4: A road where speed record is not always available. The speed is 0 when data is missing.

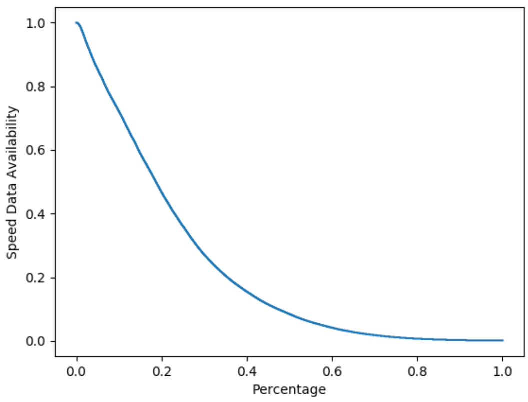

Figure 5: Y-axis represents the percent of speed data available for a road. Y = 1 means the road has data at all times. X-axis represents the percentage of the roads having the data available at least y% of the time.

There are in total 394 such road segments. Among these road segments, 17 of them are not connected to any other roads, so we eliminated these roads from the graph in order to make better use of the graph information for each node. We have 377 road segments in our graph after the elimination. As shown in the geo-spatial visualization of these nodes, most of them are located along the bay area highway or the downtown of San Francisco.

4.2 Graph Properties

In our graph, there are a total of 377 nodes and 337 edges. Unlike the social network graph where the degrees of each node can vary tremendously, the degrees of nodes in road graphs are usually very limited due to the common grid-like road structure in cities and the availability of speed data. We see in the graphs on the right that most nodes in

Figure 6: Geospatial visualization of selected roads in the Bay Area

the graph have 1 in-degree and 1 out-degree. The degree distribution is expected given that we only keep big roads that have speed data available all the time and usually big roads seldom intersect with each other.

5 Methodology

5.1 Overview

Our goal is to predict the traffic speed in a certain period of time given the historical traffic speed and the road network. In this report, we will elaborate how to construct the Graph Convolution Network (GCN) and Grated Recurrent Network (GRU) to conduct traffic prediction. In the first step, GCN captures the topological structure. It takes the graph structure and historical speed data as node attributes to generate the embedding for each node. This resultant embedding is integrated with spatial features. The second step uses GRU to capture the temporal dynamics. It takes the embedding from GCN as input and makes predictions about future speed. Overall, the spatio-temporal traffic prediction can be summarized as

