Predicting Traffic Congestion Evolution: A Deep Meta Learning Approach

Yidan Sun , **Guiyuan Jiang*** , **Siew-Kei Lam** and **Peilan He**
Nanyang Technological University, Singapore
ysun014@e.ntu.edu.sg, {gyjiang, assklam}@ntu.edu.sg, phe002@e.ntu.edu.sg.

Abstract

Many efforts are devoted to predicting congestion evolution using propagation patterns that are mined from historical traffic data. However, the prediction quality is limited to the intrinsic properties that are present in the mined patterns. In addition, these mined patterns frequently fail to sufficiently capture many realistic characteristics of true congestion evolution. In this paper, we propose a representation learning framework to characterize and predict congestion evolution between any pair of road segments. Specifically, we build dynamic attributed networks (DAN) to incorporate both dynamic and static impact factors while preserving dynamic topological structures. We propose a Deep Meta Learning Model (DMLM) for learning representations of road segments which support accurate prediction of congestion evolution. DMLM relies on matrix factorization techniques and meta-LSTM modules to exploit temporal correlations at multiple scales, and employ meta-Attention modules to merge heterogeneous features while learning the time-varying impacts of both dynamic and static features. Compared to all state-of-the-art methods, our framework achieves significantly better prediction performance on two congestion evolution behaviors (propagation and decay) when evaluated using real-world dataset.

1 Introduction

Accurate prediction of congestion evolution (such as how the congestion propagates or diminishes across road segments) benefits many traffic management applications, such as traffic prediction [Nguyen *et al.*, 2016], route planning, and bottleneck identification. Existing works on congestion evolution prediction typically rely on data mining techniques [Nguyen *et al.*, 2016]. In these works, pattern mining is applied on historical traffic data to obtain frequent patterns from causality tree/graph [An *et al.*, 2016; An *et al.*, 2018]. These patterns (subtrees, subgraphs) are then utilized for congestion evolution prediction [Nguyen *et al.*, 2016; Xiong *et al.*, 2018]. The quality of such predictions is limited to the inherent properties found in the available patterns. These evolution patterns usually cannot sufficiently capture many realistic characteristics of true congestion evolution such as the significant variation in spatiotemporal range and granularity, and the asymmetric transitivity of congestion propagations.

Factors that will impact congestion evolutions include road characteristics (e.g., road type), Points of Interests (POIs, e.g, school, shopping mall) [Zhang *et al.*, 2017], and weather conditions. However, existing works often do not consider the extent of influence that these factors have on congestion evolution, such as the time-varying impact of static factors (e.g., POIs) [Liao *et al.*, 2018]. For example, a stadium holding large social event (e.g. football game) is likely to cause unexpected congestions; schools tend to affect the traffic only during morning peak, lunch time, and evening peak hours. In this paper, we develop methods to incorporate the abovementioned factors with time-varying impact on congestions to predict the congestion evolution between any road pairs.

The contributions of our work are summarized as follows.

- We propose a representation learning framework to characterize and predict congestion propagation/decay between any pair of road segments. This departs from existing methods that mostly rely on causality trees/graphs and mining propagation patterns.
- We are the first to characterize traffic congestion evolution (propagation/decay) using dynamic attributed networks (DANs). Various factors with time-varying impacts are incorporated into DANs as node attributes. These include static attributes (e.g., POIs, road characteristics) and dynamic attributes (e.g., traffic speed).
- We propose a Deep Meta Learning Model (DMLM) for learning representations of road segments that support accurate prediction of congestion evolution. Meta learning is used to learn the evolving strengths of temporal correlations at multiple scales based on DAN attributes. DMLM also employs meta-Attention modules to merge heterogeneous features while learning the time-varying impacts of both dynamic and static features.
- The performance of our framework on two congestion evolution behaviors, congestion propagation and congestion decay, is evaluated using real-world dataset. Re-

*corresponding author

sults show that our framework significantly outperforms all state-of-the-art methods.

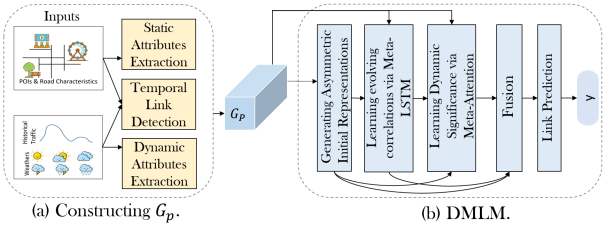
2 Related Work

Traffic Congestion Evolution. Many efforts have been undertaken to model congestion evolutions at three levels of granularity, i.e., macroscopic, microscopic, and mesoscopic. Macroscopic level formulates congestion as a cluster of likewise congested roads [Anwar *et al.*, 2016; An *et al.*, 2016], while microscopic level studies congestion evolution among subsections within an individual road [Sun *et al.*, 2018; Khajeh-Hosseini and Talebpour, 2019; Cui *et al.*, 2019]. Our work focuses on mesoscopic level, which associates congestion with individual road segment and congestion evolution across adjacent road segments. [Nguyen *et al.*, 2016; Di *et al.*, 2019; Yue *et al.*, 2018] formulate congestion propagations as directed edges from downstream to upstream roads. Causality trees and graphs are constructed based on the edges and data mining techniques are used to obtain frequent subtrees/subgraphs as evolution patterns (e.g., decay and propagation). Some works also considered congestion evolution prediction based on the obtained evolution patterns. Most works at mesoscopic level rely on tree or graph structure to mine patterns and make predictions. However, such representation limits the design space of predictors due to the dependency on structural patterns.

