Di, Xiaolei; Xiao, Yu; Zhu, Chao; Deng, Yang; Zhao, Qinpei; Rao, Weixiong

Traffic congestion prediction by spatiotemporal propagation patterns

Published in:
Proceedings - 2019 20th International Conference on Mobile Data Management, MDM 2019

DOI:
10.1109/MDM.2019.00-45

Published: 01/06/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Di, X., Xiao, Y., Zhu, C., Deng, Y., Zhao, Q., & Rao, W. (2019). Traffic congestion prediction by spatiotemporal propagation patterns. In *Proceedings - 2019 20th International Conference on Mobile Data Management, MDM 2019* (pp. 298-303). Article 8788784 (Mobile Data Management Conference; Vol. 2019-June). IEEE. https://doi.org/10.1109/MDM.2019.00-45

Traffic Congestion Prediction by Spatiotemporal Propagation Patterns

Xiaolei Di[§] Yu Xiao[†] Chao Zhu[†] Yang Deng[§] Qinpei Zhao[§] Weixiong Rao[§]
[§] Tongji University, Shanghai, China [†] Aalto University, Espoo, Finland
{xldi, wxrao}@tongji.edu.cn {yu.xiao,chao.1.zhu}@aalto.fi

***Abstract*—Accurate prediction of traffic congestion at the granularity of road segment is important for planning travel routes and optimizing traffic control in urban areas. Previous works often calculated only the average congestion levels of a large region covering many road segments and did not take into account spatial correlation between road segments, resulting in inaccurate and coarse-grained prediction. To overcome these issues, we propose in this paper CPM-ConvLSTM, a spatiotemporal model for short-term prediction of congestion level in each road segment. Our model is built on a spatial matrix which incorporates both the congestion propagation pattern and the spatial correlation between road segments. The preliminary experiments on the traffic data set collected from Helsinki, Finland prove that CPM-ConvLSTM greatly outperforms 6 counterparts in terms of prediction accuracy.**

I. Introduction

Traffic congestion has received much attention in recent years due to its great impact on people's daily life. Fine-grained congestion prediction for each road segment in urban cities can help people schedule travel routes in advance and assist traffic control to relieve traffic congestion. Consequently, it is important to design an accurate congestion prediction model at the granularity of road segment.

The widely available sensors, such as GPS receivers and traffic cameras, provide rich data for mobility analysis in urban areas. One application is to predict traffic congestion based on spatial and temporal patterns shown in the traffic. For example, in [1], [2], classic ARIMA models were used for predicting traffic congestion based on previous observations. However, these works did not take into account the spatial correlation between road segments, which affects the congestion propagation across road segments. On the other hand, many works [3], [4]simply divide an area of interest into grid cells and calculate the average value in each cell, ignoring the variation between road segments within each grid cell.

More recently, the two works [5], [6] identify congestion propagation patterns across a subset of road segments. For example, in Figure 1, when the road segment r_1 becomes congested at a certain time of day T_1 (e.g., Monday 8:00AM) because many people living around r_1 rush to their office. After a while, the congestion on r_1 may cause the congestion on the upstream roads r_2 and r_3 at T_2 and later the congestion on r_4 at T_3. Regarding temporal dimension, the congestion of road has temporal auto-correlations. However, such congestion propagation patterns do not take into account the geographical positions of road segments.

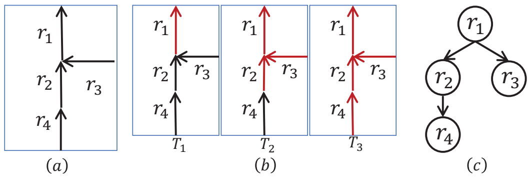

Fig. 1. Example of congestion propagation pattern

In this paper, we propose a spatiotemporal traffic prediction model, namely, CPM-ConvLSTM, to make short-term prediction of congestion level for each road segment. The key idea is to define a spatial matrix which incorporates both the congestion propagation pattern across road segments and the spatial correlation between road segments. Given the time series of historical spatial matrices, we exploit a recent popular spatiotemporal deep learning model ConvLSTM [7] by taking as input the time series of historical spatial matrices and predicting the future short-term spatial matrix.

