Predicting Fine-Grained Traffic Conditions via Spatio-Temporal LSTM

Xiaojuan Wei, Jinglin Li, Quan Yuan, Kaihui Chen, Ao Zhou, and Fangchun Yang

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

Correspondence should be addressed to Jinglin Li; jlli@bupt.edu.cn

Received 27 September 2018; Revised 6 December 2018; Accepted 24 December 2018; Published 14 January 2019

Predicting traffic conditions for road segments is the prelude of working on intelligent transportation. Many existing methods can be used for short-term or long-term traffic prediction, but they focus more on regions than on road segments. The lack of fine-grained traffic predicting approach hinders the development of ITS. Therefore, MapLSTM, a spatio-temporal long short-term memory network preluded by map-matching, is proposed in this paper to predict fine-grained traffic conditions. MapLSTM first obtains the historical and real-time traffic conditions of road segments via map-matching. Then LSTM is used to predict the conditions of the corresponding road segments in the future. Breaking the single-index forecasting, MapLSTM can predict the vehicle speed, traffic volume, and the travel time in different directions of road segments simultaneously. Experiments confirmed MapLSTM can not only achieve prediction for road segments based a large scale of GPS trajectories effectively but also have higher predicting accuracy than GPR and ConvLSTM. Moreover, we demonstrate that MapLSTM can serve various applications in a lightweight way, such as cognizing driving preferences, learning navigation, and inferring traffic emissions.

1. Introduction

Traffic prediction of road segments is a fundamental issue in the Intelligent Transportation Systems (ITS), which can be hopefully used for planning optimal driving routes [1], urban computing [2], balancing traffic control [3, 4], and enhancing driving comfort [5]. It is necessary to explore the traffic dynamics and analyze the evolution pattern of traffic flow. Due to the generation of industrial IoT big data, network infrastructures and computational models have been equipped and applied [6–8]. If the global traffic information is not recognized accurately and timely, ITS will be not successfully deployed or the deployed system will be paralyzed sooner or later.

In general, the power of effectively predicting the future traffic conditions for road segments comes from the historical and real-time traffic information. According to the duration for the future, 3-10 days, 1-3 days, within 1 day, and no more than 15 minutes, traffic flow forecast usually is included long-term, recent-term, short-term and short-time [9]. Most of the existing methods present prediction trend either by using probability and statistics of the time-dependent evolution of current road, or only using the pure spatial relationships among various road segments. Although available spatiotemporal information is combined to model the traffic network pattern, the information does not play out its full potential.

Traffic network possesses complicated spatio-temporal relationship. The prediction methods should have accuracy, robustness, adaptability and portability as the traffic flow is a high-dynamic, high-dimensional, non-linear and non-stationary random process. Traffic conditions of road segments are influenced inevitably by the spatio-temporal information in the traffic network. Deep learning can be used to model high-level abstractions by using multiple non-linear transformations, while the learning network has rarely taken the overall spatio-temporal dynamic pattern into account. It is not convincing to achieve accurate traffic prediction merely by spatial relations between regions or road segments. Hence, the prediction results perform not well at certain times, which occur especially when there are insufficient GPS trajectories through road segments. Based on this, it is proper to consider more supplementary aspects such as map-matching technology used to recognize traffic conditions for road segments accurately and finely.
In this paper, we propose a fine-grained and lightweight approach for traffic predicting of road segments, named MapLSTM, a spatio-temporal long short-term memory network (LSTM [10]) preluded by map-matching [11]. MapLSTM only requires vehicles GPS, and not need to deploy specialized traffic sensors in urban and not use the unobtainable data from ground loop. MapLSTM first obtains the historical and real-time traffic conditions of road segments via map-matching. Then LSTM is utilized to predict the traffic conditions of the corresponding road segments in the future. Breaking the single-index forecasting, MapLSTM can predict multiple traffic conditions for road segments simultaneously. To summarize, the major contributions of this paper consist of the following aspects:

(1) Breaking through the difficulty of obtaining segment-based traffic data, we perform the cognizing of road-grained traffic conditions via map-matching technology.