Link Prediction on Network. The core idea of congestion propagation prediction is similar to link prediction on complex networks (e.g., social networks), which predicts new relationships (links). Similarity based methods, which assumes that two nodes are similar if they in close proximity or are connected to similar nodes, are most widely used for link prediction [Martínez *et al.*, 2016b]. Local similarity is learned based on local neighborhood information [Martínez *et al.*, 2016a], while global similarity relies on the topological information of the entire network [Lichtenwalter *et al.*, 2010]. In general, similarity-based methods only focus on static structural information and neglect the evolving changes in the networks. In practice, nodes are often affiliated with rich attributes that co-evolve with network structures. As such, there has been a lot of interest to incorporate structure information and node attributes for link prediction [Li *et al.*, 2018; Zhang *et al.*, 2018]. However, existing works typically ignore the temporal characteristics in dynamic networks, and fail to capture long and short term node interactions. Link prediction on DANs is still currently an emerging area of research. In this paper, we study evolution prediction that takes into account dynamic traffic congestion networks (evolving topological structures), and their temporal characteristics (time-varying impacts of various factors on congestion evolution).

3 Constructing Congestion Evolution Networks

This section presents our method to construct attributed networks for modeling congestion evolution (propagation and decay), as shown in Figure 1 (a). The attributed networks capture dynamic congestion evolution using various types of temporal links, while incorporating both static and dynamic node attributes.



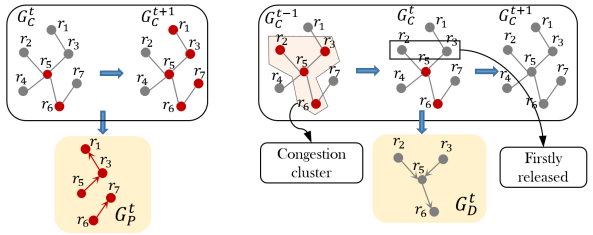
(a) Constructing G_p. (b) DMLM.

Figure 1: Main framework: (a) Modeling congestion evolution using attributed propagation/decay networks. (b) Deep Meta Learning Model for congestion evolution prediction.

Traffic Network. The traffic network is represented as a DAN $G_C = \{G_C^t\}$, where $G_C^t = \{V, E, \mathbf{C}^t\}$, t is time slice ($1 \leq t \leq T$, there are T time slices in total), $V = \{r_1, ..., r_N\}$ is a set of N vertice (each vertex is a road segment), $E = \{(i,j)\}$ is a set of undirected edges (an edge $(i,j) \in E$ indicates that r_i and r_j are spatially connected). V and E capture static topology information that remains unchanged over time. Matrix $\mathbf{C}^t \in \{0,1\}^N$ represents traffic congestion of all N road segments at time t. c_i^t is an element in $\mathbf{C}$ where $c_i^t = 1$ if road r_i's traffic state is congested at time t, otherwise, $c_i^t = 0$. To capture congestion propagation and decay behaviors on the traffic network, two DANs, G_P and G_D, are constructed based on G_C, as shown in Figure 2.

Attributed Propagation Network. $G_P^t = (V_P^t, E_P^t, AD_P^t, AS_P^t)$, where set $V_p^t \subseteq V$ is a subset of V which includes road segments involved in propagations; $E_P^t = \{(i,j)\}$ is a set of directed links, $(i,j) \in E_P^t$ represents a congestion propagation from r_i to r_j at time t; AD_P^t is a matrix of dynamic attributes associated with set V_P^t and each vertex has F_d dynamic attributes (e.g., traffic speed, time indicator, weather); AS_P^t is a matrix of static attributes associated with set V_P^t, and each vertex has F_s static attributes (e.g., POIs, road characteristics). The DAN G_P is constructed in the following way, as shown in Figure. 2 (a). The congestion states at time t and $t+1$ are illustrated in the upper box, based on which a set of congestion propagation paths are detected. Specifically, road r_5 is congested at time t, then road r_3 and r_1 become congested from time t to time $t+1$. This indicates that the congestion on r_5 propagates to r_3 and r_1 (similarly, r_6 propagates its congestion to r_7). Formally, a path $pa = \langle r_{i_1}, .., r_{i_k}, ..., r_{i_K} \rangle$ is a propagation path if $(i_j, i_{j+1}) \in E$ $(1 \leq j \leq K-1)$; $c_{i_1}^t = 1$, $c_{i_j}^t = 0$ and $c_{i_j}^{t+1} = 1 (2 \leq j \leq K)$. All nodes in the obtained propagation paths are included in set V_P^t, and any link appearing in the propagation paths is included in set E_P^t.

The static attributes AS_P^t contain two parts: road characteristics and surrounding POIs. Road characteristics contain road length, number of lanes, number of bus stops and traffic signals, and road type, where one-hot encoding is applied to distinguish between 19 road types (e.g., highway, primary road, etc.). The POIs are grouped into 14 categories (Section 5.1). The POI distribution around a road segment r_i (located in a circle area centered at the middle point of r_i) is formulated as a vector of the number of POIs in each cat-



(a) Example for constructing G_P^t (b) Example for constructing G_D^t

Figure 2: Examples of constructing G_P^t, G_D^t from G_C. Red and gray dots are congested and free-flow road segments, respectively.

egory. Thus, each r_i has a vector of dimension 14. The dynamic attributes AS_P^t of r_i at time t contain: historical speeds of previous W time slices; historical daily and weekly average speed at time t (long term periodicity); time indicator such as time-of-day and day-of-week; holiday indicator ('1' indicates a holiday while '0' otherwise); and weather conditions in hourly granularity (e.g., raining, sunny, rainstorm, etc.) represented using one-hot coding.