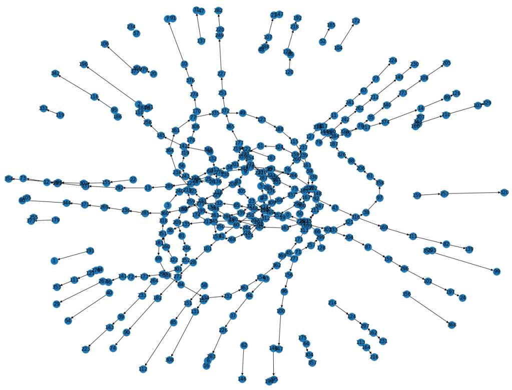

Figure 7: Visualization of constructed graph

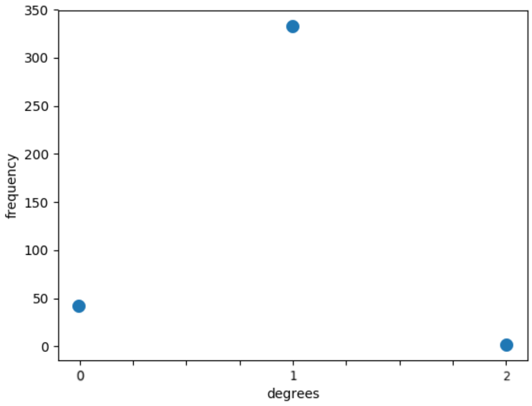

Figure 8: In-degree distribution

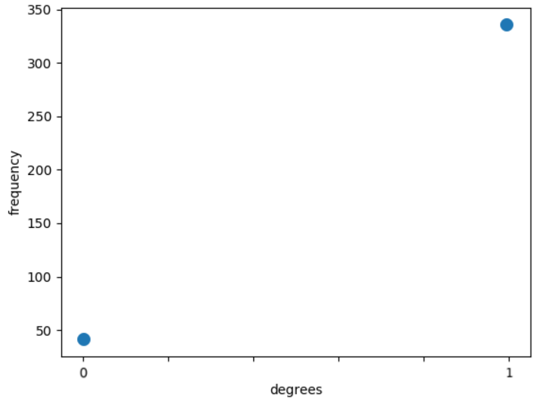

Figure 9: out-degree distribution

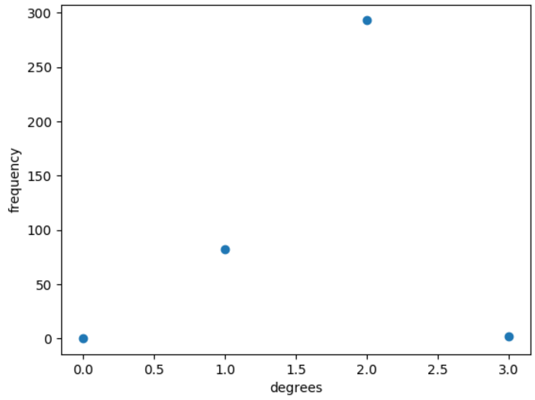

Figure 10: Total degree distribution

learning the mapping function f given road structure G and historical speed feature matrix F, to predict the speed X in the next T time steps: $[X_{t+1}, ..., X_{t+T}] = f(G; [X_{t-n}, \ldots, X_t])$

5.2 Spatial dependency modeling with GCN

The spatial dependency is the critical component in traffic prediction as the traffic speed of a particular road segment is often influenced by traffic from the roads leading to the current road segment. The topological spatial correlation can be captured by modeling the road network as a graph, and the flow of traffic can be simulated by the message passing between neighbors in GCN. The input node feature for each road segment is the historical speed. We set up a "lookback" period to extract the historical traffic speed. For example, if the lookback period is T hour, the feature matrix will be $N \times T$, where each column represents the historical speed for each node up to the target time. These features are used to construct node representation for message passing. The edge index matrix is invariant throughout the training. The output for the GCN layer is the embedding for the road segment.

The default GCNConv layer from pytorch geometric is used to construct the GCN model in this milestone. The message passing at each layer can be summarized as:

$$h_v^k = \sigma(W_k \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + B_k h_v^{k-1})$$

where $\sum$ is the aggregation function (which is "sum" in the case for GCNConv layer), W_k is the weighted matrix at the kth layer, and B_k is the bias at the kth layer.

5.3 Temporal Dependency modeling with GRU

Temporal dependency plays a significant role in the time-series problem. The recurrent neural network is usually applied to process sequential data.

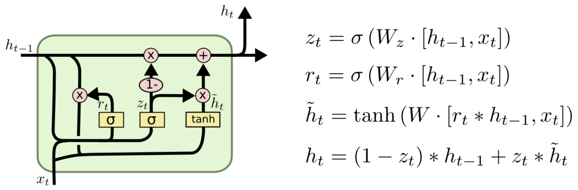

Figure 11: Total degree distribution

The variants of the recurrent neural network, like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) have proven to be able to effectively retain long-term dependencies in sequential data and solve the vanishing/exploding gradient problem of traditional RNNs. Compared with LSTM, GRU has a relatively simpler structure, faster training ability, so GRU is chosen for temporal dependency modeling in this project. The structure of the GRU allows it to adaptively capture dependencies from large sequences of data without discarding information from earlier parts of the sequence. These gates are responsible for regulating the information to be kept or removed at each time step. For a GRU cell, h_{t-1} denotes the hidden state at time $t-1$; x_t denotes the traffic information at given time t; r_t is the reset gate, which is used to control the degree of ignoring the status information at the previous moment; u_t is the update gate, which controls the degree to which the status information at the previous time is brought into the current status; c_t is the memory content stored at time t; and ht is output state at time t. The graph above illustrates a GRU cell.

Wang et al. (2018) claims that transportation systems are time-variant, and the propagation process does not need much time to reach a stable state, so they only select one time-step to capture dynamic relations. In this report, we think set up only one time-step back is not sufficient to capture the temporal dependency, so we set the lookback period to be at least 3 hours.

