Graph CNN + LSTM Framework For Dynamic Macroscopic Traffic Congestion Prediction

Sudatta Mohanty
University of California
Berkeley
CA, USA, 94720
sudatta.mohanty@berkeley.edu

Alexey Pozdnukhov
University of California
Berkeley
CA, USA, 94720
alexeip@berkeley.edu

ABSTRACT

Accurate real-time predictions for traffic congestion in a region and knowledge of its causes may allow implementation of effective dynamic control strategies. However, the complex nature of congestion propagation and network-wide spatio-temporal correlations make prediction challenging. To facilitate this process, we define a novel dynamic state variable corresponding to a zone with homogeneous and slowly evolving traffic, called Macroscopic Congestion Level (MCL). We hypothesize that future MCL is a function of the current and past network states for the region-wide network, defined by Origin-Destination (O-D) demand, link counts, link travel times and observed MCL values. We leverage the fact that transportation systems often generate graph-like data either because physical movement is constrained to a road network or due to the coordination of travel choices made by various individuals. We construct a knowledge graph and implement a Graph-CNN + LSTM model to make real-time predictions. The model accuracy is tested against several baselines: (i) 1-NN model, (ii) LSTM-only model and (iii) Graph-CNN + LSTM model with no road network related priors; on simulated data of home-work and work-home trips on a simplified freeway network representing nine counties in the SF Bay Area. Our results indicate improvement in performance which may be attributed to better feature learning by Graph-CNN. Finally, we develop a Neural Attention based framework to produce a spatio-temporal saliency heatmap of input variables. Tests on a toy network with hypothetical demand demonstrate the effectiveness of the proposed framework for identifying the specific cause of congestion.

1. INTRODUCTION

Traffic congestion is defined as a negative externality caused by an imbalance in traffic demand and capacity which adversely impacts users, facilities, and network performance [30]. Congestion patterns are generally periodic which proves useful for prediction algorithms [27]. However, congestion prediction remains challenging since traffic may deviate from regular patterns due to several factors such as inclement weather, accidents among several others [34] [13]. Moreover, in oversaturated networks, the evolution of traffic is highly chaotic with several hidden relationships. For example, route choice is not straight forward in such a scenario because rational drivers try to anticipate congestion propagation and change their behavior accordingly, which may even lead to worse overall conditions, especially during peak congestion hours [1]. Further, it has been shown that even small fluctuations in traffic demand patterns may lead to large fluctuations in the resulting congestion impacts [5]. This motivates development of models which *memorize* important recurring traffic characteristics, yet accommodate the possibility of both structured and unstructured deviations detected from recent data. Moreover, for deploying real-world dynamic control strategies, it is also important to make accurate congestion predictions over a large enough time horizon as well as identify the causes of congestion [32]. A possible solution to address these challenges is outlined in this paper.

1.1 Key Contributions

The key contributions of this paper may be summarized as:

- Effectively representing the congestion state in a zone for the purpose of modeling through a dynamic variable called Macroscopic Congestion Level (MCL).
- Predicting MCL for the near future with the help of signals received from a larger network that includes locations where traffic originates or passes through before arriving at a destination.
- Improving prediction accuracy of the proposed model by storing important priors about spatial correlations as knowledge graphs and implementing a Graph CNN + LSTM model.
- Potentially identifying the causes of congestion by formulating a Neural Attention based framework and demonstrating the capabilities of this framework through simple experiments conducted on hypothetical scenarios.

2. BACKGROUND

There exist two popular approaches for congestion prediction - (i) modeling queue propagation induced by an *active bottleneck* (LWR model) [21] or (ii) modeling the equilibrium conditions induced by desired *activity demands* of individuals (ABM) [2]. In either case, there are challenges towards getting a meaningful representation of network level congestion patterns. For LWR models, the main challenges lie in modeling boundary conditions during merging and diverging and accounting for heterogeneity in behavior of different agents in the network [23]. For ABMs, the key challenge is the requirement of extensive individual level data to

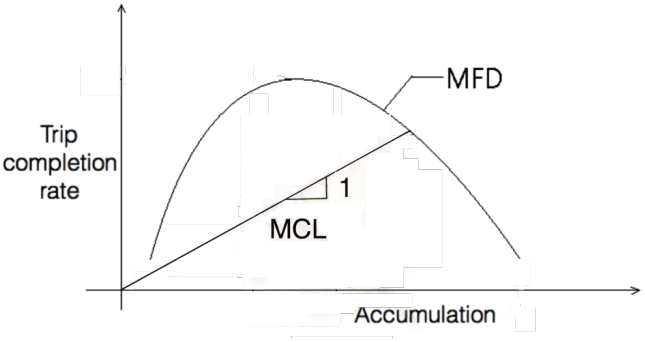

Figure 1: Macroscopic Fundamental Diagram (MFD)/Network Exit Function (NEF): It represents the relationship between trip completion rate and accumulation in a well-defined zone with steady state traffic. Macroscopic Congestion Level (MCL), $\xi(t)$, is defined as the inverse of the slope of the point on the curve at time t.

substantially replicate population behavior. Usually, ABMs aim to replicate traffic on *typical* days and are not sensitive enough to deviations from recurrent conditions [37]. In an attempt to alleviate the challenges posed by microscopic analysis, [7] proposed a *Macroscopic Fundamental Diagram* (MFD) which derives a relationship between aggregate traffic characteristics in a well-defined zone with homogeneous traffic conditions. It was further shown that these relationships exist in real-world networks [12]. The key variables of interest are *accumulation*, defined as the number of travelling vehicles in a region, and *trip completion rate*, defined as the frequency of completion of trips in a region (Figure 1). The trip completion rate increases with an increase in traffic demand as long as the accumulation is low and individuals don't face any delays. However, high accumulation leads to congestion delays which in turn leads to reduction in trip completion rate. An optimal control strategy should aim to maintain near maximum possible trip completion rate for long time periods and avoid periods of very high accumulation, which may lead to *gridlocks* [7].

Congestion in a region may be predicted with the help of inputs such as Origin-Destination (O-D) demands, which encode critical high-level information like peak hour times, and link counts/travel times, which encode critical low-level relationships like those between immediate geographical neighborhoods in a traffic network. The congestion state in neighboring regions may also help detect phenomenon such as *queue spillover* [6]. In order to develop highly accurate models which are both spatially and temporally deep, CNN-LSTM frameworks, such as [28], have been suggested. A common theme in CNN-LSTM frameworks for traffic prediction is the use of images as the input data format. Unfortunately, several trivial pieces of prior information need to be learned before the model can make predictions. This may include information such as traffic movement being constrained on a physical road network or the fact that there is a lagged dependence between traffic demand and resulting congestion based on the travel time along a suitable network path. A possible and relatively simple solution to encode such information is to represent the input data in a structured format with the help of graphs. This ensures that relative distances between nodes are calculated along the existing physical network rather than the conventional Euclidean distances.