Attributed Decay Network. Similar to G_P^t, the DAN $G_D^t = (V_D^t, E_D^t, AD_D^t, AS_D^t)$ is defined to capture the decay behaviors based on G_C at time slots $t-1$, t, and $t+1$, as shown in Figure 2 (b). G_c^{t-1} contains a congestion cluster C_m^t consisting of 4 congested road segments that are spatially connected. The congestion cluster contracts over time in the following way. Nodes 2 and 3 released their congestion states from time $t-1$ to t, then their neighboring nodes 5 and 6 released their congestion states from time t to $t+1$. In this process, there are two congestion decay paths, i.e., $r_2 \rightarrow r_5 \rightarrow r_6$ and $r_3 \rightarrow r_5 \rightarrow r_6$, where nodes 2 and 3 are sources of congestion decay. Formally, a path $pa = \langle r_{i_1}, .., r_{i_k}, ..., r_{i_K} \rangle$ is a decay path if $(i_j, i_{j+1}) \in E$ $(1 \leq j \leq K-1)$; $c_{i_j}^{t-1} = 1 (1 \leq j \leq K)$; $c_{i_1}^t = 0$, $c_{i_j}^t = 1 (2 \leq j \leq K)$, and $c_{i_j}^t = 0 (1 \leq j \leq K)$ (consequently released). With the obtained decay paths, G_D^t can be constructed by including all nodes of the decay paths into set V_D^t and all links of the decay paths into set E_D^t. AD_D^t and AS_D^t are constructed in the same way as AD_P^t and AS_P^t.

Congestion Evolution Prediction. The congestion evolution prediction problem is formulated as a link prediction problem on DAN: Given a DAN (e.g., G_P, G_D), predict the existence of directed links from a road segment r_i to r_j, based on dynamic topological structures and diverse node attributes.

4 DMLM Model for Evolution Prediction

Figure 1 (b) shows the framework of Deep Meta-Learning Model (DMLM) for predicting congestion evolutions, which consists of 5 main steps: 1) Generating asymmetric initial representation by incorporating multiple temporal correlations while preserving structure and asymmetric transitivity; 2) Learning multiple evolving temporal correlations using `meta-LSTM` modules based on static and dynamic node attributes; 3) Learning time-varying significance for fusing heterogeneous features using `meta-attention` modules; 4) Generating final source and target representations by fusing learned latent features as well as their time-varying significance. 5) Predicting the propagation/decay between road pairs based on the obtained representations. The DMLM models for propagation prediction and decay prediction are trained separately. In the rest of the paper, we discuss how to build the model for propagation prediction. The model for decay prediction is obtained in the same way.

4.1 Asymmetric Initial Representations

As the congestion propagation is inherently asymmetric, i.e., the possibility of propagation from r_i to r_j is different from that of r_j to r_i, we propose to learn asymmetric representations [Ou *et al.*, 2016; Zhou *et al.*, 2017] for the road segments. For an APN G_P, we generate a source representation $\mathbf{ep}_{i,t}^{src}$ and a target representation $\mathbf{ep}_{i,t}^{trg}$ for each node r_i at time t. Then a function ϕ takes $\mathbf{ep}_{i,t}^{src}$ and $\mathbf{ep}_{j,t}^{trg}$ as input variables to predict the likelihood of existence of propagation from r_i and r_j. Due to the asymmetric property, $\phi(\mathbf{ep}_{i,t}^{src}, \mathbf{ep}_{j,t}^{trg}) \neq \phi(\mathbf{ep}_{j,t}^{src}, \mathbf{ep}_{i,t}^{trg}) \in [0, 1]$.

Three auxiliary matrices are constructed to capture diverse temporal dependencies of evolutions, e.g., recent dependency, daily periodicity, and weekly repeatability [Sun *et al.*, 2019; Lv *et al.*, 2018]. Given APN G_P, we build 3 matrices for each time slice t, i.e., recent, daily and weekly propagation trend matrices: $M_{P,re}^t$, $M_{P,da}^t$, and $M_{P,we}^t$. Specifically, the matrix $M_{P,re}^t$, $M_{P,da}^t$ and $M_{P,we}^t$ characterize recent traffic, daily periodic and weekly periodic patterns, respectively. They are constructed as $f(lx, Tx) = \frac{1}{lx} \sum_{l=1}^{lx} M_P^{t-Tx*l}$, where $M_P^t \in \{0,1\}^{N \times N}$ and $M_P^t(i,j) = 1$ when $\{i,j\} \in E_P^t$ of G_P, lx is the number of previous days/weeks considered, Tx is the number of intervals per day/week ($Tx = 288$ for one day when the time slot is 5 minutes). For $M_{P,re}^t$, $(lx, Tx) = (6, 1)$, for $M_{P,da}^t$, $(lx, Tx) = (7, 288)$, and for $M_{P,we}^t$, $(lx, Tx) = (6, 288 * 7)$. The aforementioned parameters are based on the settings from [Guo *et al.*, 2019].