The contributions of our study are summarized as follows:

- Unlike the tree structures presented in [5], [6], we represent congestion prorogation patterns with a directed acyclic graph (DAG).
- The issue of the widely used grid map can not make accurate prediction at the granularity of road segment. The proposed spatial matrix can overcome this issue by incorporating the DAG-based congestion prorogation patterns. The spatial matrix can be comfortably fed into ConvLSTM for traffic prediction.
- Our preliminary experiments on the traffic data set collected from Helsinki, Finland prove that CPM-ConvLSTM greatly outperforms 6 counterparts in terms of prediction accuracy.

The rest of the paper is organized as follows. Section II first gives the problem setting, and Section III describes the design of proposed CPM-ConvLSTM. After that, Section IV reports our experiments, and Section V reviews the related works. Finally, Section VI concludes this paper.

II. Problem Definition

Before giving the problem definition, we introduce the following definitions.

Definition 1 ***Road network***: *A road network $\mathcal{N} = \{I, R\}$ contains a set of road intersections I (vertices) and directed road* *segments R (edges) connecting intersections. Let* $R = \{r_1, ..., r_n\}$ *be a set of N road segments within a road network.*

Definition 2 ***Congestion level****: The congestion level is a real number in the range of 0.0 to 10.0, A bigger value indicates more congested traffic.*

Definition 3 ***Traffic congestion propagation pattern****: The traffic congestion propagation pattern indicates that a strong spatialtemporal relationship among the congestion levels of certain road segments in a road network* $\mathcal{N}$ *(which can be treated as a subgraph of* $\mathcal{N}$*), the example can be seen in Figure 1. Let* $CP = \{cp_1, ..., cp_k\}$ *be a set of traffic congestion propagation patterns. Given a set of road segments R, a congestion propagation pattern* cp_i *is defined as a subset of road segments such that* $\emptyset \subset cp_i \subseteq R$.

Given the definition of traffic congestion propagation pattern *CP*, we could use a graph structure (typically a DAG: directed acyclic graph) to describe the *CP*. For example, Figure 1(c) indicates an example pattern, which can be modeled by a tree structure (a special case of DAG).

We use a *spatial matrix* to maintain a traffic congestion propagation pattern with respect to (w.r.t) a certain time interval (e.g., one hour from Monday 8:00AM to 9:00AM). Given a 12-hour time span from Monday 8:00AM to 8:00PM, a given pattern is then associated with 12 spatial matrices. Formally, a spatial matrix of certain pattern w.r.t a time interval is defined as follows.

Definition 4 ***Spatial Matrix****: We build a grid map by dividing the area covered by the road segments in each* CP *into grid cells and convert the grid map into a spatial matrix. In a spatial matrix, each matrix element indicates the congestion level of one road segment corresponding to the connectivity between nodes in the DAG representing traffic propagation pattern.*

More details of spatial matrix will be given in Section III.

Problem 1 *Given a historical dataset of congestion levels in a road network* $\mathcal{N}$ *including a set of road segments R from* T_1 *to* T_n*. After constructing several* CPs *from* $\mathcal{N}$*, for each CP in each time period* $[T_i, T_{i+1}]$*(with* $1 \le i \le n-1$*), we build a spatial matrix* M_i*. Given the* $(n-1)$ *matrices* $M_1...M_{n-1}$*, we predict the spatial matrix* M_n *for the time span from* T_n *to* T_{n+1} *for every* CP.

III. Design of CPM-ConvLSTM

A. Overview

In this section, we give an overview of the solution, namely, CPM-ConvLSTM, to Problem 1. As shown in Figure 2, the algorithm consists of three steps.

(*1*) **Congestion propagation pattern graph construction**: In the first step, given the historical dataset of congestion levels in a road network, we need to detect *CPs* in the road network. After removing the redundancy of patterns, we use a graph structure to maintain *CP*.

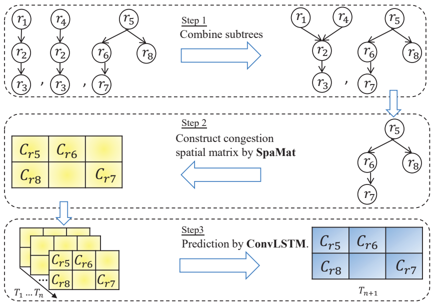

Fig. 2. Architecture of CPM-ConvLSTM

(*2*) **Spatial matrix construction**: The second step is to construct a spatial matrix w.r.t each *CP* for a certain time period. The matrix is constructed with help of the geographical positions of the road segments in *CP*.