5.4 Loss function, Optimizer, and Evaluation Metrics

The goal of learning is to minimize the error between the real traffic speed on each road segment and the predicted values, which is denoted as Y_t and Y_t', respectively. Then the Mean square error loss function is used constructed using the equation:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_t - \widehat{Y}_t)^2$$

Adam optimizer is used as the training optimizer. The learning rate of 0.0001 and 0.0005 are tested in GCNGRU model training. The evaluation metrics in this project would be the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_t - \widehat{Y}_t)^2}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_t - \widehat{Y}_t|$$

5.5 Overall algorithm

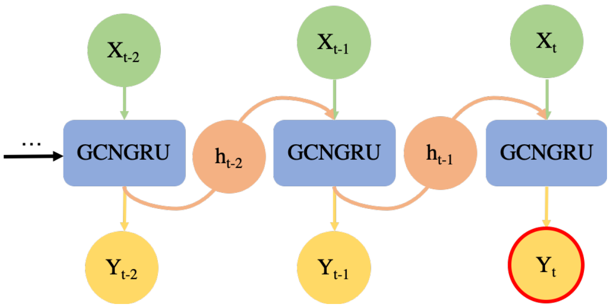

Figure 12: Mechanism of spatio-temporal prediction with GCNGRU cell.

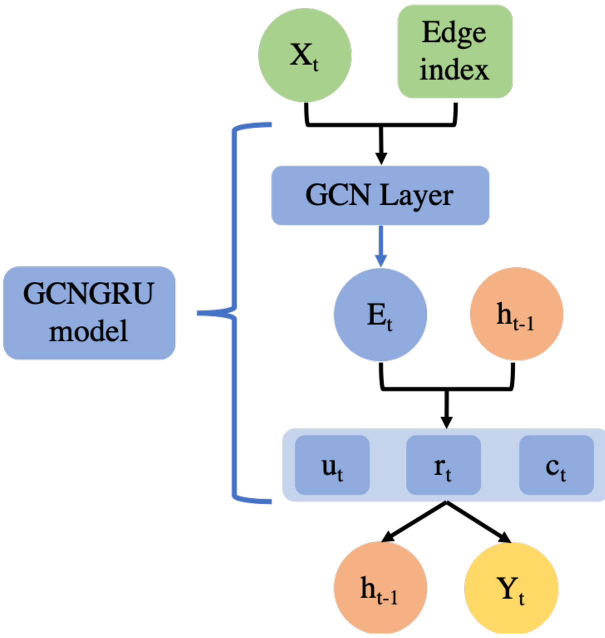

Figure 13: The internal structure of GCNGRU model, E_t refers to the resultant Embedding matrix from GCN layer.

In order to capture the spatial and temporal dependency at the same time, we created a "GCN-

GRU" model based on graph convolutional network and gated recurrent units. For each data from train data loader, input data needs to be transformed into "torch_geometric.data" format as feature matrix, which share the dimension [batchsize, number of nodes, input feature]. The GCN layer takes the transformed feature matrix and the edge index matrix with dimension [2, number of edges], which is an invariant, as inputs to compute the embedding for each node with dimension [number of nodes, hidden dimension].

Then the GRU layer will take the embedding as input and output a result with dimension [number of nodes, 1], and hidden results in certain time steps. As shown in Figure 12, at time $t-2$, input matrix is passed into the GCNGRU cell, and the output is Y_{t-2} and h_{t-2}. Due to the characteristics of GRU layer, h_{t-2} will take as another input into the GCNGRU cell in the next time step. At the desired time step t, the result of GCNGRU cell will be treated as the prediction values, and the parameters of GCNGRU model will be updated by Adam optimizer.

6 Experiments

6.1 The Baseline: GRU Model

In the baseline evaluation, the spatial correlations between the roads are ignored and only the temporal dependency among the roads are considered in training. To fully separate the spatial dependency, each road is modeled using GRU. We want to test if the historical data itself will be a good predictor for traffic forecasting.

As a reminder, the lookback value is the number of timesteps used to predict the next time step value. Due to the relative simplicity of the model, the learning rate is set to be 0.001 and the other hyper-parameters are tuned to reach its best performance. The lookback values of 3 and 10 are used as the number of features in this baseline model and the RMSE and MAE are shown in Table 1. With more features in the input data, the test error decreases.