(2) Based on a large scale of taxi GPS trajectories, we propose MapLSTM to extract features from the high-dynamic, high-dimensional, non-linear and non-stationary traffic flow. And we confirm that MapLSTM have a higher predicting accuracy than GPR [1] and ConvLSTM [12].

(3) We demonstrate MapLSTM can serve to various pragmatic applications: cognizing driving preferences, learning navigation and inferring traffic emissions.

The remainder of this paper is organized as follows: Section 2 reviews the literature on traffic prediction. Section 3 describes the materials and gives details of our mechanism MapLSTM. The next, Section 4 demonstrates the effectiveness and applications. This paper ends in Section 5 with conclusion on our work.

2. Literature Review

Traffic condition prediction can not only be used as the design basis of signal control of ITS but also provide decision support for dynamic route guidance. Whereas, there are still some bottlenecks in short or long term traffic prediction through a lot of real spatio-temporal data.

Spatio-temporal semi-supervised learning model proposed in [13] can infer the volume of each road with real-world data collected from 155 loop detectors and 6918 taxis over 17 days. There are totally 19165 road segments in the urban area, but only 155 road segments are equipped with loop detectors, which results there in an inherent deviation in acquiring the city-wide traffic volumes. Although the constructed affinity graph can characterize the similarities among roads based similar speed patterns, the factors influencing traffic flow are not only the speed of vehicles, but also the road topology, road structure, the regional characteristics, and so on. Traffic conditions of each road segment cannot be predicted accurately only by the spatial relations on the macro.

A vehicle speed is influenced by many factors: the vehicle type, the traffic conditions and the driver’s behaviour. A data driven model is proposed in [14] for vehicle speed prediction where the average traffic speed is estimated based on historical traffic data at first and then the statistical relationship with individual vehicle speed is presented by hidden markov models. Finally, the individual vehicle speed is predicted by forward-backward algorithm. Another mechanism proposed in [15] is a cooperative method which combines with fuzzy markov model and auto-regressive model. These machine learning approaches for vehicle speed prediction do not care about the basic data sources, and focus more on the accuracy of prediction algorithms rather than the accuracy of segment-based traffic prediction.

DeepSense [16] is a typical deep learning approach for traffic prediction with Taxi GPS traces. DeepSense gains the prediction results based on sufficient dataset by using Restricted Boltzmann Machine. Due to the night data is too sparse, so DeepSense made a prediction based filling according to the data of the same time in history. But this prediction-based prediction approach may lose credibility in deep learning. In addition, DeepSense extract and classify the speed only on 0 ~ 60km/h to reflect the traffic congestion or smooth, which lacks universality in some other regions.

Understanding traffic density from large-scale images is another way to recognize the traffic status. Reference [17] as a related work selects a region of interest in a video stream at first, then counts the number of vehicles in the region for each frame, so the density is calculated by dividing that number by the region length. Reference [18] is another image-based learning to measure traffic density using a deep convolutional neural network. These vision-based cognitive methods mainly play a role in local regions, which can dedicate to the operational control but cannot make an efficient decision in tactical planning with in the long run.

In addition, the predicted object is univocal in the existing methods, more is traffic volume or speed, which can merely infer the traffic state of the road segment is congestion, slow, normal, moderate, and unimpeded. It is necessary to explore fine-grained and accurate perception in a simple way.

3. Materials and MapLSTM

In this section, we first provide materials on GPS trajectory, map-matching, and LSTM. Then we depict MapLSTM designed for traffic prediction.

3.1. Materials

3.1.1. GPS Trajectory. Taxis can be considered as ubiquitous mobile sensors constantly probing a city’s rhythm and pulse. Being inherent characteristic, GPS-based taxies have proven to be an extremely useful data source for uncovering the underlying traffic behaviour. So far, the taxi GPS data have been used for urban computing, detecting hot spots, map reconstruction, finding routes, and so on [2,19].