(*3*) **Congestion level prediction**: In the third step, for a given *CP*, when given the input of those spatial matrices from T_i to T_n, we predict the congestion level of related road segment at T_{n+1}, e.g. ConvLSTM [7].

B. Congestion propagation pattern graph construction

STCTree [5] can effectively detect *CPs* which are represented by congestion tree structure. Given the historical database of congestion levels in a road network, STCTree generates a forest of congestion trees. However, the forest leads to significant redundancy. For example, in the 1st step of Figure 2, the road segments r_2 and r_3 repeatedly appear within two congestion trees.

The redundancy above leads to the following issues. Since a road segment, e.g., r_2 in Figure 2, could appear within multiple congestion propagation pattern trees, the associated congestion levels are repeatedly predicted in these trees. Given the congestion levels, how to merge them into a single one is non-trivial. For example, using an average of the congestion levels may not correctly estimate the true congestion level on the road segment.

Consequently, to eliminate the redundancy in the forest of congestion trees, we propose to combine the congestion trees involving redundancy into a DAG. Given the two congestion trees involving the two redundant road segments r_2 and r_3, Figure 2 gives an example DAG which removes the redundant segments r_2 and r_3.

C. Spatial matrix construction

Given a spatial pattern (i.e., a DAG), we need to represent the DAG by a matrix which is next taken as the input of the 3rd step (congestion level prediction). Using the traditional adjacent matrix to represent the DAG could be useful to indicate the topology information of a DAG. However, the adjacent matrix might miss the geographical relation among road segments. To this end, we develop a spatial matrix of

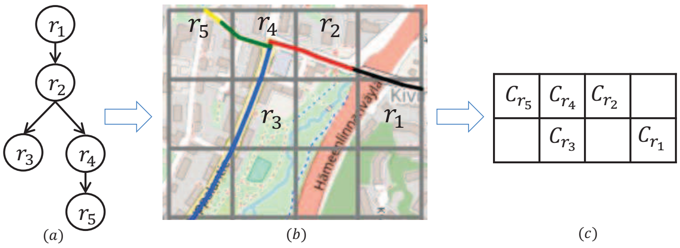

Fig. 3. Example of spatial matrix construction

Algorithm 1 SpaMat:spatial matrix construction

```
Input: CGraph:a congestion graph
Output: CSMat:the spatial matrix for the congestion propagation pattern.
 1: roadGrid = GridMap();
 2: positionRoad[(0,0)] = firstRoad;
 3: for each road ∈ CGraph do
 4:     position = RelativePosition(road.refRoad, road, roadGrid);
 5:     while position ∉ positionRoad.keys do
 6:         refRoad = positionRoad[position];
 7:         position = RelativePosition(refRoad, road, roadGrid);
 8:     end while
 9:     roadPosition[road] = position;
10:     positionRoad[position] = road;
11: end for
12: minX, minY = GetMinXY(positionRoad);
13: for each position ∈ positionRoad.keys do
14:     roadMatId = (position[1] − minY) ∗ cols + (position[0] − minX) + 1;
15:     spatialMatrix[roadMatId] = C_{positionRoad[position]};
16: end for
```

congestion levels(definition 4), named CSMat, for each *CP*. Each entry of the CSMat corresponds to a divided spatial grid. The grid contains at most one road segment with its associated congestion level.

Suppose that we have derived a *CP* in Figure 3(a) involving 5 road segments $r_1, ..., r_5$. Next, if we divide the area covered by the corresponding road network into a 4×3 grid map, a road segments (e.g., r_2) appears within multiple grid cells and a grid cell (e.g., at the 1st row and 2nd column) contains zero or multiple partial road segments as shown in Figure 3(b). Unlike this traditional grid map, the CSMat makes sure that each matrix element corresponds to at most one entire road segment ignoring the distance of it and an entire road segment corresponds to only one element. In Figure 3(c), the CSMat involves only two rows and four columns. Here, a road segment e.g. r_3, though appearing across 3 grid cells (see Figure 3(b)), only one matrix element at the 2nd row and 2nd column corresponds to r_3. Though a road segment appears within only one matrix element, using a CSMat can roughly approximate the geographical information of road segments.

