Dest-ResNet: A Deep Spatiotemporal Residual Network for Hotspot Traffic Speed Prediction*

Binbing Liao[1], Jingqing Zhang[2], Ming Cai[1], Siliang Tang[1], Yifan Gao[1], Chao Wu[1], Shengwen Yang[3], Wenwu Zhu[4], Yike Guo[2], Fei Wu[1]
[1] Zhejiang University, [2] Imperial College London, [3] Baidu Inc., [4] Tsinghua University
{bbliao,cm,siliang,yfgao0826,chao.wu,wufei}@zju.edu.cn;yangshengwen@baidu.com
{jingqing.zhang15,y.guo}@imperial.ac.uk;wwzhu@tsinghua.edu.cn

ABSTRACT

With the ever-increasing urbanization process, the traffic jam has become a common problem in the metropolises around the world, making the traffic speed prediction a crucial and fundamental task. This task is difficult due to the dynamic and intrinsic complexity of the traffic environment in urban cities, yet the emergence of crowd map query data sheds new light on it. In general, a burst of crowd map queries for the same destination in a short duration (called "hotspot") could lead to traffic congestion. For example, queries of the Capital Gym burst on weekend evenings lead to traffic jams around the gym. However, unleashing the power of crowd map queries is challenging due to the innate spatiotemporal characteristics of the crowd queries.

To bridge the gap, this paper firstly discovers hotspots underlying crowd map queries. These discovered hotspots address the spatiotemporal variations. Then Dest-ResNet (Deep spatiotemporal Residual Network) is proposed for hotspot traffic speed prediction. Dest-ResNet is a sequence learning framework that jointly deals with two sequences in different modalities, i.e., the traffic speed sequence and the query sequence. The main idea of Dest-ResNet is to learn to explain and amend the errors caused when the unimodal information is applied individually. In this way, Dest-ResNet addresses the temporal causal correlation between queries and the traffic speed. As a result, Dest-ResNet shows a 30% relative boost over the state-of-the-art methods on real-world datasets from Baidu Map.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; **Data mining**;

KEYWORDS

Traffic speed prediction; Crowd map query; Social media; LSTM

*Ming Cai and Siliang Tang are the corresponding authors. This work was done while Binbing Liao was a research intern at Baidu.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240656

ACM Reference Format:
Binbing Liao, Jingqing Zhang, Ming Cai, Siliang Tang, Yifan Gao, Chao Wu, Shengwen Yang, Wenwu Zhu, Yike Guo, Fei Wu. 2018. Dest-ResNet: A Deep Spatiotemporal Residual Network for Hotspot Traffic Speed Prediction. In *Proceedings of 2018 ACM Multimedia Conference (MM '18)*, Oct. 22-26, 2018, Seoul, Republic of Korea. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240656

1 INTRODUCTION

Accurate and real-time prediction of traffic speed is crucial and fundamental for the successful deployment of intelligent transportation systems (ITS) since traffic speed prediction is particularly useful for many applications such as traffic network planning, route guidance, and congestion avoidance [32]. However, traffic prediction is challenging due to the dynamic and intrinsic complexity of the traffic environment in cities.

A number of traditional methods have been proposed for traffic prediction, such as ARIMA models [2, 27, 29, 30], RF [16] and SVR [14]. Recently, some deep learning approaches have been proposed for traffic prediction, such as stacked autoencoders (SAEs) [21], deep belief network [13] and LSTM [22]. However, the traffic in the urban area is highly dynamic and varies greatly on account of diverse and complicated factors such as crowd activities. Due to the intrinsic complexity and the lack of reliable data sources, previous works ignored the social factors (e.g., festivals, crowd map queries), which can potentially have a great impact on the traffic system.

With the explosive growth of mobile technology, map applications such as Baidu Map and Google Map provide an auxiliary rich source of information for traffic speed prediction. Figure 1 shows the average traffic speed and the number of queries for the Capital Gym, Beijing on April 8, 2017. The query counts at time t is the number of queries for the Capital Gym, and the estimated arrival time of the queries is t. It can be observed that the query counts (in red) are much more than average historical query counts (in blue) at around 18:00, which account for a sudden drop in the traffic speed. Note that a set of queries for the same destination issued at a given time may lead to a traffic jam at their searched destination after a while. Therefore, crowd map queries are able to provide an early warning (46 minutes as shown in Table 1) of traffic jams for many applications in ITS, especially for congestion avoidance.

More interestingly, the burst of crowd map queries at a given time usually indicates a "hotspot" being held at the searched destination (the "hotspot" is "Fish Leong Concert" in Figure 1). Therefore, the discovery of hotspots from a great many crowd map queries gives a way to explain the reasons for traffic jams.

We aim to harness the power of crowd map queries for traffic speed prediction. However, there are three challenges for the

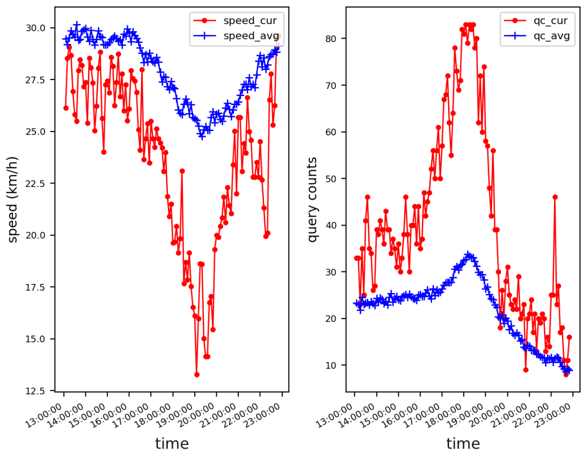


Figure 1: The traffic speed (left) and crowd query counts (right) around the Capital Gym, Beijing on April 8, 2017. The red "dot" denotes the current traffic speed (query counts) while the blue "plus" represents the average historical traffic speed (query counts). At 19:00, there is the Fish Leong Concert in the Capital Gym.