The GPS records of a large number of taxis in a city are routinely saved to a log file $L$, resulting in a very large data set $L = \{p_{11}, p_{12}, \ldots, p_{i1}, p_{i2}, \ldots, p_{in} \ldots \}$. Fields for each GPS record generally contains TaxiID, Location (Longitude, Latitude), Speed, Event, GPS_state, Bearing, and 6 commonly used timing units: YYYY-MM-DD HH:MM:SS. Figure 1 shows an example of GPS log and trajectories. A trajectory $T$ is a time series of GPS points with the time interval between any consecutive GPS points not exceeding...
a certain threshold $\Delta t$ (usually $\Delta t \geq 1 \text{ min}$), i.e., $T_i : p_{i1} \rightarrow p_{i2} \rightarrow p_{i3} \rightarrow \cdots \rightarrow p_{in}$.

3.1.2. Map-Matching. Map-matching is the process of aligning a sequence of observed GPS positions with the road network on a digital map [20]. As a preprocessing step of MapLSTM, map-matching can effectively improve the existing huge amount of low-sampling-rate GPS trajectories in data set.

As shown in Figure 2, map-matching can be performed with the same or different time interval as the GPS points. The GPS points without map-matching can only be mapped to the road network. Not all GPS points can be mapped to their corresponding segments due to the GPS positioning error. But after map-matching, all GPS points can be corrected to the corresponding road segments.

3.1.3. LSTM. LSTM [10] is a time recurrent neural network, which is the most widely used method to process and predict events with relatively long intervals in time series. LSTM can learn about long-term reliant information by input gate $I$, output gate $O$, and forget gate $F$, where, $I$ determines how much of the network input at the current time $x_t$ is saved to the cell state $c_t$. $O$ determines how much of the control unit state $c_t$ is output to the current output value $h_t$ of LSTM. $F$ determines how much of the cell state from the previous time $c_{t-1}$ remains to the current time $c_t$. In short, the input $X$ at different time determines the cell state $C$ at the corresponding time and the current cell state $c_t$ will be affected by the previous cell $c_{t-1}$.

The calculation of each element of LSTM is shown in Algorithm 1. At the current time $t$, $f_t$ denotes forget gate, $i_t$ represents input gate obtained by the previous output $h_{t-1}$ and the current input $x_t$, $C_t$ denotes the cell state and $\tilde{C}_t$ denotes the cell state at the previous time, $o_t$ denotes the output gate, and $h_t$ denotes the cell output. LSTM can not only save information long ago under the control of $F$ but also avoid the current irrelevant content into memory based the gate $I$.

3.2. MapLSTM. MapLSTM is fine-grained and lightweight way. It only requires sampled GPS points of vehicles and not need to deploy expensive traffic sensors in urban and not use the unobtainable data from ground loop. In this section, we describe MapLSTM in detail.
3.2.1. Framework. Figure 3 shows the framework of MapLSTM, which consists of three processes: map-matching, data processing and LSTM predicting.

(a) Map-Matching. A large number of sampled GPS points stored in GPS log need to be matched to road segments. In order to facilitate the operation, it is necessary to manually redivide road segments based on the road network before matching. Generally, the division is based on the intersection, or no redivision, just based on the inherent segments structure in road network, if the calculation resources and road segments information are sufficient and detailed. We do our best to maintain the original topography relationship between the divided road segments. After map-matching, all GPS points can be shifted to the corresponding road segments.

(b) Data Processing. The road segments experienced map-matching also mean the information has been extended, where the road segments and vehicle information are paired off according to their ID and location. Therefore, we can have information statistics including vehicle speed, traverse time, and traffic volume based road segments are input to LSTM concurrently for predicting task. The hidden layers of LSTM can control the long-term or short-term impact on the current state. After output layer of LSTM, it goes through a full connected network with three layers, in which the purpose is to better explore the implied relationships between states.

MapLSTM enables cognition of road segment-based traffic conditions in a lightweight way. For the real-time cognition of global situations, MapLSTM is still valid by collaboration computing where a groups of cells work together to accomplish a relatively large task. Edge computing after cloud computing is a typical collaborative computing environment and has been widely used [21, 22].