The general idea of constructing a CSMat is described as follows. Given a input congestion propagation pattern DAG named CGraph, we first start from the root of CGraph (i.e., the node with an indegree equal to 0), and traverse the CGraph level by level in a top-down manner. For each traversed node in the CGraph, we make sure that only one matrix element corresponds to the visited node. The matrix elements can approximate the geographical position of road segments within a road network. Thus, the CSMat can be intuitively treated as a distorted grid map which can approximate the topology information of CGraph.

Algorithm 1 gives the pseudocode of constructing the congestion spatial matrix based on CGraph. Here we take the CGraph in Figure 3(a) as example. First, we divide the area of interest involving the CGraph into a $H \times W$ grid map, Figure 3(b) gives the 4×3 grid map dividing the area of interest into $100m \times 100m$ grid cells. The first visited node (i.e., a road segment such as r_1 in Figure 3(a)) is the one with an indegree equal to 0, and we choose a grid cell in the grid map as the matrix element corresponding to the node. In case that this road segment appears in multiple grid cells, we choose only one of them which covers more of the road segment than the other grids. After that, we traverse the remaining road segments of CGraph hierarchically (line 3). For those road segments linked by the edges in CGraph, we ensure that the corresponding matrix elements in CSMat are adjacent. For instance, in Figure 3(c), r_1 is linked to r_2, and the corresponding matrix elements are adjacent. Similarly, r_2 is linked to r_3 and r_4, and the element corresponding to r_2 is also adjacent to the ones of r_3 and r_4. Here, the key of using the adjacent element is to approximate the geographical positions of road segments within a road network. When all road segments in CGraph are traversed, we then output the CSMat.

D. Convolutional LSTM

Convolutional LSTM (ConvLSTM) [7] is an extension of LSTM by combining convolutions with LSTM. ConvLSTM uses convolutions directly as part of reading input into the LSTM units, which is different from the hybrid model of CNN and LSTM. The main equations in ConvLSTM are shown as follows.

$$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \quad (2)$$

$$\mathcal{C}_t = f_{t-1} \circ \mathcal{C} + i_t \circ tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \quad (4)$$

$$\mathcal{H}_t = o_t \circ tanh(\mathcal{C}_t) \quad (5)$$

where i_t, f_t, o_t are the output of input gate, forget gate and output gate for timestamp t respectively. $\mathcal{C}_t, \mathcal{H}_t$ are the cell output and hidden state of the cell at time step t, the $*$ and $\circ$ denote the convolution operation and Hadamard product, respectively.

The input of ConvLSTM is a three-dimension spatiotemporal tenser, where the first two dimensions are spatial dimension and the output is a two-dimension spatial tenser at next time step. With the help of ConvLSTM, given the n spatial matrices from T_1 to T_n, we predict the spatial matrix at T_{n+1}, as shown

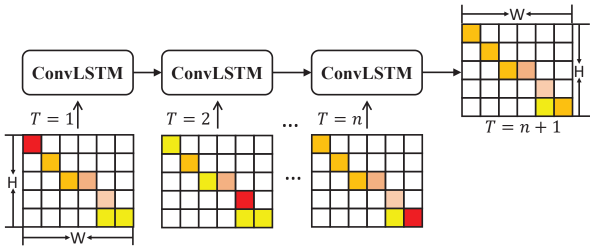

Fig. 4. ConvLSTM model

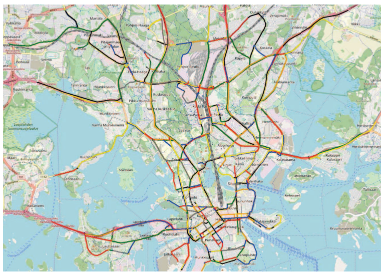

Fig. 5. Study Area

in Figure 4. We use ConvLSTM to extract spatiotemporal features of each *CP* (represented by CSMat) from T_1 to T_n and predict the values at T_{n+1} based on the extracted features.

IV. Evaluation

In this section, we first introduce the data used in experiments, and next evaluate the performance of CPM-ConvLSTM against six counterparts.