$M_{P,re}^t$, $M_{P,da}^t$ and $M_{P,we}^t$ are fed into an Non-negative Matrix Factorization (NMF) module (as shown in Figure 3 (a)) [Lee and Seung, 2001] to generate initial representations (3 source representations and 3 target representations corresponding to $M_{P,re}^t$, $M_{P,da}^t$ and $M_{P,we}^t$ respectively). The representations $\mathbf{epi}_{re,i}^{src,t}$ and $\mathbf{epi}_{re,i}^{trg,t}$ for r_i at time t is obtained based on $M_{P,re}^t$ by solving an objective function: $\min \|M_{P,re}^t - \mathbf{S}_{re}^t (\mathbf{T}_{re}^t)^T\|^2$, $s.t.$ $\mathbf{S}_{re}^t, \mathbf{T}_{re}^t \geq 0$, where $\mathbf{epi}_{re,i}^{src,t} \in \mathbb{R}^K$ is the ith row of matrix $S \in \mathbb{R}^{N \times K}$, and $\mathbf{epi}_{re,i}^{trg,t}$ is the ith row of the transposition of matrix $\mathbf{T} \in \mathbb{R}^{K \times N}$ (i.e., $(\mathbf{T})'$). $\mathbf{epi}_{da,i}^{src,t}$, $\mathbf{epi}_{da,i}^{trg,t}$, $\mathbf{epi}_{we,i}^{src,t}$, and $\mathbf{epi}_{we,i}^{trg,t}$ are calculated in the same way.

4.2 `Meta-LSTM` Component

The constructed APN G_P and ADN G_D have included diverse factors (attributes), which can be used to identify impressionable congestion evolutions. This section introduces a `Meta-LSTM` module to learn the evolving temporal correlations from multiple views. As shown in Figure 3 (b), the `Meta-LSTM` component processes the `source` representation part and the `target` representation part separately,

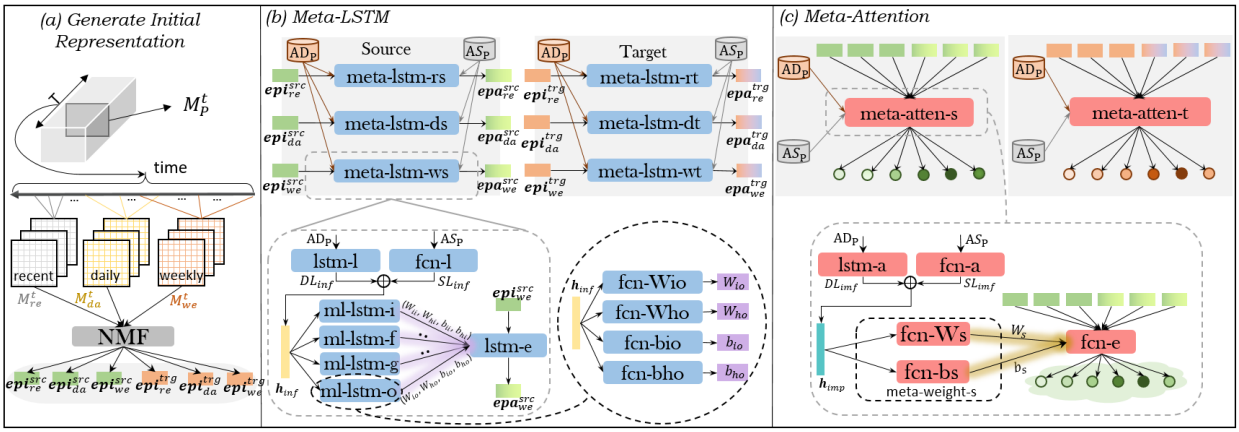


Figure 3: The three main components of DMLM: (a) Generating asymmetric initial representation, (b) Learning evolving temporal correlations using `meta-LSTM`, (c) Learning time-varying significance for feature fusion using `meta-attention`.

using two separate modules of the same structure. Each part has three separate `meta-lstm` modules, which fuse multi-view temporal correlation information (recent, daily, and weekly patterns) using heterogeneous weights that are learned via meta-learning based on diverse node attributes of APN G_P. Taking `meta-lstm-ws` for learning source representation based on G_P^t as an example, the inputs involve three parts: representation of weekly propagation trend $\mathbf{epi}_{we}^{src}$, static attributes AS_P, and dynamic attributes AD_P. `meta-lstm-rs` produces $\mathbf{epa}_{we}^{src} \in \mathbb{R}^K$.

As shown in Figure 3 (b), `meta-lstm-we` consists of a 3-layers LSTM module `lstm-l` that captures long-term and short-term dependency of dynamic attributes, a 2-layers FCN `fcn-l` that captures latent impacts of static attributes, and four meta learners for learning parameters in meta-LSTM (i.e., `lstm-e`). At each time step, the `lstm-l` learns a latent vector $DL_{inf} \in \mathbb{R}^{Dl_o^{ml}}$ from node r_i's dynamic attributes $AD_P^t(i) \in \mathbb{R}^{F_d}$ (the i-th row of AD_P^t), which incorporates dynamic latent information. The `fcn-l` learns a $SL_{inf} \in \mathbb{R}^{Df_o^{ml}}$ from $AS_P(i) \in \mathbb{R}^{F_s}$ to incorporate static latent information. The knowledge learned from both static and dynamic attributes are concatenated before being fed into the four meta-learners, i.e., $\mathbf{h}_{inf} = DL_{inf} \oplus SL_{inf}$. The four meta-learners learn weight parameters for input gate, forget gate, cell, and output gate in `lstm-e` module, based on $\mathbf{h}_{inf}$. Based on the learned weights from meta leaners, the `lstm-e` takes $\mathbf{epi}_{we}^{src}$ as input to generate $\mathbf{epa}_{we}^{src}$.