3.2.2. Map-Matching Algorithm. Before map-matching, it is necessary to have a information understanding about roads and vehicles. Table 1 describes an example with a sample of the information. All the information about roads and vehicles can be correlated based the auxiliary information (ID, longitude and latitude).

ST-Matching [20] is a pathway with candidate computation and spatio-temporal analysis for low-sampling-rate GPS trajectories. We follow ST-Matching analysis architecture and make a map-matching work on a real digital map in Beijing. As described in Algorithm 2, for the available
3.2.3. Training Data Generating. The raw trajectory data cannot be used directly for our predicting task. It is necessary to match and statistics at first. If we want to get the traffic status prediction of road segments, we need to make a segment-based statistics about the traverse time in different directions, the vehicle speed and the traffic volume.

The data of traverse time in different directions, the average vehicle speed and traffic volume of road segments can be generated by Algorithm 3. When map-matching is done, more fine-grained data can also be obtained such as the average speed and traffic volume under different directions of road segments. The data after map-matching and statistics can be used, which also mean that the training data and the testing data of prediction network are generated.

4. Experiment

We compare the following experiments to verify the performance of MapLSTM.

(1) Gaussian Process Regression (GPR) [1]. It is one of the most popular used prediction algorithms and often used to compare performance as a baseline.

(2) ConvLSTM [12]. It extends LSTM to have convolutional structures in both the input-to-state and state-to-state transitions, and captures spatiotemporal correlations better.

(3) ConvLSTM+. It is ConvLSTM increased epoch numbers.

Table 1: An example with a sample of the main information about road and vehicle.

<table>
<thead>
<tr>
<th>Name</th>
<th>The Main Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road</td>
<td>ID</td>
</tr>
<tr>
<td></td>
<td>59565200918</td>
</tr>
<tr>
<td>Vehicle</td>
<td>ID</td>
</tr>
<tr>
<td></td>
<td>6409</td>
</tr>
</tbody>
</table>

Input: Beijing Road network R, Coordinate axis A, Trajectories T, where T = [t_1, t_2, t_3, ..., t_n], 

Output: The one-to-one results of road segments and vehicles information: M_T'.

(1) Initialize CPSet = 0;
(2) Repeat i = 1, 2, 3, ..., n;
(3) For j = 1 to n;
(4) S_j = GetCandidatePoints(p_j, R);
(5) CPSet.add(S_j);
(6) End for;
(7) k = 1;
(8) While k ≤ |CPSet| do
(9) V_j = GetSpaVal(R, A, CPSet(k), dist(p, CPSet(k)));
(10) V_j = GetTempVal(R, A, Speed, time(p, p_{i+1}));
(11) M_j = MatchSeq(V_j, V_j);
(12) k = k + 1;
(13) End while;
(14) Visualized M_j → M_T';
(15) return M_T'.

Algorithm 2: Map-matching algorithm.
4.1. Datasets. A large scale of real taxi trajectory data are used in our predicting task. The data package of GPS log includes over 400,000 taxicabs’ trajectories in November 2012, Beijing. And full-scale entries are contained during 24 hours for each day. We use data between 8:00 ~ 20:00 in weekdays as the traffic pattern can be learned better in the daytime. We can get dataset \( 22 \times 13 \times 30 \) when the time interval is 2 minutes.

There are too many segments in road network \( R \), so we manually redivide the road segments based \( R \) to verify the feasibility of MapLSTM. The road segments after redivision is stored to set \( R_{ss} \). Figure 4 depicts the traffic data of \( R_{ss} \) on November 1, 2012 at 8 o’clock, including the traverse time in different directions \( TT_{WE} \) (from west to east), \( TT_{EW} \) (from east to west), \( TT_{SN} \) (from south to north), and \( TT_{NS} \) (from north to south), the average vehicle speed, and the traffic volume.