A. Data Description

In this paper, we collected the traffic congestion data of Helsinki, Finland by using HERE Traffic API. In Figure 5, we selected the area with longtitude from 24.83903E to 25.00858E and latitude from 60.14826N to 60.23113N. The area includes 689 road segments. As shown in Table I, each data sample has a unique road segment ID, the latitude and longitude coordinates of the road from start point to end point and the congestion level at a specific time. The requests were sent to HRER sever every 60 seconds from 2018/09/01 to 2018/10/06. The dataset is divided into two subsets: weekdays and weekends data sets since the congestion levels in weekdays and weekends are significantly different. We extract 105 *CPs* including 553 road segments from the historical congestion data. In addition, we use the first 80% of the crawled data (from the start date) as the training data, and the remaining one as the test data to evaluate the performance.

TABLE I
Example of a road congestion record

Time	2018-09-01 20:34:00
Road ID	1
Start	60.21063/24.88800
End	60.20902/24.248890
Congestion Level	3.5

B. Performance Metrics and Baseline Method

Performance Metrics We use Mean Squared Error (MSE) and Mean Absolute Error (MAE) for performance evaluation.

$$MSE = \sum_{n} MSE = \frac{\sum_{j}^{N}(\frac{1}{S}\sum_{i}^{S}(y_i - \overline{y}_i)^2)}{N} \tag{6}$$

$$MAE = \sum_{n} MAE = \frac{\sum_{j}^{N}(\frac{1}{S}\sum_{i}^{S}|y_i - \overline{y}_i|)}{N} \tag{7}$$

where y_i and $\overline{y}_i$ are the actual and predicted congestion levels of a specific road segment, respectively. S is the total number of road segments in a *CP* and N is the total number of congestion propagation pattern graphs. A small MSE/MAE indicates good prediction performance.

Counterparts: We mainly compare the proposed CPM-ConvLSTM against 6 following counterparts.

- LSTM is used to predict the congestion level independently at the granularity of road segment by treating the congestion levels of each road segment as long-term time series data. We use a two-layer LSTMs for the time series prediction.
- CNN-LSTM: A 2D CNN is used to extract spatial features, and a two-layer LSTM is used to extract temporal features. By combining the CNN and LSTM, the hybrid model can perform the prediction based on a grid map. Similar to the LSTM approach above, we again perform the prediction at the granularity of road segment.
- CP-CNN-LSTM: We improve CNN-LSTM by incorporating the proposed *CPs* alone.
- ConvLSTM: We use a convLSTM2D model based on grid map alone, without congestion propagation pattern graph and spatial matrix construction. Again we perform the prediction on the granularity of each individual road segment.
- CP-ConvLSTM: We improve ConvLSTM by incorporating the *CPs* alone. Differently from the approaches above, this approach does not perform the prediction independently on each road segment but instead incorporates *CPs*.
- CPM-CNN-LSTM: We improve CNN-LSTM by using congestion propagation pattern graphs which are represented by spatial matrixes.

C. Performance Comparison

We first give the main results of 7 approaches in Table II. CPM-ConvLSTM leads to the lowest MSE and MAE on both

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT MODELS

	weekday		weekend	
Models	MSE	MAE	MSE	MAE
LSTM	0.598624	0.444490	0.317636	0.303783
CNN-LSTM	2.103802	0.993106	1.139783	0.715469
ConvLSTM	1.736250	0.898181	0.977357	0.653928
CP-CNN-LSTM	0.837977	0.544024	0.467619	0.419031
CP-ConvLSTM	0.660635	0.468560	0.337836	0.338780
CPM-CNN-LSTM	0.358728	0.315467	0.166550	0.228535
CPM-ConvLSTM	**0.270992**	**0.187260**	**0.075692**	**0.110878**

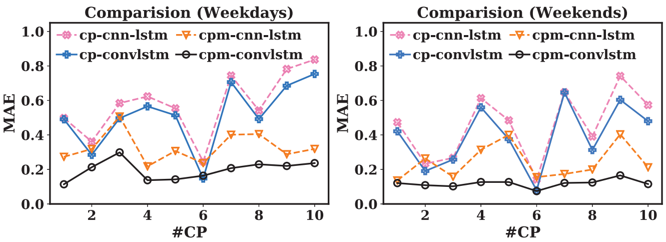

Fig. 6. Performance comparison on partial congestion graphs. Left: weekdays, Right: weekends

workday data and weekend data.