The four meta learners are of the same structure as shown in Figure 3 (b). For example, the meta learners for output gate (`ml-lstm-o`) relies on four separate 2-layers FCN modules to learn 4 parameters for the output gate. Specifically, the four FCN modules (i.e., `fcn-wio`, `fcn-who`, `fcn-bio` and `fcn-bho`) take $\mathbf{h}_{inf}$ as input and produce 4 parameters $\mathbf{W}_{io} \in \mathbf{R}^{K\times K}$, $\mathbf{W}_{ho} \in \mathbb{R}^{K\times K}$, $\mathbf{b}_{io} \in \mathbb{R}^K$, and $\mathbf{b}_{ho} \in \mathbb{R}^K$ for the output gate of `lstm-e`. In the same way, the parameters of `lstm-e`'s other three parts (input gate, forget gate, cell) are learned using 3 meta learners. Then, `lstm-e` takes $\mathbf{epi}_{we}^{src}$ as input and produces $\mathbf{epa}_{we}^{src}$ based on following equations:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_{hi}),\\
\mathbf{f}_t &= \sigma(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_{hf}),\\
\mathbf{g}_t &= \sigma(\mathbf{W}_{ig}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_{hg}),\\
\mathbf{o}_t &= \sigma(\mathbf{W}_{io}\mathbf{x}_t + \mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_{ho}),\\
\mathbf{C}_t &= \mathbf{f}_t \circ \mathbf{C}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t,\\
\mathbf{h}_t &= \mathbf{o}_t \circ \tanh(\mathbf{C}_t).
\end{aligned}
$$

where $\circ$ is Hadamard product, and σ is sigmoid functuion.

4.3 `Meta-Attention` Component

APN G_P has included various factors that impact the traffic situations. The dynamic as well as the static attributes have time-varying impact on the congestion evolutions (as highlighted earlier). To capture such dynamic significance, we designed a `Meta-Attention` component to merge multiple temporal correlations as well as static and dynamic attributes. As shown in Figure 3 (c), `Meta-Attention` component relies on two separate meta-attention modules to learn source and target representations (`meta-atten-s` and `meta-atten-t`). In `meta-atten-s`, a 3-layers LSTM layer (`lstm-a`) is used to learn a latent vector representation ($DL_{imp} \in \mathbb{R}^{Df_o^{ma}}$) for dynamic attributes and a 2-layers FCN layer (`fcn-a`) is used to learn a latent vector representation ($SL_{imp} \in \mathbb{R}^{Df_o^{ma}}$) for the static attributes. Then, $\mathbf{h}_{imp} = DL_{imp} \oplus SL_{imp}$ is fed into a meta learner, `meta-weight-s`, to learn weight parameters for the attention module `fcn-e`. The meta learner uses two separate two-layers FCNs, `fcn-Ws` and `fcn-bs`, to learn weight matrix and bias terms for `fcn-e`. Then, `fcn-e` takes three initial representations ($\mathbf{epi}_{re}^{src}$, $\mathbf{epi}_{da}^{src}$,$\mathbf{epi}_{we}^{src}$) together with three attributed representations ($\mathbf{epa}_{re}^{src}$,$\mathbf{epa}_{da}^{src}$,$\mathbf{epa}_{we}^{src}$) as inputs, and produces corresponding six scores a_{re}^{epi}, a_{da}^{epi}, a_{we}^{epi}, a_{re}^{epa}, a_{da}^{epa} and a_{we}^{epa} based on the following equations:

$$
a_x^y = \frac{\exp \alpha_x^y}{\sum_{l,m} \exp \alpha_l^m},\ x, l \in \{re, da, we\}, y, m \in \{epi, epa\}.
$$

$$
\alpha_x^y = \texttt{fcn-e}(y_x),\ x \in \{re, da, we\}, y \in \{epi, epa\}.
$$

With the Meta-Attention module, each node in G_P obtains six attention scores for source representation learning: $a^{epi}_{re,src}$, $a^{epi}_{da,src}$, $a^{epi}_{we,src}$, $a^{epa}_{re,src}$, $a^{epa}_{da,src}$, $a^{epa}_{we,src}$; and six attention scores for target representation learning: $a^{epi}_{re,trg}$, $a^{epi}_{da,trg}$, $a^{epi}_{we,trg}$, $a^{epa}_{re,trg}$, $a^{epa}_{da,trg}$, $a^{epa}_{we,trg}$.

4.4 Fusion and Prediction

For a road segment r_i at time t in APN G_P, a source representation $\mathbf{ep}^{src} \in \mathbb{R}^K$ and a target representation $\mathbf{ep}^{trg} \in \mathbb{R}^K$ are obtained via the following fusion process.

$$\mathbf{ep}^{src} = \sum_{x,y} a^{y}_{x,src} * y^{src}_{x},\ x \in \{re, da, we\}, y \in \{\mathbf{epi}, \mathbf{epa}\}.$$

$$\mathbf{ep}^{trg} = \sum_{x,y} a^{y}_{x,trg} * y^{trg}_{x},\ x \in \{re, da, we\}, y \in \{\mathbf{epi}, \mathbf{epa}\}.$$

The likelihood of a directed link from r_i to r_j is calculated as inner product of r_i's source representation and r_j's target representation, i.e., $y^p(i,j) = \sigma(\mathbf{ep}^{src}_i \cdot \mathbf{ep}^{trg}_j)$, σ is sigmoid function. There exists a link if $y^p(i,j) \geq 0.5$ and no link otherwise.