6.2 The GCNGRU model

Then the full GCNGRU model is used to capture the spatial-temporal dependency to predict the future traffic speed for all nodes in the graph at the same time. The lookback value of 3 and 10 are used to construct input data with different numbers of features. The training time is significantly

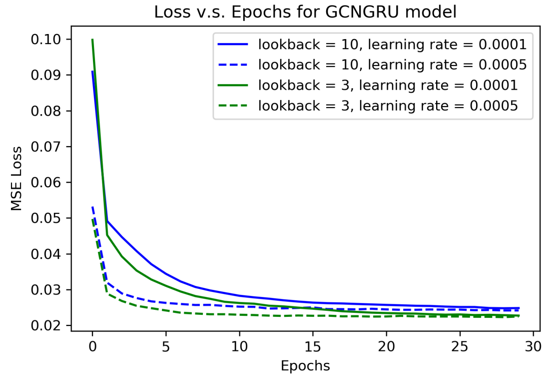

Figure 14: Loss over Epochs for GCNGRU models

longer than the baseline model, so only 30 epochs are used in the training. The RMSE and MSE losses in training are shown in Figure 14. The evaluation metrics results for GCNGRU model with 0.0001 learning rate is relatively worse, since given epoch number, it has not been trained to its best, so only learning rate 0.0005 conditions will be discussed in this report. As shown in Table 1, the GCNGRU model reaches the best performance under all evaluation metrics, proving the effectiveness of the spatio-temporal traffic forecasting. At lookback equals 3, GCNGRU reduced about 10% and 17.9% test error under RMSE and MAE metrics, respectively; At lookback equals 10, GCNGRU reduced about 2.3% and 1.3% test error under RMSE and MAE metrics, respectively.

To better understand the performance of the GCNGRU model, we select the roads in different regions and visualize the prediction results of the test dataset. In the region where the upstream and downstream are clearly defined, the GCNGRU model will outperform the GRU model. Figure 15 shows the 4th Street section between Howard Street and Folsom Street. Its upstream and downstream roads are shown in Figure 17. Compared with the prediction from GRU, GCNGRU model could capture the peaks of speed data better. Similarly in Figure 16, the GCNGRU outperforms GRU on a segment on Nimitz Freeway in the East Bay Area, where its upstream and downstream are all along the same Freeway.

On the other hand, the GRU model could sometime outperform the GCNGRU model in specific regions. In Figure 18, Road 58 is the segment along John B Williams Freeway, and in the graph, it only has one out-degree, in which the GCN layer could not aggregate its neighbors' feature in this

Model	GRU	GRU	GCNGRU	GCNGRU
Lookback	3	10	3	10
Learning Rate	0.001	0.001	0.005	0.005
RMSE	3.582	3.325	3.224	3.248
MAE	2.774	2.414	2.278	2.382

Table 1: Model Performance

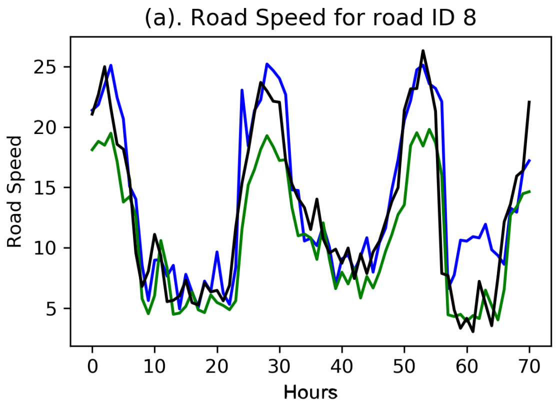

Figure 15: When GCNGRU Model outperforms GRU Model, Example 1

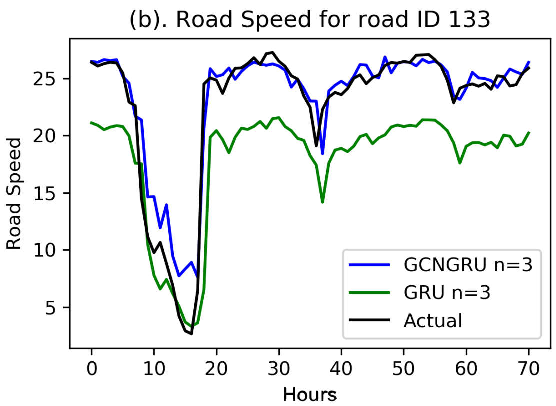

Figure 16: When GCNGRU Model outperforms GRU Model, Example 2

directed graph. About 11.1% of the nodes in the graph only have one out-degree, and these data would degrade the performance of the GCNGRU model.