The performance of LSTM is better than CNN-LSTM and ConvLSTM. It is mainly because LSTM can make prediction individually for each road segment and CNN-LSTM/ConvLSTM performs the prediction at the granularity of grid cells by using the average congestion level of a grid cell as the congestion level of roads within the cell. It is not hard to find that the average leads to a higher error than the simple LSTM approach. In addition, we note that ConvLSTM is expected to outperform CNN-LSTM, consistent with the result in ConvLSTM [7].

As the improvement of CNN-LSTM and ConvLSTM, CP-CNN-LSTM and CP-ConvLSTM incorporate *CP* and the used *CP* does contribute to higher errors. Furthermore, by representing the *CP* by SpaMat, CPM-CNN-LSTM and CPM-ConvLSTM can further reduce the errors of CNN-LSTM and ConvLSTM, respectively. Until now, it is not hard to find that SpaMat is rather helpful to improve the performance of CNN-LSTM and ConvLSTM.

Next, we randomly choose 10 identified *CPs* and are interested in how four used approaches perform on each of the *CPs*. As shown in Figure 6, on both weekdays and weekends, CPM-ConvLSTM consistently achieves the smallest error. Instead, the errors of three other approaches significantly fluctuate on various *CPs*.

After that, by randomly selecting a certain road segment, we plot the predicted congestion levels every one hour throughout the day in Figure 7. As shown in this figure, the prediction result of CPM-ConvLSTM is rather close to the actual congestion levels on this road segment.

Finally, by varying the length of time period, we compare MSE and MAE of CPM-ConvLSTM in Table III. By varying the period from 1 to 15 minutes, we find that a longer time interval indicates higher errors. It makes sense: a shorter time

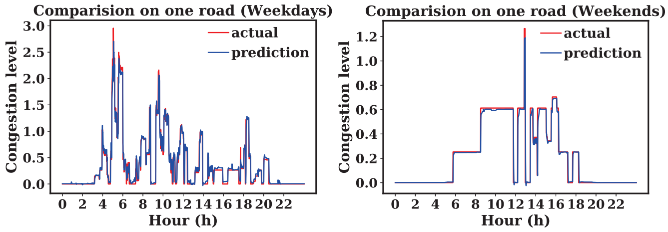

Fig. 7. Comparison of actual and predicted congestion. Left: weekdays, Right: weekends

TABLE III
PERFORMANCE COMPARISON ON DIFFERENT INTERVALS

	Weekday		Weekend	
Time Period (mins)	MSE	MAE	MSE	MAE
1	0.270992	0.187260	0.075692	0.110878
5	0.386714	0.335744	0.156547	0.200715
10	0.564721	0.447017	0.254139	0.280672
15	0.729354	0.533880	0.366197	0.364972

interval means that the more recent and the more accurate congestion levels are fed into CPM-ConvLSTM.

V. RELATED WORK

In the literature, plenty of works have been proposed for traffic predictions. The models used by these works generally fall into two categories: statistical methods and neural network based methods.

Traffic prediction can be considered as a time series prediction problem. For statistical methods, ARIMA, Markov chain, KNN and Support Vector Regression (SVR) are widely used in traffic prediction. ARIMA [1], [2] and Markov chain [8] are both good at time sequence prediction, which consider the temporal auto-correlations of traffic data to enhance prediction performance. But these methods depend on stationary time series data. KNN [9] is used to identify similar traffic patterns for short-term prediction, which can not achieve good performance when historical data has a few similar patterns. Some works [10], [11] used SVR to capture the high dynamics and sensitivity of traffic data to improve the performance of prediction. However, in summary, these methods can only be used for the prediction of individual road segment, and the spatial dependency of adjacent road segments in the road network is not considered.

Recently, deep learning models have been used to handle traffic prediction. Liu [12] made a detailed summary of current deep learning-based approaches for traffic prediction from mobility data. Convolutional Neural Network (CNN) is widely applied to traffic prediction. Meng [13] proposed PCNN for short-term traffic congestion prediction based on a deep convolutional network, which only aims to consider temporal dependencies. Ma [14] used CNN for traffic speed prediction based on 2D time-space matrix. The trajectories are simply and linearly ordered in the space dimension, which could lose spatial information among trajectories. Zhang [3] proposed residual CNN named ST-ResNet based on grid map to extract spatial information for traffic flow prediction. But they ignored the temporal tendency of traffic flows.