4.2. Training. In MapLSTM, the obtained dataset is divided into training set and test set in an 8 : 2 ratio. The prediction model has \( \text{batch size} = 20 \), \( \text{lr decay} = 0.93 \), \( \text{hidden size} = 250 \), and \( \text{num steps} = 6 \) (the size of window, that means using data from the previous 6 time units to predict the next one). The sizes of the three full connection layers are 180 x 150, 250 x 180, and 250 x 250, which is related to the total number of road segments.

ConvLSTM has the same dataset as MapLSTM, and the model has \( \text{input} = 21 \times 21 \), \( \text{batch size} = 8 \), \( \text{num steps} = 6 \), \( \text{kernel} = 5 \times 5 \), \( \text{filters} = 10 \), and \( \text{max epoch} = 70 \). ConvLSTM+ is iterated 20 times more than ConvLSTM.

4.3. Performance Evaluation. Mean absolute error (MAE) is the most commonly used criteria in predictive algorithms and is employed to evaluate the proposed MapLSTM.

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |(f_i - y_i)|
\]

where \( f_i \) is the predicted value and \( y_i \) is the observed value. The smaller the MAE, the stronger the predictable ability of algorithms.

As shown in Table 2, whether it is the MAE of vehicle speed, traffic count, or travel time in different direction \( TT_{WE} \) (from west to east), \( TT_{EW} \) (from east to west), \( TT_{SN} \) (from south to north), and \( TT_{NS} \) (from north to south), MapLSTM is smaller than GPR and ConvLSTM. For a certain algorithm, the closer the value of “Train” and “Test” of each parameter is, the more robust it is. The results of ConvLSTM are similar to MapLSTM but do not exceed MapLSTM. That is because ConvLSTM with the ability to capture spatiotemporal correlations is good at predicting relatively single spatial pattern, but the spatial patterns of road traffic are complex. In the future, we will focus on complex spatial correlations in traffic environment. Compared to ConvLSTM, some parameters of ConvLSTM+ are slightly better because ConvLSTM+ increased the number of epoch.

It is important to note that MAE is affected by the accuracy of the raw data and it will decline if the dataset is large enough.

4.4. Applications

4.4.1. Cognizing Driving Preference. Different drivers have different preferences about different types of roads, and they also have different impulse to reroute roads due to their different tolerance about the cost expectations of current congestion. For example, the drivers with low tolerance may choose a highway bypass which have a lower congestion cost expectations but have more traffic lights. Tolerance of drivers changes dynamically with various spatial-temporal conditions such as travel distance, congestion time, and arrival time. Therefore, a large deviation between the traffic optimization results and the actual expectation of drivers will lead to failure of traffic scheduling. Quite a few drivers...
Table 2: MAEs comparison of GPR, ConvLSTM, and MapLSTM.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Speed</th>
<th>Count</th>
<th>TT_WE</th>
<th>TT_EW</th>
<th>TT_SN</th>
<th>TT_NS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR</td>
<td>70.79</td>
<td>46.43</td>
<td>66.81</td>
<td>70.7</td>
<td>63.2</td>
<td>66.08</td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>Train</td>
<td>18.78</td>
<td>7.18</td>
<td>18.75</td>
<td>18.27</td>
<td>17.77</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>19.27</td>
<td>7.32</td>
<td>18.71</td>
<td>18.11</td>
<td>18.14</td>
</tr>
<tr>
<td>ConvLSTM+</td>
<td>Train</td>
<td>19.44</td>
<td>7.13</td>
<td>18.75</td>
<td>18.59</td>
<td>17.82</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>18.91</td>
<td>6.89</td>
<td>18.71</td>
<td>17.96</td>
<td>17.85</td>
</tr>
<tr>
<td>MapLSTM</td>
<td>Train</td>
<td>18.33</td>
<td>5.59</td>
<td>16.42</td>
<td>16.5</td>
<td>16.94</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>18.53</td>
<td>7.05</td>
<td>16.91</td>
<td>17.21</td>
<td>17</td>
</tr>
</tbody>
</table>

Figure 5: Segment-based traffic information at a certain time.

choose a looked like shortest road, only to find the route is congested by many vehicles whose drivers make a similar decision.