Loss function. The loss function for training DMLM to predict propagation behavior is defined as follows.

$$L_P = -\frac{1}{n}\sum_{k=1}^{n} \hat{y^p}_k \cdot \ln {y^p}_k + (1 - \hat{y^p}_k) \cdot \ln(1 - {y^p}_k)$$

where n is the number of samples, ${y^p}_k$ is prediction, and $\hat{y^p}_k$ is ground truth.

5 Experiments

5.1 Dataset

Road Network. The road network for our experiments is obtained from OpenStreetMap[1], which covers a rectangle area in Downtown of Singapore (Southwest: 1.2718, 103.8002; Northeast: 1.3323, 103.8653) consisting of 1858 road segments. The topological information and static attributes of roads are extracted (as discussed in Section 3).

POIs. The POIs are collected from government website [2]. We group all POIs into 14 categories: shopping services, religion building, business, hotel, residence, education, food, government, scenic spot, medical care, sports, entertainment, bus stop, MRT station.

Traffic Data. The historical traffic speeds are calculated based on bus trajectories derived from bus arrival data[3]. Speeds are aggregated every 5 minutes. Then, a road segment is detected as congested at a time interval if its traffic speed is slower than a threshold value. Similar to existing works, we tested the performance using several thresholds. For road r_i, the threshold θ^p_i is selected as different percentiles of traffic speeds of each road, denoted as p. For example, p = 90% indicates that 90% of the road's speed values in $\mathbf{s}_i$ are larger than θ^p_i. With this threshold, 10% of the traffic speeds will be identified as congested in $\mathbf{c}_i$. In the experiments, p in $\{90\%, 75\%, 60\%\}$ are tested for the prediction performance under different congestion situations [Nguyen *et al.*, 2016]. The traffic data is collected from Aug. 01 to Nov. 30, 2018, where the data of the first 90 days (75%) is used for training and the remaining data is used for testing.

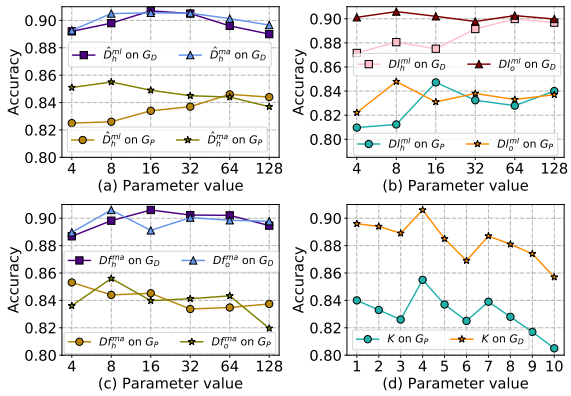


Figure 4: Impacts of model parameters.

5.2 Evaluation Metrics and Baseline Methods

Metrics. The performance of the proposed methods is evaluated using metrics accuracy and F1-score.

Baselines. 1) STC [Nguyen *et al.*, 2016] extracts frequently appearing congestion propagation relationship by mining frequent subtrees. Then, a dynamic Bayesian network is constructed using the obtained patterns to predict propagations. 2) Pro-Graph [Xiong *et al.*, 2018] works on a graph structure and predicts where certain congestions will propagate to the near future. 3) MMDNE [Lu *et al.*, 2019] proposes temporal embedding method to incorporate evolution of dynamic networks at micro- and macro-level, and predicts links between node pairs. 4) GCN-GAN [Lei *et al.*, 2019] considers non-linear characteristics and link weights in dynamic network leveraging on graph convolutional network (GCN), LSTM and generative adversarial network (GAN) for link prediction. 5) ANRL [Zhang *et al.*, 2018] employs a neighbor enhancement autoencoder to model node attributes and a skip-gram model to formulate structure among nodes for learning node representations. Link prediction is then performed based on cosine similarity function. 6) node2bits [Jin *et al.*, 2019] uses a time- and attribute-aware framework to study interactions of users in heterogeneous networks and predict whether two users correspond to the same entity (has a link). The first two baselines are the most recent works that investigate congestion evolution prediction problems. Baselines 3) and 4) are link prediction methods considering temporal dimension properties. Baseline 5) incorporates node attributes but ignore properties over temporal dimension, while baseline 6) considers both node attributes and temporal attributes. We also include the ablation version of our method by removing major components. 7) Ours-woML (without meta-LSTM) employs the original LSTM instead of the meta-LSTM structure. 8) *Ours-woMA* (without meta-Attention)

[1]https://www.openstreetmap.org/export

[2]https://data.gov.sg/dataset?q=Places+of+Interest

[3]https://www.mytransport.sg/content/mytransport/home/dataMall.html

employs a widely used attention mechanism instead of the proposed meta-Attention structure.

Table 1: Overall performance on congestion propagation prediction.