The goal now is to extract features from this graphical input since convolution and pooling operations for CNNs are only defined on regular grids. Two possible approaches have been proposed to extend CNNs to graphs. The first approach is defining neural network architectures in the spatial domain to learn from graphical data [9] [10] [19]. These architectures are optimized for specialized tasks and are thus easy to train. However, the operations do not involve convolutions or pooling operations which are easily generalizable. The second approach, which is also followed in this paper, is derived from the *Spectral Graph Theory* [3] [8] [17]. As proposed by [8], we consider a graph $G = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ is the set of nodes ($|\mathcal{V}| = n$), $\mathcal{E}$ is the set of edges and $W \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix encoding the connection weight between any two vertices. The graph Laplacian may be diagonalized using the Fourier basis U as $L = U\Lambda U^T$ where $\Lambda = diag(\lambda_0, ..., \lambda_{n-1})$. Now, any input signal x may be filtered by a graph $g(\theta)$ to produce the following output:

$$y = g_\theta(L)x = g_\theta(U\Lambda U^T)x = U g_\theta(\Lambda)U^T x \tag{1}$$

where, $U^T x$ is the Graph Fourier Transform.

Since operations in the Fourier basis are costly ($\mathcal{O}(n^2)$ for translation), [8] proposes a Chebychev polynomial approximation for the function $g_\theta(\Lambda)$. Pooling operations may be performed based on agglomerative clustering by defining a suitable neighborhood [3]. This allows coarsening of the graph in the same vein as CNNs reduce the input data dimensionality. Such an architecture has recently been used for traffic prediction by [20]. They propose a graphical structure assuming that traffic flow is a diffusion process where transitions may occur due to random walks and see 12-15% improvement over ARIMA baselines. In this research, prior knowledge about route choice is used to define the graphical structure which proves even more useful for modeling real-world dynamics.

In traffic prediction literature, deep neural networks have often been criticized for being non-interpretable "black-boxes", and thus unsuitable for real-world policy deployments [31] [22]. In order to better understand a trained model which may also help us identify the specific cause of congestion at a given space and time, we propose a *Neural Attention* based framework [29] [35]. A simple scheme described by [24] for the purpose of deriving spatio-temporal saliency during video captioning may be applied. The loss in information at time t due to constrained input set corresponding to all features except feature f is a proxy for the relative importance of the given feature:

$$Loss_{f,t} = D_{KL}(p(Y(t)), q(Y(t))) \tag{2}$$

where: $Loss_{f,t}$ is the information loss corresponding to feature f at time t
D_{KL} is the KL-divergence between the two probability distributions
p & q are the probability distributions for producing classification output $Y(t)$ from a trained model using all features and all features except feature f respectively

In case the problem is a regression problem rather than a classification problem, the KL-divergence loss function can be replaced by an L_2 loss based function.

The representation of congestion using aggregate characteristics, the ability to incorporate graphical priors while making predictions as well as the extension of Neural Attention based frameworks to interpret trained models makes network level congestion prediction for dynamic controls an achievable task.

3. MACROSCOPIC CONGESTION REPRESENTATION AND PREDICTION

3.1 Mathematical Formulation Of Macroscopic Congestion Level (MCL)

We first leverage the MFD to define a scoring function, which is well suited for the purpose of modeling congestion in a region of interest. Given a graph $G = (\mathcal{V}, \mathcal{A})$ representing the road network where $\mathcal{A}$ represents directed road links approximated by straight line segments and $\mathcal{V}$ represents end points of road links, the area covered by the road network is partitioned into Z zones/sub-regions in such a way that traffic in each zone is homogeneous with steady state conditions existing at any given time [16]. We assume that each link is completely contained within a zone. Zone z contains subset of links A^z. Defining $n_i(t)$ as the number of vehicles travelling on link i at time t (excluding parked vehicles). Let $A_i(t)$ and $L_i(t)$ represent the cumulative number of vehicles who have arrived and left link i respectively by time t. Therefore, we have $n_i(t) = A_i(t) - L_i(t)$. Now, the aggregate accumulation, $n^z(t)$, in zone z can be derived as a function of $n_i(t)$:

$$n^z(t) = \sum_{i \in A^z} n_i(t) \tag{3}$$

The total trip completion rate, $T^z(t)$, is defined as:

$$T^z(t) = \sum_{i \in A^z} \frac{\partial L_i(t)}{\partial t} \tag{4}$$

From [7], we know that the performance in terms of macroscopic congestion in zone z at time t may be measured as a function of $n^z(t)$ and $T^z(t)$ only. We define Macroscopic Congestion Level (MCL) in zone z, $\xi^z(t)$, as:

$$\xi^z(t) = \begin{cases} 0, \ T^z(t) < \tau^z \& \ n^z(t) < \alpha^z \\ n^z(t)/T^z(t), \ otherwise \end{cases} \tag{5}$$

where, τ^z is a threshold trip completion rate and α^z is a threshold accumulation below which analysis of macroscopic congestion in the region is deemed uninteresting.

The value of $\xi^z(t)$ is unstable when $T^z(t)$ and $n^z(t)$ both have low values. These are situations where traffic demand is very low and hence trip completion rate observed is also low. We are typically not interested in modeling these scenarios very accurately. We may apply further smoothening to the value of $\xi^z(t)$ in order to avoid noise at low MCL values.

Though exact measurements of $\xi^z(t)$ would require arrival and departure counts on all links in z, the assumption of homogeneous loading in each partitioned zone means that estimates can be made by sampling counts from a partition of z's links. This makes it possible to track MCL values even when data from all links in a region is unavailable.