Figure 19 shows the target and predicted speed values for roads on 28th June at 10 am using the GCNGRU model. It should be noticed that the blue dots represent the road speed is low, which refers to the traffic congestion. The GCNGRU model could correctly predict traffic congestion in the downtown area of San Francisco. Additionally, the GCNGRU model performs well in prediction of speed near the intersection of highways,

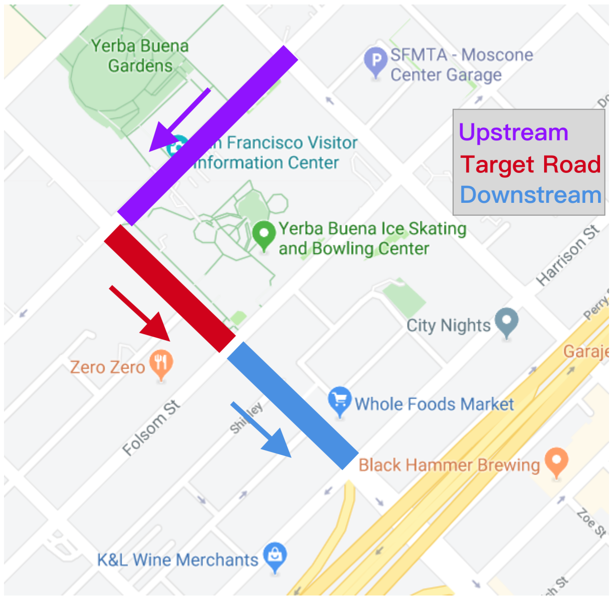

Figure 17: Upstream and Downstream for the 4th Street

where the topological structure is relatively complex, for example, the intersection between 280 and 101 in the south part of San Francisco. The prediction for the road at Oakland where merging 980 and 880 is not accurate enough since that road does not have upstream in our graph. Overall, the GCNGRU model could produce accurate traffic forecasting with spatio-temporal dependence.

7 Discussion

In this report, we are trying to balance between minimizing the data noise and maximizing the effectiveness of the topological structure. As shown in Graph Construction section in this report, we only select the roads with a full historical data records in June 2019 to filter out the data with zero values, which are defined as the noise. The advantage of this process is to make sure there are no anomalies in the input data, since GCN's vulnerability to the noise of data. However, this action sacrifices the informative road structure given by Uber Movement. Originally, there are 878,236 road segments, including the main road and minor

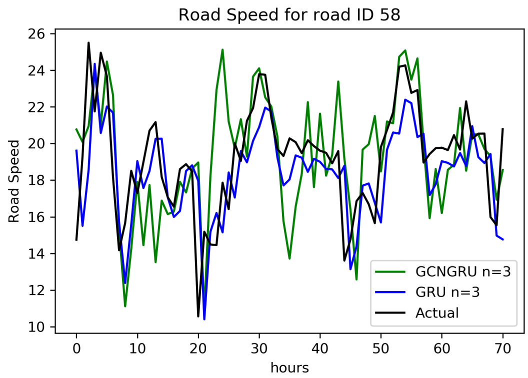

Figure 18: When GRU Model outperforms GCNGRU Model

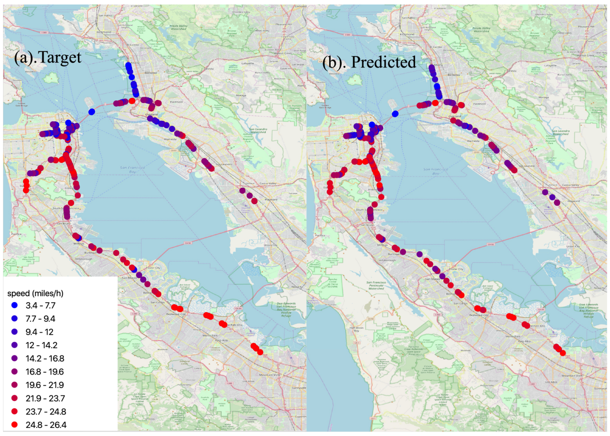

Figure 19: Target and predicted speed values for roads on the 28th of June at 10am

streets, but eventually, we only consider 377 nodes with 337 edges connecting them.