	Accuracy			F1-score		
	90%	75%	60%	90%	75%	60%
STC	0.652	0.648	0.671	0.685	0.677	0.769
Pro-Graph	**0.860**	0.692	0.547	0.097	0.117	0.118
MMDNE	0.487	0.504	0.496	0.396	0.403	0.469
GCN-GAN	0.503	0.501	0.508	0.014	0.004	0.034
ANRL	0.499	0.50	0.498	0.665	0.667	0.663
Node2Bits	0.518	0.542	0.534	0.635	0.675	0.667
Ours-woML	0.830	0.804	0.837	0.855	0.829	0.850
Ours-woMA	0.851	0.849	0.863	0.863	0.857	0.871
Ours (DMLM)	0.859	**0.862**	**0.876**	**0.872**	**0.874**	**0.883**

Table 2: Overall performance on congestion decay prediction.

	Accuracy			F1-score		
	90%	75%	60%	90%	75%	60%
MMDNE	0.375	0.434	0.397	0.319	0.314	0.318
GCN-GAN	0.501	0.503	0.501	0.003	0.003	0.003
ANRL	0.497	0.50	0.499	0.661	0.665	0.662
Node2Bits	0.489	0.487	0.487	0.154	0.305	0.274
Ours-woML	0.798	0.752	0.698	0.843	0.704	0.531
Ours-woMA	0.883	0.864	0.852	0.892	0.877	0.870
Ours (DMLM)	**0.939**	**0.906**	**0.887**	**0.942**	**0.913**	**0.897**

5.3 Experiment Setup and Hyper-parameters

We implemented the `DMLM` model using PyTorch framework on Intel(R) Xeon(R) CPU E5-1650 v2 @ 3.50GHz with 32G RAM. The source code is available at https://github.com/HelenaYD/DMLM. The critical hyper-parameters are optimized via grid search as follows. Size of hidden layers of meta learners's FCNs in `Meta-LSTM` and `Meta-Attention`: $\hat{D}_h^{ml} = 64$, $\hat{D}_h^{ma} = 8$; Size of hidden and output layers in `lstm-l` and `lstm-a`: $Dl_h^{ml} = 16$, $Dl_o^{ml} = 8$, $Dl_h^{ma} = 8$, $Dl_o^{ma} = 32$; Size of hidden and output layers in `fcn-l` and `fcn-a`: $Df_h^{ml} = 4$, $Df_o^{ml} = 16$, $Df_h^{ma} = 4$, $Df_o^{ma} = 8$; and the dimension of final representations $K = 4$. For decay prediction, the above parameters are set to: 16, 16, 64, 8, 128, 8, 64, 16, 16, 8, respectively. In addition, the learning rate is 0.0001 and the batch size is 40.

5.4 Results and Analysis

Table 1 and Table 2 show the comparison with 6 baselines and 2 ablation versions of our method. In addition to testing the prediction performance on congestion propagation, we also predict the congestion decay by training another DMLM using ADN G_D. The results show that our method outperforms all the baselines in both accuracy or f1-score for both propagation and decay prediction. STC and Pro-Graph are designed for congestion propagation, and cannot be easily extended to decay prediction. Thus we only tested their performance on propagation prediction. Pro-Graph obtains comparable accuracy with our DMLM when $p = 90\%$ as it relies on propagation patterns (e.g., repeatability) to make predictions. As such, it produces higher accuracy for heavy congestions that repeat frequently, and the accuracy decreases rapidly with decreasing p. Both MMDNE and GCN-GAN considered temporal dependency and structure information but failed to obtain satisfying results. This indicates that it is critical to incorporate diverse node attributes in addition to the temporal dependency and network structure information. On the other hand, ANRL considered node attributes but ignore temporal dimension, leading to poor performance. In general, the baselines produce low-quality solutions because they either did not sufficiently incorporate the spatiotemporal correlations or failed to capture the dynamic significances of both the dynamic and static node attributes. On the other hand, Ours-woML and Ours-woMA produce obvious performance degradation for both propagation and decay prediction. This demonstrates that the `Meta-LSTM` and `Meta-Attention` modules play important roles in the DMLM. In addition, combining NMF with diverse temporal dependencies captures the underlying evolution trends, which makes prediction more stable.

Figure 4 illustrates the impacts of several factors on the prediction performance, including $\hat{D}_h^{ml}$, $\hat{D}_h^{ma}$, Dl_h^{ml}, Dl_o^{ml}, Df_h^{ma}, Df_o^{ma}, and K. These factors affect the dimensions of intermediate feature vectors. As shown in Figure 4 (a), increasing $\hat{D}_h^{ml}$ and $\hat{D}_h^{ma}$ first leads to perform improvement, and then results in performance degradation. In general, feature vectors that are too short cannot sufficiently characterize the complex correlations, while vectors that are too long not only fail to improve representation capability, but also lead to higher computation overhead. The parameters, e.g., Dl_h^{ml} and Dl_o^{ml} in Figure 4 (c), Df_h^{ma} and Df_o^{ma} in (d), also produce similar trends (the other parameters are not shown due to the page limit). The figure also shows that even under similar trends, the best parameter setting for propagation prediction is different from decay prediction.

6 Conclusion

This paper proposed methods to model and predict the likelihood of congestion evolution between any pair of road segments (connected via single or multiple paths). Dynamic attributed networks were constructed to incorporate both dynamic and static impact factors while preserving dynamic topological structures. A DMLM model was proposed to learn representations of road segments, which enable accurate prediction of congestion evolution. DMLM relies on meta-LSTM and meta-Attention techniques to incorporate the heterogeneous and time-varying impacts of multiple correlations, and various dynamic and static node attributes. Experiments on real-world datasets showed that the proposed method achieves significantly better results than all state-of-the-art methods.