3.2 Network State Definition

Network state, $X(t)$, at time t is defined as a vector of Origin-Destination (O-D) demands, $D(t)$, link counts, $C(t)$, and link travel times, $TT(t)$. For the purpose of modeling macroscopic congestion in zone z, we may restrict the O-D demand input to only those which have the destination zone z, since those demands are hypothesized to have much higher correlation with the congestion generated in zone z. This reduces the number of input O-D demand variables from $\mathcal{O}(|Z|^2)$ to $\mathcal{O}(|Z|)$. Similar heuristics may be employed in order to reduce the dimensionality of vectors $C(t)$ and $TT(t)$ also (for example, filtering out links which are at a network distance greater than some threshold δ). A possible data driven approach for dimensionality reduction after graph transformation of network state input may be through graph sampling [4]. Therefore, we may define $X^z(t)$ as the input network state for the purpose of predicting macroscopic congestion in zone z as[1]:

$$X^z(t) = \begin{bmatrix} D^{\cdot,z}(t) \\ C^z(t) \\ TT^z(t) \\ \xi(t) \end{bmatrix} \tag{6}$$

where:
$D^{\cdot,z}(t)$ is the subset of O-D demands with destination zone z
$C^z(t)$ is a subset of link counts for predicting MCL in zone z
$TT^z(t)$ is a subset of link travel times for predicting MCL in zone z
$\xi(t)$ is the vector of observed MCL values in all zones at time t.

3.3 Graph Transformation

From the road network graph $G = (\mathcal{V}, \mathcal{A})$ and the defined zones Z, we construct a weighted undirected graph $\tilde{G} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, W)$. Two possible procedures are discussed for generation of $\tilde{G}$:

1. Shortest Path Based Weight
   The idea here is to assign weights based on the shortest path between zones along the road network graph.

   - Each $\tilde{v} \in \tilde{\mathcal{V}}$ represents the centroid of zone $z \in Z$
   - The weight between nodes $\tilde{v}_\alpha, \tilde{v}_\beta \in \tilde{\mathcal{V}}$ representing zones z_α, $z_\beta \in Z$ is determined as the shortest directed path length between nodes calculated on the road network graph G using the links $\mathcal{A}$

     $$W_{\alpha,\beta} = D_{SP,G}(v_\alpha, v_\beta) \tag{7}$$

     where, $D_{SP,G}$ is the shortest path calculated on graph G.

[1]Input state variables, especially O-D demand, may not always be accurate or completely observed. Sensitivity analysis when input is noisy/incomplete/correlated is performed in Appendix A

- An input signal x from observed network state may be transformed into an input signal $\tilde{x}$ for $\tilde{G}$ as follows:
  - The O-D demand D^{z_α, z_β} is distributed equally as part of input signals for corresponding nodes v_α and v_β.
  - Each link count C_i on link i is distributed based on prior probabilities of origin/destination zones of all trips observed on link i.
  - Each link travel time TT_i on link i is distributed based on prior probabilities of origin/destination zones of all trips observed on link i.

  Therefore the transformed input signal for node v_α for predicting macroscopic congestion in zone z may be calculated as:

$$\tilde{x}_\alpha^z(t) = \begin{bmatrix} D^{z_\alpha, z}(t)/2 \\ \sum_i p(\alpha|i) C_i(t)/2 \\ \sum_i p(\alpha|i) TT_i(t)/2 \end{bmatrix} \quad (8)$$

  where, $p(\alpha|i)$ is the prior probability of an agent observed on link i having O-D pair as either (z, z_α) or (z_α, z).

2. Trajectory Clustering Based Weight
   The idea here is to determine "distances" between O-D pairs based on the similarities in route choices distribution corresponding to O-Ds.
   - Each $\tilde{v} \in \tilde{\mathcal{V}}$ represents pair of zones $(z_\alpha, z_\beta) \in Z$.
   - The weight is determined as the K-L divergence between the distribution of trips corresponding to two O-D pairs over clusters of trajectories. We follow the following steps:
     (a) Trajectory clustering
     The goal here is to cluster trajectories of various trips executed based on similarities in the routes chosen. We may implement any trajectory clustering algorithm from existing literature such as partition and group approach, a mixture of regression models or *Dynamic Time Warping* (DTW) [18] [11] [26].
     (b) O-D probability distribution over clusters
     For each O-D pair (z_α, z_β), we determine a probability distribution $\Delta^{z_\alpha, z_\beta}$ over derived clusters as the proportion of trips with origin z_α and destination z_β belonging to each cluster.
     (c) Weight as a function of relative entropy
     Suppose nodes v_a, v_b represent the pairs of zones (z_α, z_β) and (z_γ, z_δ) respectively. Then, the weight between nodes $v_a, v_b \in \tilde{\mathcal{V}}$ is calculated as:

$$W_{ab} = D_{KL}(\Delta^{z_\alpha, z_\beta}, \Delta^{z_\gamma, z_\delta}) \quad (9)$$

     where, D_{KL} represents the K-L divergence between two distributions.
   - An input signal x from observed network state may be transformed into an input signal $\tilde{x}$ for $\tilde{G}$ as follows:
     - The O-D demand D^{z_α, z_β} is assigned to node v_a which represents O-D pair (z_α, z_β)
     - Each link count C_i on link i is distributed based on prior probabilities of origin and destination zones of all trips observed on link i.
     - Each link travel time TT_i on link i is distributed based on prior probabilities of origin and destination zones of all trips observed on link i.

     Therefore the transformed input signal for node v_a for predicting macroscopic congestion in zone z may be calculated as:

$$\tilde{x}_a^z(t) = \begin{bmatrix} D^{z_\alpha, z}(t) \\ \sum_i p(\alpha, \beta|i) C_i(t) \\ \sum_i p(\alpha, \beta|i) TT_i(t) \end{bmatrix} \quad (10)$$

     where, $p(\alpha, \beta|i)$ is the prior probability of an agent observed on link i having O-D pair as (z_α, z).

3.4 Model Formulation

The goal of the model is to predict future MCL values in zone z as a function of current and past input signals $\tilde{x}^z$ from observed network state on a particular instance of the transformed graph $\tilde{g}_\theta$. We define the following model for prediction:

$$[\hat{\xi}^z(t+h+p), .., \hat{\xi}^z(t+h)] = f(\tilde{g}_\theta(L)\tilde{x}^z(t), .., \tilde{g}_\theta(L)\tilde{x}^z(t-p)) \quad (11)$$

where:
$\hat{\xi}^z(t)$ is the predicted MCL in zone z at time t
f is a function approximator (eg. Graph CNN-LSTM model)
h is the *minimum dependency lag* between MCL in a target zone z and input state (a function of travel times from the corresponding locations to zone z)
p is the *maximum dependency persistence* of input state on MCL in zone z (a function of the demand pattern and the output capacity of zone z).

3.5 Neural Attention Model

For the purpose of modeling attentions, we consider the sequence of prediction produced by the trained model in equation (11) as the *ground truth* prediction:

$$\hat{\xi}^z = [\hat{\xi}^z(t+h+p), ..., \hat{\xi}^z(t+h)] \quad (12)$$

Now, the attention weights are derived as described by [24]. When the input feature set is constrained, the unobserved feature values are estimated through prior distributions based on previously observed data. Next, the prediction vector is re-evaluated based on this new feature instance. Let this vector be $\tilde{\xi}^z$. The loss due to a constrained input feature set is calculated as the deviation (in L_2 norm) between $\tilde{\xi}^z$ and $\hat{\xi}^z$. The detailed procedure is described as follows.