The results show that the GCNGRU model only outperforms the GRU model in test error by a small amount, especially at case lookback equals 10, and we are not satisfied with that. The limitation of the GCNGRU model comes from the relative simplicity of the graph structure of the road. The key concept of GCN is to generate node embeddings based on local network neighborhoods, which means every node defines a computation graph based on its neighborhood. One limitation of GNNs is that some simple graph structure cannot be distinguished by conventional GNNs. Existing GNNs use non-injective neighbor aggregation functions, thus having low discriminative power. This means it is hard for GNN in traditional road structures to reach injective neighbor aggregation at every step of node representation. Graph isomorphism Network (GIN) (Xu+ICLR'2019) could be the solution to this problem, but it is out of the scope of this project.

8 Conclusion

In this project, we propose a GCNGRU model, which combines the GCN and GRU layers, to predict traffic speed in the Bay Area. Uber Movement data are processed and filtered to produce roads with full speed historical data, and these roads are modeled as a graph network, in which roads are the nodes and edges are the connection relationships between roads. The historical traffic information is treated as the node attributes on the graph. The more historical information selected, the more node features used in the graph network. The GCN part is used to capture the topological structure of the graph to obtain spatial dependency, and the GRU part is used to capture the dynamic changes of speed trend to obtain the temporal dependency. The combined GCNGRU model is used to perform spatial-temporal traffic forecasting tasks. The trained GCNGRU model could outperform the baseline model in speed prediction when roads' upstream and downstream are cleared defined in the graph. Due to the sparsity of our selected road network, the prediction performance may degrade if the nodes in the graph only have one neighbor. In summary, we successfully overcome the two challenges mentioned in the overview: 1. We have obtained efficient spatial representation for the Bay Area network from Uber navigation data, and the spatial dependence does improve the prediction performance. 2. The proposed GCNGRU model captures the non-linear temporal dynamics and outputs accurate time series prediction results.

9 Future Work

1. More complete graph structure. The performance of the GCNGRU model is limited to the simplicity of the road structure. In the future, if full road speed data are available, it is possible to construct the full topological relation for the whole area to obtain spatial dependency and improve the prediction performance.

2. Node features. In this project, only the historical data are used as the node attributes, without any physical road attributes considered. More node features could be used as the attributes for the node in the graph. For

example, the length of the road, and width of the road, or even the POI (points of interest) near the roads.

3. Computational expense. The GCNGRU model training is time-consuming. On average it takes about 2.7 hours to train 30 epochs. In the existing literature exploring the GCN application in traffic forecasting, none of them utilized the graph with more than 300 nodes. Spatial-Temporal Graph-based model for traffic forecasting on massive graphs could be next hot topic.

4. Dynamic topological structure. In Wang et al. 2018, they proposed the dynamic spatio-temporal graph-based CNNs, in which the topological structure would change over time. This real-time dynamic graph structures could improve the computational complexity by utilizing the full graph in the peak time and shrinking the graph when traffic is small. This method is worth exploring because it could save computational resources.

References

[1] Jiahui Wang, Zhecheng Wang, Shuyi Yin. Efficient Traffic Forecasting with Graph Embedding. URL `http://snap.stanford.edu/class/cs224w-2018/reports/CS224W-2018-42.pdf`

[2] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng and Haifeng Li. T-GCN: A Temporal Graph ConvolutionalNetwork for Traffic Prediction. In *IEEE Transactions on Intelligent Transportation Systems*, 2019. URL `https://arxiv.org/abs/1811.05320`

[3] Menglin Wang, Baisheng Lai, Zhongming Jin, Yufeng Lin, Xiaojin Gong, Jianqiang Huang, Xiansheng Hua. Dynamic Spatio-temporal Graph-based CNNs for Traffic Prediction. *CoRR*, abs/1812.02019, 2018. URL `http://arxiv.org/abs/1812.02019`

[4] SFMTA. "San Francisco Mobility Trends Report 2018." Accessed November 11, 2019. URL `http://www.cnn.com/POLITICS/01/21/obama_inaugurated/index.html`

[5] Uber Movement Data. © 2019 Copyright Uber Technologies, Inc. URL `https://movement.uber.com/?lang=en-US`

[6] Wang X, Chen C, Min Y, et al. Efficient metropolitan traffic prediction based on graph recurrent neural network[J]. URL `https://arxiv.org/pdf/1811.00740.pdf`

[7] Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-driven Traffic Forecasting. *In International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SJiHXGWAZ`