The traditional route planning methods are more inclined to train drivers’ basic selection tendency and do not have personalized features. The participants in these methods are considered the rational contenders perfectly. The planned result is the purely rational optimal solution and does not express the noncomplete rational decision-making preference for drivers in the actual routing decisions. Although the questionnaire may be a handy pathway for cognizing driving preferences, it lacks efficiency and comprehensiveness.

The premise of learning driving preferences is to obtain an understanding about the roads conditions. The more we aware of road properties, the more satisfied we cognise the personalized preferences. MapLSTM can have a fine-grained cognition of road traffic conditions, so we can learn the driving preferences easily. For drivers of vehicles, there are two preferences getting the most attention: time and distance.

Figure 5 shows the traffic information about the travel time, distance, vehicle density, and speed of each road segments in Rs where the vehicle is driving from place A to place B. In order to compare the preference in driving, the full driving routes based different driving preferences including average speed, vehicle count, distance and travel time are shown in Figure 6.

4.4.2. Learning Navigation. Navigating vehicles to their destination is an important service for ITS. In addition to using historical and real-time traffic conditions, the state-of-the-art systems take into account the impact on the future traffic conditions which can be obtained by predicting. For example, the method in [23] has the ability of learning experience-based autonomous navigation based the global traffic dynamic, and the method in [1] is another dynamic planning scheme based on situation awareness where the city sensors are deployed to maintain an up-to-date view of the city’s current traffic state.
As mentioned above, the existing methods are still laborious for lightweight, fine-grained, and accurate prediction. So we propose MapLSTM to predict traffic conditions effectively. We analyze and compare the use about the predicted traffic conditions in navigation planning, as in Table 3, the lower the computing complexity, the lighter the planning algorithm; the higher the navigation accuracy, the better the navigation performance; perdurability represents the sustainability of a transportation system; the higher the perdurability, the more sustainable the transportation system.

**4.4.3. Inferring Traffic Emissions.** In the COPERT model [28], hot emissions are one of the key essentials about traffic emissions. Hot emissions occur when the engine of vehicle is at its normal mode. Hot emission factor $EF$, the amount of pollutant a single vehicle emits per kilometer (g/km), is calculated as a function of travel speed $v(km/h)$ [29].

$$EF = \frac{(a + cv + ev^2)}{(1 + bv + dv^2)}$$  \hspace{1cm} (2)

where, $a$, $b$, $c$, $d$, $e$ are the pollution emission parameters of COPERT model, these values are given in [29] to caluminate different kinds of emissions and gas consumption: CO, Hydrocarbon, Nox, Fuel Consumption (FC).

As for other pollutants like CO$_2$ and PM$_{2.5}$, their emission factors are proportional to FC.

$$EF_{CO_2} = 3.18 * EF_{FC}$$

$$EF_{PM_{2.5}} = 3 * 10^{-5} * EF_{FC}$$  \hspace{1cm} (3)

**4.4.4. Other Applications.** Table 4 compares the applications about traffic prediction in recent two years. It can be seen from Table 4 that the traffic prediction methods is more inclined to use machine learning and deep learning algorithm to achieve more accurate and larger regional prediction; the advance cannot be separated from the rapid development of machine learning and deep learning in recent years.

**5. Conclusions**

Urban road traffic system is the lifeblood of a city, which ensures its operation. Predicting traffic conditions for road segments is the prelude of working on intelligent transportation. In this paper, we proposed MapLSTM, a traffic predicting mechanism for road segments, to promote the development of ITS. MapLSTM can accelerate the landing of many applications in a lightweight and fine-grained way. In the future, autonomous humanlike driving based on road topography is worth concern, and we will focus on complex spatial correlations in traffic environment.
Table 3: Applications and comparisons about the predicted traffic conditions in navigation planning.