Let the hidden state value calculated at time t when all input feature information is available be hid_t. Now, when the input set is constrained such that only information about feature f is available at lag l, we recalculate the hidden state value at lag l by replacing the values corresponding to all features $f' \neq f$ by their corresponding prior estimates. As a result, all hidden state values for lags 1 to $(l-1)$ also get affected. We represent these modified hidden state values at time t as $\widetilde{hid}_t$. Now, in case no information for any feature

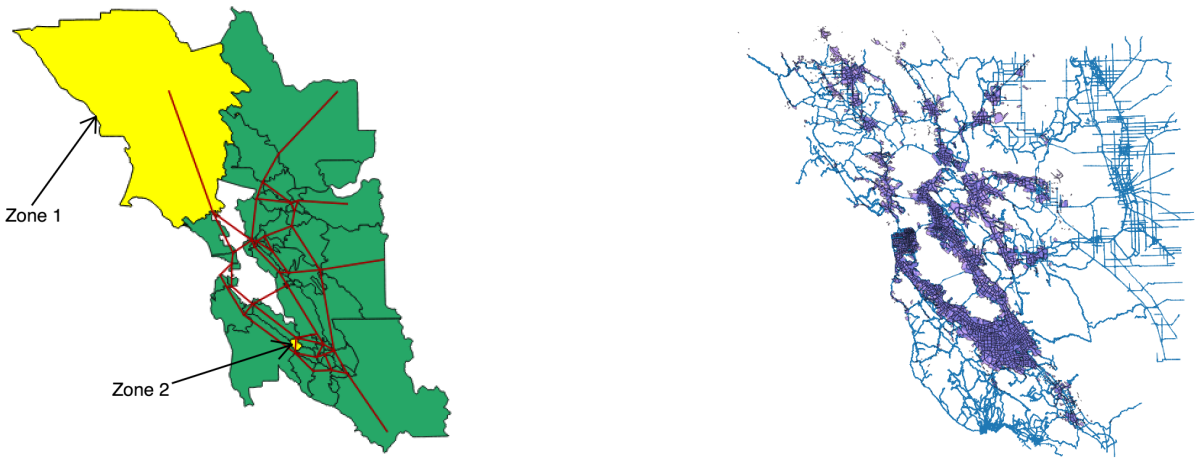

Figure 2: (a) Simplified freeway network and (b) Full scale network - representing 9 counties of SF Bay Area. Each region is partitioned into zones. Target zones for MCL prediction are highlighted in yellow.

is available at lag l, we recalculate the hidden state value by replacing the values corresponding to all features by their corresponding prior estimates. Once again, all hidden state values for lags 1 to $(l-1)$ also get affected. We represent these modified hidden state values at time t as $\widehat{hid}_t$.

The loss value corresponding to feature f at lag l, $Loss_{f,l}$, can be calculated as follows:

$$\hat{\xi}^z_{1,f,l} = \sum_{i=1}^{l-1}\{W_{LSTM\,i} * \widehat{hid}_{t-h-i} + B_{LSTM\,i}\}+ \sum_{i=l}^{p}\{W_{LSTM\,i} * hid_{t-h-i} + B_{LSTM\,i}\}$$

$$\hat{\xi}^z_{2,f,l} = \sum_{i=1}^{l}\{W_{LSTM\,i} * \widetilde{hid}_{t-h-i} + B_{LSTM\,i}\}+ \sum_{i=l+1}^{p}\{W_{LSTM\,i} * hid_{t-h-i} + B_{LSTM\,i}\}$$

$$Loss_{1,f,l} = ||\hat{\xi}^z - \hat{\xi}^z_{1,f,l}||_2$$

$$Loss_{2,f,l} = ||\hat{\xi}^z - \hat{\xi}^z_{2,f,l}||_2$$

$$Loss_{f,l} = Loss_{1,f,l} - Loss_{2,f,l} \quad (13)$$

where, $W_{LSTM\,i}$ and $B_{LSTM\,i}$ are weights and biases of the LSTM model at lag i.

Finally, the spatio-temporal attention is calculated as the proportion of total loss contributed to by feature f at lag l:

$$att_{f,l} = \begin{cases} \frac{Loss_{f,l}}{\sum_{k\in F}\sum_{i=1}^{p} Loss_{k,i}}, & \sum_{k\in F}\sum_{i=1}^{p} Loss_{k,i} \neq 0 \\ \frac{1}{p*N}, & \sum_{k\in F}\sum_{i=1}^{p} Loss_{k,i} = 0 \end{cases} \quad (14)$$

where:
F is the set of all input features
N is the dimensionality of the input feature set.

4. EXPERIMENTAL RESULTS

4.1 Model Prediction Accuracy

The prediction accuracy for the proposed model (see equation (11)) was tested on two networks described in Figure 2. The first network (Figure 2(a)) comprises of 39 nodes and 54 links representing a simplified freeway network for nine counties of San Francisco Bay Area. The entire region is categorized into 54 zones such that a zone's entire street network is represented by a single link. This is done for the purpose of convenience. The second network is a detailed road network representing the nine counties of San Francisco Bay Area, consisting of 352,012 nodes and 564,368 links. Here, the area is partitioned into 1454 zones as per Metropolitan Transportation Commission's (MTC's) Travel Model One[2]. Only trips confined within the partitioned zones were considered for these experiments.

The congestion patterns were generated using an agent-based activity demand model through a well-known open-source traffic simulation software MATSim [15]. The process involves first deriving typical desired *activity chains* (in this case, home → work → work chains) of individuals in the population and then determining equilibrium by maximizing individual utilities to come up with a set of possible executed trajectories. Individuals gain utility by performing an activity at desired times and lose utility due to increased travel times. Spatio-temporal distribution of home and work locations and typical desired home/work activity times were estimated demand using Census Transportation Planning Products (CTPP) data for years 2006-2010 [3]. To ensure variability in demand across days, the start time and duration for both home-work trips and work-home trips were sampled from Gaussian probability distributions. Based on the mean and standard deviation of the probability distribution from which these parameters are sampled, four scenarios were generated as described in Table 1. Simulations

[2] https://github.com/BayAreaMetro/modeling-website/wiki/TravelModel, https://mtc.maps.arcgis.com/home/item.html?id=b85ba4d43f9843128d3542260d9a2f1f

[3] http://ctpp.transportation.org/Pages/5-Year-Data.aspx

Table 1: Table describing parameter set for Gaussian distributions which determine the start time and duration of h-w and w-h trips generated for testing proposed Graph CNN + LSTM model

Scenario	μ_{st} h-w	μ_{st} w-h	σ_{st}	μ_d	σ_d
1	8.5	17.5	1	1	0
2	8.5	17.5	2	1	0
3	8.5	17.5	1	1	1
4	8.5	17.5	2	1	1

μ_{st}, μ_d: Mean start time and duration
σ_{st}, σ_d: Stdev of start time and duration
h-w, w-h: Home-work and work-home trip

were performed for 1000 days and based on the simulation results, the actual executed trajectories for each individual were derived. These trajectories were then utilized to derive accumulation and trip completion rates in the zones of interest as well as input feature values (i.e. O-D demand, link counts and link travel times). Therefore, the MCL values in the target zones can now be predicted with the help of derived input data.