Acknowledgements

This research project is supported in part by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme with the Technical University of Munich at TUM-CREATE.

References

[An *et al.*, 2016] Shi An, Haiqiang Yang, Jian Wang, Na Cui, and Jianxun Cui. Mining urban recurrent congestion evolution patterns from gps-equipped vehicle mobility data. *Inform. Sciences*, 373:515–526, 2016.

[An *et al.*, 2018] Shi An, Haiqiang Yang, and Jian Wang. Revealing recurrent urban congestion evolution patterns with taxi trajectories. *ISPRS Interl. J. of Geo-Inform.*, 7(4):128, 2018.

[Anwar *et al.*, 2016] Tarique Anwar, Chengfei Liu, Hai L Vu, and Md Saiful Islam. Tracking the evolution of congestion in dynamic urban road networks. In *CIKM*, pages 2323–2328, 2016.

[Cui *et al.*, 2019] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2019.

[Di *et al.*, 2019] Xiaolei Di, Yu Xiao, Chao Zhu, Yang Deng, Qinpei Zhao, and Weixiong Rao. Traffic congestion prediction by spatiotemporal propagation patterns. In *MDM*, pages 298–303. IEEE, 2019.

[Guo *et al.*, 2019] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*, volume 33, pages 922–929, 2019.

[Jin *et al.*, 2019] Di Jin, Mark Heimann, Ryan A Rossi, and Danai Koutra. node2bits: Compact time-and attribute-aware node representations for user stitching. In *ECML-PKDD*, pages 483–506. Springer, 2019.

[Khajeh-Hosseini and Talebpour, 2019] Mohammadreza Khajeh-Hosseini and Alireza Talebpour. Back to the future: Predicting traffic shockwave formation and propagation using a convolutional encoder-decoder network. In *ITSC*, pages 1367–1372. IEEE, 2019.

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NeurIPS*, pages 556–562, 2001.

[Lei *et al.*, 2019] Kai Lei, Meng Qin, Bo Bai, Gong Zhang, and Min Yang. Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks. In *INFOCOM*, pages 388–396. IEEE, 2019.

[Li *et al.*, 2018] Jundong Li, Kewei Cheng, Liang Wu, and Huan Liu. Streaming link prediction on dynamic attributed networks. In *WSDM*, pages 369–377, 2018.

[Liao *et al.*, 2018] Binbing Liao, Jingqing Zhang, Chao Wu, Douglas McIlwraith, Tong Chen, Shengwen Yang, Yike Guo, and Fei Wu. Deep sequence learning with auxiliary information for traffic prediction. In *KDD*, pages 537–546, 2018.

[Lichtenwalter *et al.*, 2010] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *KDD*, pages 243–252, 2010.

[Lu *et al.*, 2019] Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S Yu, and Yanfang Ye. Temporal network embedding with micro-and macro-dynamics. In *CIKM*, pages 469–478, 2019.

[Lv *et al.*, 2018] Zhongjian Lv, Jiajie Xu, Kai Zheng, Hongzhi Yin, Pengpeng Zhao, and Xiaofang Zhou. Lc-rnn: A deep learning model for traffic speed prediction. In *IJCAI*, pages 3470–3476, 2018.

[Martínez *et al.*, 2016a] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. Adaptive degree penalization for link prediction. *Journal of Computational Science*, 13:1–9, 2016.

[Martínez *et al.*, 2016b] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, 49(4):1–33, 2016.

[Nguyen *et al.*, 2016] Hoang Nguyen, Wei Liu, and Fang Chen. Discovering congestion propagation patterns in spatio-temporal traffic data. *IEEE Trans. Big Data*, 3(2):169–180, 2016.

[Ou *et al.*, 2016] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *KDD*, pages 1105–1114, 2016.

[Sun *et al.*, 2018] Jian Sun, Zian Ma, and Xuemei Chen. Some observed features of traffic flow phase transition at urban expressway diverge bottlenecks. *TRANSPORTMETRICA B*, 6(4):320–331, 2018.

[Sun *et al.*, 2019] Yidan Sun, Guiyuan Jiang, Siew-Kei Lam, Shicheng Chen, and Peilan He. Bus travel speed prediction using attention network of heterogeneous correlation features. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 73–81. SIAM, 2019.

[Xiong *et al.*, 2018] Haoyi Xiong, Amin Vahedian, Xun Zhou, Yanhua Li, and Jun Luo. Predicting traffic congestion propagation patterns: A propagation graph approach. In *SIGSPATIAL*, pages 60–69, 2018.

[Yue *et al.*, 2018] Wenwei Yue, Changle Li, and Guoqiang Mao. Urban traffic bottleneck identification based on congestion propagation. In *ICC*, pages 1–6. IEEE, 2018.

[Zhang *et al.*, 2017] Tianqi Zhang, Lishan Sun, Liya Yao, and Jian Rong. Impact analysis of land use on traffic congestion using real-time traffic and poi. *Journal of Advanced Transportation*, 2017, 2017.

[Zhang *et al.*, 2018] Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, and Can Wang. Anrl: Attributed network representation learning via deep neural networks. In *IJCAI*, volume 18, pages 3155–3161, 2018.

[Zhou *et al.*, 2017] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *AAAI*, pages 2942–2948, 2017.