<table>
<thead>
<tr>
<th>Literature</th>
<th>Raw data source</th>
<th>Object-based</th>
<th>Pathway</th>
<th>Core algorithm</th>
<th>Complexity</th>
<th>Accuracy</th>
<th>Save time</th>
<th>Perdurability</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>smart sensors</td>
<td>city</td>
<td>self-aware</td>
<td>Gaussian Process Regression</td>
<td>middle</td>
<td>middle</td>
<td>middle</td>
<td>low</td>
</tr>
<tr>
<td>[23]</td>
<td>GPS points</td>
<td>region</td>
<td>autonomous</td>
<td>Value Iteration Network</td>
<td>middle</td>
<td>middle</td>
<td>high</td>
<td>middle</td>
</tr>
<tr>
<td>[25]</td>
<td>GPS points</td>
<td>region</td>
<td>agents</td>
<td>Ant Colony+RL</td>
<td>middle</td>
<td>middle</td>
<td>middle</td>
<td>middle</td>
</tr>
<tr>
<td>[26]</td>
<td>vehicles sharing</td>
<td>city</td>
<td>RIS</td>
<td>statistics</td>
<td>low</td>
<td>low</td>
<td>middle</td>
<td>low</td>
</tr>
<tr>
<td>Year</td>
<td>Literature</td>
<td>Basic data source</td>
<td>Target</td>
<td>Term</td>
<td>Core algorithm</td>
<td>Complexity</td>
<td>Granularity</td>
<td>Object-based</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------</td>
<td>-----------------</td>
<td>----------</td>
<td>--------------------------------</td>
<td>------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>2018</td>
<td>[18]</td>
<td>web camera</td>
<td>traffic density</td>
<td>short</td>
<td>Convolutional neural network</td>
<td>high</td>
<td>fine-grained</td>
<td>intersection</td>
</tr>
<tr>
<td></td>
<td>[27]</td>
<td>an open dataset</td>
<td>traffic flow</td>
<td>long/short</td>
<td>Generative adversarial network</td>
<td>high</td>
<td>coarse-grained</td>
<td>freeway</td>
</tr>
<tr>
<td></td>
<td>[17]</td>
<td>web camera</td>
<td>traffic density</td>
<td>short</td>
<td>Fully convolutional networks</td>
<td>high</td>
<td>fine-grained</td>
<td>restricted area</td>
</tr>
<tr>
<td>2017</td>
<td>[15]</td>
<td>an experimental car</td>
<td>vehicle speed</td>
<td>short</td>
<td>Auto-regressive model</td>
<td>middle</td>
<td>fine-grained</td>
<td>road segment</td>
</tr>
<tr>
<td></td>
<td>[14]</td>
<td>floating car</td>
<td>vehicle speed</td>
<td>short</td>
<td>HMMs+SUMO</td>
<td>middle</td>
<td>coarse-grained</td>
<td>motorway</td>
</tr>
<tr>
<td></td>
<td>[13]</td>
<td>Loop Detector</td>
<td>traffic volume</td>
<td>short</td>
<td>ST semi-supervised learning</td>
<td>low</td>
<td>fine-grained</td>
<td>road segment</td>
</tr>
<tr>
<td></td>
<td>[1]</td>
<td>traffic loops</td>
<td>traffic flow</td>
<td>long</td>
<td>Gaussian process regression</td>
<td>low</td>
<td>coarse-grained</td>
<td>region</td>
</tr>
</tbody>
</table>
Data Availability

We used the source code of ConvLSTM in our paper; the URL is: “https://github.com/carlihome/tensorflow-convlstm-cell.” Moreover, we used the dataset “T-Drive Taxi Trajectories” released by MSRA; the URL is “https://www.microsoft.com/en-us/research/project/urban-computing.” There is just one week of data in released dataset. Although one week of data can also conduct secondary analyses, we used one month of data of “T-Drive Taxi Trajectories” in our experiments for better performance, in which data was from the previous cooperation project.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Natural Science Foundation of Beijing under Grant no. 4181002, and the Natural Science Foundation of China under Grant no. 61876023.

References