As described in section 3.3, two implementations of the Graph CNN + LSTM model were tested. For the shortest path based weight model, the centroid of each zone in Figure 2 represented the node $\tilde{v}$. Shortest paths were calculated between each pair of centroids using Djikstra's algorithm. In case the centroid did not lie on an existing link, straight line paths (in Euclidean space) were assumed from the centroid to its nearest link. As a result, we derived a complete graph, $\tilde{G}$. For the freeway network (Figure 2(a)), shortest paths were calculated between each pair of centroids. For the detailed network (Figure 2(b)), only the k (= 2) nearest neighbors were connected by shortest paths in order to conserve memory. For the trajectory clustering based weight model, a sample of 565,000 trips were analyzed. Pairwise distances were calculated between the trips using DTW distance. Then, clustering was performed using standard *K-Means* clustering with $k = 100$ clusters. The number of clusters was motivated from a finding by [33] that most real-world trips get clustered around a few routes. The discrete O-D probability distribution of trips over these clusters was determined with probability mass at cluster κ equal to the ratio of trips contained in cluster κ and the total number of trips corresponding to the particular O-D. Finally, the O-D adjacency matrix was calculated as the KL-divergence between any two O-D probability distributions. Once again, to limit memory consumption, only the "important" O-Ds were included for graph creation. Thresholds of 50 trips per O-D for the freeway network and 10 trips per O-D for the detailed network were chosen. As a result, for the freeway network, we filter out 2265 out of the possible 2916 O-D pairs leaving 651 O-D pairs. For the detailed network, we filter out 2,111,943 out of the possible 2,114,116 O-D pairs leaving 2173 O-D pairs. We further consider the hypothesis that, for real-world traffic data, a sparse representation of the O-D adjacency matrix should encode most of the useful information since most trips get clustered around a few routes. Consequently, we derive a k-NN graph with $k = 5$ for the freeway network and $k = 2$ for the detailed network.

The proposed Graph CNN + LSTM model was implemented to predict congestion in the highlighted zones (Zone 1, Zone 2 in Figure 2(a) and Zone 3 in Figure Figure 2(b)). The performance of the proposed framework is tested against four baseline models:

i *1-NN model*:
The MCL in zone z at time t is predicted as the observed MCL in the same zone at time $(t - \Delta)$ where, $\Delta = 1440$ *mins* (i.e. observed MCL value at the same time on the previous day).

$$\xi^z(t + \Delta) = \xi^z(t) \tag{15}$$

ii *LSTM-only model*:
The MCL in zone z at time t is predicted through an LSTM model directly applied on the input network state signals at times $((t-h), (t-h-1), ..., (t-h-p))$. The values of h and p were determined by cross-validation. It is defined by the following equation:

$$[\xi^z(t + h + p), ..., \xi^z(t + h)] = f(x^z(t), ..., x^z(t - p)) \tag{16}$$

iii *k-NN Euclidean Weight Graph + LSTM model*:
We derive the graph $\tilde{G} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, W)$ described in section 3.3 by assuming no knowledge of the road network. We hypothesize that features that are in the neighborhood of each other have similar observed values. Therefore, the nodes $\tilde{v}$ represent the feature set F and the weights W are determined as the mean Euclidean distance between the observed values along two feature dimensions.

$$W_{i,j} = \sum_t (X_i(t) - X_j(t))^2$$

where, X_i and X_j represent the observed network state value for dimensions i and j respectively.
Further, to make the graph sparse, we construct edges only along the k nearest neighbors to each node $\tilde{v}$ as per the adjacency matrix ($k = 5$). Here, it is essential to first normalize the observed values along each dimension.

iv *Holt-Winters model* [14]:
A purely statistical exponential smoothening approach for time series prediction using past values with daily seasonality incorporated into the model.

The performance of each model for prediction of macroscopic congestion in zone 1, zone 2 (Figure 2(a)) and zone 3 (Figure 2(b)) for the scenarios mentioned in Table 1 are displayed in Table 2 [4].

We observe that the prediction accuracy for the Graph CNN + LSTM models is better than the LSTM-only model which in turn provides better prediction accuracy than the naive models (i.e. 1-NN and Holt-Winters). The is possibly because naive models don't learn any features from the network state input which might be useful for making predictions. Among the naive models, 1-NN performs better when the amount of variation across days is low and Holt-Winters performs better when the variation is high. The difference in performance between deep learning models and naive models

[4] An open source implementation of the algorithm on the sample scenarios is made available at `https://github.com/sudatta0993/Dynamic-Congestion-Prediction`.

Table 2: Table describing the relative performance of 1-NN, LSTM-only, Graph CNN + LSTM (k-NN Euclidean Weights), Graph CNN + LSTM (Shortest Path Weights) and Graph CNN + LSTM (k-NN Trajectory Clustering Weights) for the prediction of $\xi^z(t)$ in target zones highlighted in Figure 2(b) and demand scenarios described in Table 1

Sc.	Target Zone	1-NN	Holt Winters	LSTM	GCNN+LSTM (k-NN Eu)	GCNN+LSTM (SP)	GCNN+LSTM (k-NN TC)
1	1	0.503	1.104	0.356	0.222	0.194	0.187
1	2	1.300	3.014	0.801	0.637	0.601	0.596
1	3	0.932	0.988	0.752	0.671	0.551	0.530
2	1	1.077	1.121	0.875	0.715	0.462	0.358
2	2	1.831	3.011	1.107	0.998	0.612	0.558
2	3	1.202	1.070	1.050	0.965	0.738	0.537
3	1	1.010	1.510	0.871	0.660	0.205	0.102
3	2	2.256	2.015	1.228	1.070	0.789	0.435
3	3	1.866	1.803	1.657	1.474	1.401	1.386
4	1	0.924	1.082	0.609	0.605	0.377	0.194
4	2	3.155	2.700	2.124	2.024	1.694	0.858
4	3	1.144	0.554	0.467	0.433	0.343	0.313