LSTM [15] is known to have good performance for handling time series data. Yu [16] applied deep LSTM on time sequence to extract detailed information and predict the traffic flows and accidents, which unfortunately did not consider the spatial information. In addition, LSTM is suitable to make traffic prediction for individual road in a small region. But the traffic condition of a road segment in road network is influenced by both temporal and spatial factors.

Some researchers combined CNN and LSTM to extract spatiotemporal features and make traffic prediction. Wu [17] proposed CLTFP consisting of 1D CNN and LSTM to predict traffic flow for some locations, which only targets small scale road segments. SRCNs [18] and Hetero-ConvLSTM [4] applied on grid map extract spatiotemporal features and make traffic prediction for every grid. But, the size of grid map is hard to set in large scale road network. Other work related to our study is congestion propagation patterns detection, which is important to relive traffic. Nguyen [5] proposed a STC-Tree algorithm based on spatiotemporal information to detect congestion propagation pattern trees and used AprioriSubtree algorithm to select the frequent patterns from the forest. Liang [6] followed the propagation tree and applied a variant of it with indirect influence consideration. But they both did not make traffic prediction.

In summary, while the neural network based methods have achieved good performance in traffic prediction, there are still some problems. Existing models either fail to account for the spatial dependency or can not accurately predict the congestion levels for individual road segments. In our study, we use congestion propagation patterns to overcome the complexity of large scale road network and highlight the propagation relationship among roads. The spatial matrix are applied to form the spatiotemporal traffic congestion data, which provides for the input of ConvLSTM to extract the spatiotemporal information and improve the prediction performance.

VI. Conclusion

In this paper, we propose a deep learning-based congestion prediction model, namely, CPM-ConvLSTM. The model leverages the identified congestion propagation patterns to predict the congestion level at the granularity of road segment. The key contribution of this paper is to use a spatial matrix to incorporate the congestion propagation patterns and spatial adjacency of road segments. Our preliminary experiments on a real-life dataset indicates that the ConvLSTM improved by spatial matrix can greatly outperform the state-of-the-arts in terms of prediction errors.

As the future work, we are interested in finer-grained congestion propagation patterns which can involve both spatial and temporal knowledge, and more meaningful representation of congestion propagation patterns. In addition, we also expect to introduce the external factors, such as weather and surrounding environment point of interests to improve the prediction performance.

VII. Acknowledge

This work is partially supported by National Natural Science Foundation of China (Grant No. 61572365, No. 61772371, No. 61672380), The Fundamental Research Funds for the Central Universities, and Academy of Finland under grant number 317432.

References

[1] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[2] Q. T. Tran, Z. Ma, H. Li, L. Hao, and Q. K. Trinh, "A multiplicative seasonal arima/garch model in evn traffic prediction," *International Journal of Communications, Network and System Sciences*, vol. 8, no. 04, p. 43, 2015.

[3] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[4] Z. Yuan, X. Zhou, and T. Yang, "Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 984–992.

[5] H. Nguyen, W. Liu, and F. Chen, "Discovering congestion propagation patterns in spatio-temporal traffic data," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 169–180, 2017.

[6] Y. Liang, Z. Jiang, and Y. Zheng, "Inferring traffic cascading patterns," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2017, p. 2.

[7] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[8] G. Yu, J. Hu, C. Zhang, L. Zhuang, and J. Song, "Short-term traffic flow forecasting based on markov chain model," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 2003, pp. 208–212.

[9] B. Yu, X. Song, F. Guan, Z. Yang, and B. Yao, "k-nearest neighbor model for multiple-time-step prediction of short-term traffic condition," *Journal of Transportation Engineering*, vol. 142, no. 6, p. 04016018, 2016.

[10] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

[11] W.-C. Hong, "Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, no. 12-13, pp. 2096–2107, 2011.

[12] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Network*, vol. 32, no. 4, pp. 40–46, 2018.

[13] M. Chen, X. Yu, and Y. Liu, "Pcnn: Deep convolutional networks for short-term traffic congestion prediction," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–10, 2018.

[14] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 2017, pp. 777–785.

[17] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," *arXiv preprint arXiv:1612.01022*, 2016.

[18] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, 2017.