Sc.: Scenario index
GCNN+LSTM (k-NN Eu): Graph CNN + LSTM model with k-NN graph and Euclidean distance based weights
GCNN+LSTM (SP): Graph CNN + LSTM model with complete graph and shortest path based weights
GCNN+LSTM (k-NN TC): Graph CNN + LSTM model with k-NN graph and trajectory clustering based weights

generally grows with higher uncertainty in the demand start times and travel duration (Except in the case of Scenario 4 for predicting congestion in Zone 3, where the Holt-Winters model performs well). We may attribute the gain in performance for the LSTM model over the naive models to better learning of the temporal difference as a result of various input signals observed at any time. We may attribute the gain in performance for Graph-CNN+LSTM models over the LSTM-only model to better feature learning by exploiting the graphical nature of the input data.

We then compare the relative performance of different graph adjacency structures for the Graph-CNN + LSTM models. We observe that the trajectory clustering based weights slightly outperform the shortest path based weights which in turn outperform the Euclidean distance based weights. We expect better performance when the information about the structure of the underlying road network is encoded into the generated graph. Therefore the higher prediction accuracy for the shortest path weights and trajectory clustering weights over Euclidean weights is intuitive. The performance gain for the trajectory clustering weights over the shortest path weights may simply be attributed to more road network information (i.e. executed routes of agents) being encoded into the generated graph. It must however be noted that the convergence time for the trajectory clustering based weighted graph was found to be around 2-3 times higher on average than the shortest path based weighted graph. This is because the number of nodes, $|\tilde{\mathcal{V}}|$, is $\mathcal{O}(|Z|^2)$ for the trajectory clustering based weighted graph while the corresponding value is $\mathcal{O}(|Z|)$ for the shortest path based weighted graph, although the number of links, $|\tilde{\mathcal{E}}|$, was ensured to be roughly the same.

4.2 Neural Attention Model Performance

Toy Network

We design a simple experiment for testing the Neural At-

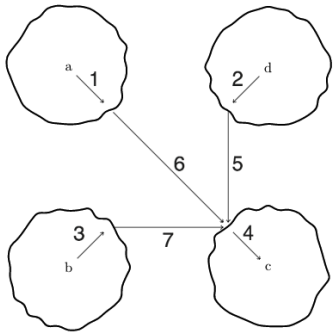

Figure 3: A test network to analyze the Neural Attention Model framework to derive a spatio-temporal saliency heatmap of input variables

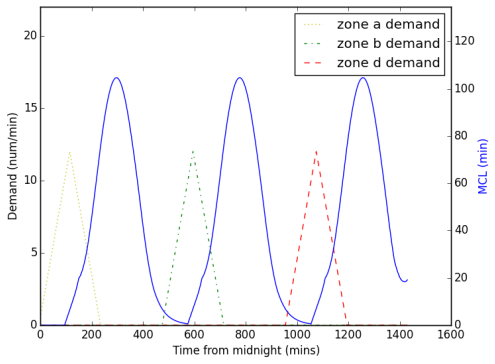

Figure 4: Plot showing demand from the three zones $D^{a,c}(t)$ (yellow), $D^{b,c}(t)$ (green) and $D^{d,c}(t)$ (red) over time plotted on the primary y-axis. On the secondary y-axis, the macroscopic congestion level, $\xi^c(t)$, (blue) is plotted.

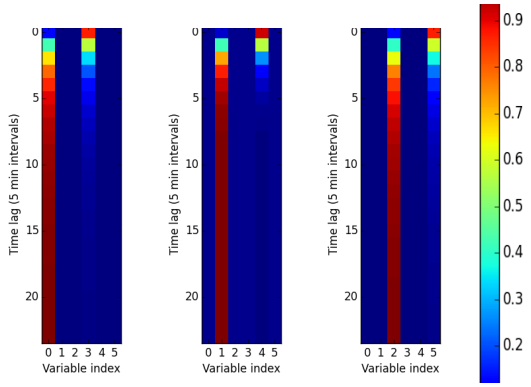

Figure 5: Spatio-temporal attention heatmap for prediction during - (a) 6-8 AM, (b) 2-4 PM and (c) 8-10 PM; Variable definitions: Index 0 = Zone a demand, Index 1 = Zone b demand, Index 2 = Zone d demand, Index 3 = First difference of zone a demand, Index 4 = First difference of zone b demand, Index 5 = First difference of zone d demand

tention based frameworks (Section 3.5). We define a toy network (Figure 3) which contains four zones (a, b, c and d). All traffic moves from zones a, b and d to zone c. We assume existence of triangular Microscopic Fundamental diagram [21] and queue propagation using the LWR model [25]. The demand patterns from each input zone and the resulting congestion pattern from LWR model are displayed in Figure 4. Due to symmetry, both shortest path graph and trajectory clustering graph have the form $W = C*(\mathbf{1}\mathbf{1}^T - I)$, where C is a constant. Therefore, there is no additional information obtained from the structure of the graph. As a result, Graph CNN + LSTM and LSTM-only models produce similar results.

We make predictions at three time intervals - 6-8 AM, 2-4 PM and 8-10 PM. Intuitively, from Figure 4, we notice that zone a demand is critical for making predictions between 6-8 AM, zone b demand is critical for making predictions between 2-4 PM and zone d demand is critical for making predictions between 8-10 PM. Therefore, we would expect the Neural Attention framework to identify zone a demand, zone b demand and zone d demand as most critical for predicting congestion score at 6-8 AM, 2-4 PM and 8-10 PM respectively.

The Neural Attention output is displayed in Figure 5. For predictions made at 6-8 AM, the only input variables which impact the congestion values are variable indices 0 and 3 (i.e. zone a demand and its first difference values respectively). This observation follows our intuition from Figure 4. For variable index 0, the relative importance for higher lag values is higher whereas for variable index 3, the relative importance for higher lag values is lower. This occurs since the congestion curve is concave in nature (Figure 4). The analysis for predictions at 6-8 AM can be extended to predictions at 2-4 PM and 8-10 PM with zone b and zone d acting as the dominant input zones respectively. This illustrates that the Neural Attention framework is capable of extracting spatio-temporal saliency of inputs.

Table 3: Top four O-D demand attentions and the corresponding percentage change in MCL value if the O-D is absent, while predicting MCL in Zone 2 between 10AM-12PM.

att_{O-D} Relative Weight	MCL % change if O-D absent
58.7 %	37.3 %
9.1 %	13.4 %
6.1 %	5.8 %
4.2 %	2.5 %

Simplified Freeway Network

Next, we would like to test the Neural Attention based framework (see equation (14)) for predicting congestion in Zone 2 in the freeway network described in Figure 2(a) and demand scenario 1 described in Table 1. We are restricting our analysis to only evaluate relative importance of all O-D demands between 10AM-12PM with the destination zone as Zone 2. While the ground truth contribution of each O-D demand towards MCL is hard to evaluate, a proxy for the same is the percentage change in MCL in case that particular O-D demand is absent. We expect O-D demands from zones with high relative importance to impact MCL the most.

Table 3 displays the top four contributing O-D demands (out of the possible 54) towards MCL prediction as per the Neural Attention framework. The percentage change in MCL in case of absence is also highest for these four O-Ds. We can note that these O-D demands contribute 78.1% of the total Neural Attention weights. The predicted order of importance for these O-D demands is in accordance with the order of percentage change in MCL values if the O-D demand is absent. The zone which contributes the maximum weight is Zone 2 itself. This is due to several short trips within the zone during this time. One possible reason is the presence of several tech companies in the region and the preference of employees to live close to the work place[5]. While the initial results for the attention framework look promising, the authors acknowledge that their application towards a dynamic control strategy would provide an even more convincing argument for using them in practice.

5. CONCLUSION

Through this paper, we address the key challenge of modeling congestion in a region as a dynamic state variable and predicting future states in order to aid dynamic control strategies. An MFD based representation of congestion state is shown to be the most convenient for this purpose. Recent advances in the field of deep learning further allow us to incorporate priors represented in the form of graphs for making prediction which is shown to lead to significant accuracy improvements. Finally, a framework for identifying the relative spatio-temporal impact of various inputs is discussed which is shown to be useful for identifying the cause of congestion at any given location and time. It remains to be shown that these predictions and relative impacts can be successfully incorporated into a dynamic control strategy for reducing congestion impacts.

[5]see `http://www.vitalsigns.mtc.ca.gov/`

6. FUTURE WORK

- The techniques for predicting congestion in a single target zone can be easily generalized to predicting congestion across multiple zones and thus extracting network-wide congestion predictions.
- The MCL states over time and corresponding controls/actions to manage congestion can be represented with the help of *Markov Decision Process* (MDP) which would allow model-based *Reinforcement Learning* (RL) techniques to derive optimal dynamic control policies.
- Graph sampling techniques such as those described in [4] may allow memory efficient utilization of graphical priors.

7. ACKNOWLEDGEMENTS

The authors would like to thank Prof. Michael Cassidy and Prof. Mark Hansen for their valuable help and guidance towards this research.

8. REFERENCES

[1] R. Arnott and K. Small. The economics of traffic congestion. *American scientist*, pages 446–455, 1994.

[2] C. R. Bhat and F. S. Koppelman. Activity-based modeling of travel demand. In *Handbook of transportation Science*, pages 35–61. Springer, 1999.

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

[4] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero. Learning sparse graphs under smoothness prior. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 6508–6512. IEEE, 2017.

[5] C. F. Daganzo. The nature of freeway gridlock and how to prevent it. In *Internaional symposium on transportation and traffic theory*, pages 629–646, 1996.

[6] C. F. Daganzo. Queue spillovers in transportation networks with a route choice. *Transportation Science*, 32(1):3–11, 1998.

[7] C. F. Daganzo. Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, 41(1):49–62, 2007.

[8] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852, 2016.

[9] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[10] E. N. Feinberg, D. Sur, B. E. Husic, D. Mai, Y. Li, J. Yang, B. Ramsundar, and V. S. Pande. Spatial graph convolutions for drug discovery. *arXiv preprint arXiv:1803.04465*, 2018.

[11] S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 63–72. ACM, 1999.

[12] N. Geroliminis and C. F. Daganzo. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9):759–770, 2008.

[13] T. F. Golob and W. W. Recker. Relationships among urban freeway accidents, traffic flow, weather, and lighting conditions. *Journal of transportation engineering*, 129(4):342–353, 2003.

[14] C. Holt. Forecasting trends and seasonals by exponentially weighted averages. carnegie institute of technology. Technical report, Pittsburgh ONR memorandum, 1957.

[15] J. Illenberger, G. Flötteröd, and K. Nagel. Enhancing matsim with capabilities of within-day re-planning. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 94–99. IEEE, 2007.

[16] Y. Ji and N. Geroliminis. On the spatial partitioning of urban transportation networks. *Transportation Research Part B: Methodological*, 46(10):1639–1656, 2012.

[17] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[18] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[19] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

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

[21] M. J. Lighthill and G. B. Whitham. On kinematic waves. ii. a theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 229, pages 317–345. The Royal Society, 1955.

[22] H. Z. Moayedi and M. Masnadi-Shirazi. Arima model for network traffic prediction and anomaly detection. In *Information Technology, 2008. ITSim 2008. International Symposium on*, volume 4, pages 1–6. IEEE, 2008.

[23] S. Peeta and A. K. Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1(3-4):233–265, 2001.

[24] V. Ramanishka, A. Das, J. Zhang, and K. Saenko. Top-down visual saliency guided by captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] P. I. Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956.

[26] S. Sankararaman, P. K. Agarwal, T. Mølhave, J. Pan, and A. P. Boedihardjo. Model-driven matching and segmentation of trajectories. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 234–243. ACM, 2013.

[27] A. Skabardonis, P. Varaiya, and K. Petty. Measuring recurrent and nonrecurrent traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board*, (1856):118–124, 2003.

[28] X. Song, H. Kanasugi, and R. Shibasaki. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *IJCAI*, pages 2618–2624, 2016.

[29] Y. Tang, N. Srivastava, and R. R. Salakhutdinov. Learning generative models with visual attention. In *Advances in Neural Information Processing Systems*, pages 1808–1816, 2014.

[30] M. Taylor. Exploring the nature of urban traffic congestion: concepts, parameters, theories and models. In *PROCEEDINGS, 16TH ARRB CONFERENCE, 9-13 NOVEMBER 1992, PERTH, WESTERN AUSTRALIA; VOLUME 16, PART 5*, 1992.

[31] J. Van Lint, S. Hoogendoorn, and H. J. van Zuylen. Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5):347–369, 2005.

[32] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias. Short-term traffic forecasting: Where we are and where were going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014.

[33] C. Wu, J. Thai, S. Yadlowsky, A. Pozdnoukhov, and A. Bayen. Cellpath: fusion of cellular and traffic sensor data for route flow estimation via convex optimization. *Transportation Research Procedia*, 7:212–232, 2015.

[34] F. Xu, Z. He, Z. Sha, L. Zhuang, and W. Sun. Assessing the impact of rainfall on traffic operation of urban road network. *Procedia-Social and Behavioral Sciences*, 96:82–89, 2013.

[35] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[36] X. Yang, Y. Lu, and W. Hao. Origin-destination estimation using probe vehicle trajectory and link counts. *Journal of Advanced Transportation*, 2017, 2017.

[37] M. Yin, S. E. Sheehan, S. Feygin, J.-F. Paiement, and A. Pozdnoukhov. A generative model of urban activities from cellular data. In *IEEE Trans on Intelligent Transportation Systems (to appear)*, 2017.

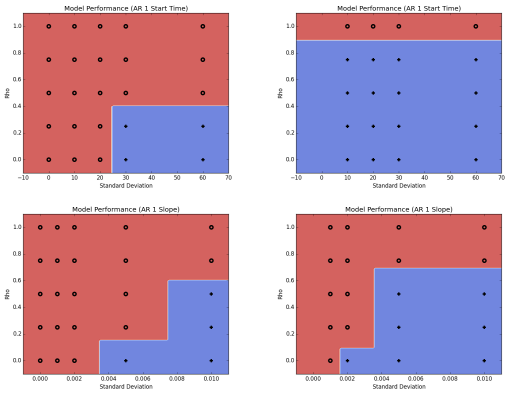

Figure 6: Efficiency bounds for prediction accuracy deep learning model vs 1-NN model as a function of (ρ_{st}, σ_{st}) (Top) and (ρ_{sl}, σ_{sl}) (Top) for - (a) p = 24 hrs (Left) and (b) p = 2 hrs (Right)

[38] X. Zhou and H. S. Mahmassani. A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework. *Transportation Research Part B: Methodological*, 41(8):823–840, 2007.

APPENDIX

A. SENSITIVITY ANALYSIS

Among components of vector $X(t)$ (see equation (6)), O-D demands, $D(t)$, are not directly observable in real-time. Several data-driven techniques have been proposed to estimate O-D demand in recent times [33] [36]. However, there is a possibility of errors or missing data at certain times. Another possibility is that of high correlation in O-D demands across multiple days which favors naive models such as the 1-NN model. We analyze the sensitivity of prediction accuracy in these situations.

A.1 Effect Of Correlation In O-D Demand

Figure 4 describes the variation of O-D demand within a day for the toy scenario. The triangular demand pattern is characterized by two parameters - (i) start time of demand and (ii) slope of demand. From past models for O-D demand estimation assuming auto-regressive structures such as [38], we define a simple AR-1 process governing parameter values:

$$st_{t+1,z} = \rho_{st,z} * st_{t,z} + (1 - \rho_{st,z}) * \mu_{st,z} + \epsilon_{st,z},\ \epsilon_{st,z} \sim \mathcal{N}(0, \sigma_{st,z})$$
$$sl_{t+1,z} = \rho_{sl,z} * sl_{t,z} + (1 - \rho_{sl,z}) * \mu_{sl,z} + \epsilon_{sl,z},\ \epsilon_{sl,z} \sim \mathcal{N}(0, \sigma_{sl,z})$$

where, ρ_{st} and ρ_{sl} are auto-correlation parameters for start time and slope of demand respectively.

We compare the prediction accuracy of the 1-NN model and the LSTM model for various (ρ_{st}, σ_{st}) pairs as well as (ρ_{sl}, σ_{sl}) pairs. Higher value of ρ is hypothesized to favor the baseline 1-NN model while higher value of σ is hypothesized to favor the deep learning model.

In each plot in Figure 6, the parameter values governing the demand generating process are plotted on the x and y axes and an indicator of the better performing model is plotted

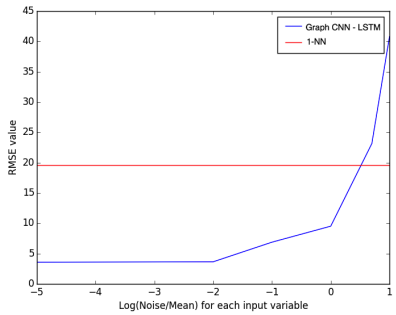

Figure 7: Impact of noise in O-D demand data on the prediction accuracy of deep learning model

using a symbol ("+" signifies that deep learning model outperformed 1-NN model and "o" signifies the opposite). The area of the "blue" region signifies the relative superiority of the deep learning model. We see that, when we set the value of hyperparameter p as 24 hrs, the 1-NN model starts to outperform the deep learning model for a large set of scenarios. This is because the deep learning model tries to predict too far into the future and therefore has larger errors (see equation (11)). However, by reducing the value of parameter p to 2 hrs, we make the deep learning model more robust to high correlations in OD demands. The optimal p can be chosen by trading off robustness with the length of the prediction horizon.

A.2 Effect Of Missing O-D demands

We re-evaluate the prediction accuracy for the toy network and demand scenario (see Figures 3 and 4) with certain O-D demands missing. The results are summarized in Table 4

Table 4: Root Mean Squared For deep learning model for various scenarios with omitted variables and different prediction times

Variables Omitted \ Prediction Time	6-8 AM	2-4 PM	8-10 PM
None	6.274	3.359	5.281
Zone a demand	**45.283**	3.360	5.570
Zone b demand	6.723	**42.457**	5.345
Zone d demand	6.186	3.381	**52.843**

As per intuition from Figures 4 and 5, zone a demand is critical for making predictions between 6-8 AM, zone b demand is critical for making predictions between 2-4 PM and zone d demand is critical for making predictions between 8-10 PM. In Table 4, we notice that the omission of critical variables is catastrophic for prediction accuracy. In real-world settings, we may first use the Neural Attention based framework to determine any critical variables for prediction at certain times of the day. In case data corresponding to such a variable is missing, we may revert to other heuristics for prediction.

A.3 Effect of Noisy O-D demand

We assume that O-D data in the toy scenario (Figures 3 and 4) is corrupted by noise. For simplicity, we assume that $(Noise/Signal)$ ratio is constant across all origin demands. The results obtained are summarized in Figure 7. On the x-axis, $(Noise/Signal)$ is plotted on a log scale and on the y-axis the RMSE value for deep learning model (blue) and 1-NN model (red) are plotted. The break-even point between the two models occurs when $(Noise/Signal)$ is approximately equal to 4. This demonstrates the noise tolerance of the deep learning model.