organic integration of the queries and the road traffic speed: 1) **Spatiotemporal variation.** The crowd map queries for the same destination (e.g., the Capital Gym) can be issued at different source locations and at different times by individual users; 2) **Spatial impact.** A set of queries for the same destination from different source locations have different impacts on the traffic speed of road segments towards the destination due to diverse routes from their preferred directions; 3) **Temporal causal correlation.** For example, a set of queries for the Capital Gym issued before 15:00 imply the information of how many users would arrive at the Capital Gym after 15:00 (i.e., the temporal causal correlation) since users usually inquire the traffic before heading towards their destinations.

Motivated by the idea that performance gain can be achieved by appropriately integrating multi-modal information from different sources, this paper attempts to predict future traffic speed via the organic integration of the current road traffic speed and crowd map queries. The intuition of this paper is that a set of explosive map queries searching for the same destination at a given time generally foresee a traffic jam after a while, and therefore the appropriate integration of crowd map queries and traffic speed can boost the performance of future traffic speed prediction. Technically, the contributions of this paper can be summarized in three aspects.

- This paper proposes Dest-ResNet for hotspot traffic speed prediction. Dest-ResNet is a sequence learning framework that jointly deals with two sequences in different modalities, i.e., the traffic speed sequence and the query sequence. The motivation behind Dest-ResNet attempts to learn a residual network to amend the errors caused when the unimodal information is learned individually. As a result, Dest-ResNet addresses the temporal causal correlation between queries and the traffic speed. Furthermore, the generality of Dest-ResNet makes it promising for many other multi-modal sequence learning applications, e.g. text and speech.
- In order to address the spatiotemporal variation, this paper proposes a hotspot discovery method from crowd map queries. In this method, a grid-based map segmentation is first utilized to group queries issued at similar geographical locations, then users' arrival time is estimated w.r.t their searched destinations mentioned in queries and an arrival time tensor is constructed. Finally, the discovery of the hotspots depends on the arrival time tensor.
- The queries for the same destination come from different source locations have different impacts on the traffic speed of road segments towards the destination. Therefore, this paper proposes a query modeling algorithm to encode the intricate spatial impacts between road segments and queries.

The rest of this paper is organized as follows. Sec. 2 presents the related works of traffic prediction. In Sec. 3, we introduce the problem definition and an overview of Dest-ResNet. Sec. 4 presents the discovery of hotspots and the query modeling. In Sec. 5, we describe the proposed Dest-RestNet in detail. The Sec. 6 presents qualitative and quantitative results of hotspot discovery and traffic speed prediction. Finally, we conclude this paper in Sec. 7.

2 RELATED WORK

2.1 Traffic Prediction

As a critical component in ITS, the traffic (speed, flow, and density) prediction has been explored by considerable research, in which both parametric and non-parametric methods are popular. On the one hand, autoregressive integrated moving average (ARIMA) [2], which assumes the stationary of traffic speed sequence, has been widely applied as a parametric technique for traffic prediction. The ARIMA(0, 1, 1) model is found the most statistically significant for all forecasting [17]. However, the inefficiency of the ARIMA model restricts its suitability for the large-scale real-time application.

On the other hand, due to the nonlinearity and dynamics of traffic, the non-parametric methods such as k-NN [4, 8, 31], RF [16], SVR [14], OL-SVR [3], Bayesian network [26] and neural networks [12, 23, 24, 28, 33] have been applied in a handful of studies. Nevertheless, due to the shallow architectures, these models fall behind the recently deep learning approaches for traffic prediction, such as stacked autoencoders (SAEs) [21] and deep belief network [13]. However, both of the deep learning models are tested on traffic data collected from highways, where the traffic condition are relatively stable. In contrast, our work is more general and robust for urban traffic which has higher variance due to diverse and complicated factors such as crowd activities. [22] simply use long short-term memory network (LSTM) to predict traffic speed in an urban area. Nonetheless, these methods mentioned above are naive without considering the social factors (e.g., festivals, crowd map queries) which potentially have a great impact on the traffic. The traffic data we use from Baidu Map covers a large urban area and provides rich online information to support more accurate traffic forecasting.

2.2 Traffic Prediction with Multi-modal Data

A few researchers have attempted to predict the traffic speed with related multi-modal data. [10, 20] propose an optimization framework to extract traffic speed indicators based on location-based social media and incorporate them into traffic speed prediction via linear regression. However, due to the nonlinearity of traffic, linear regression for traffic prediction is insufficient. In addition, [18, 19] use the concatenation and content attention to integrate the crowd map queries for traffic prediction in an encoder-decoder way. However, simply concatenation and attention are not enough. Dest-ResNet is a sequence learning framework that attempts to learn a residual network to amend the errors caused when the unimodal information is learned individually. As a result, Dest-ResNet performs better than [18, 19].

3 PRELIMINARIES

3.1 Problem Definition

Let $\mathcal{L} = \{l^i | i = 1, 2, \ldots, K\}$ be a collection of K road segments in a given area. A road segment is a uniform section of a road. The traffic speed of the road segment $l \in \mathcal{L}$ in a duration is $\boldsymbol{v}^l = (v_1^l, v_2^l, \ldots, v_t^l)$, where v_t^l is a scalar that represents the traffic speed of the road segment l at time t.

Let $Q = \{q^i | i = 1, 2, \ldots, N\}$ be a corpus of users' map query records. Each record q^i is defined by a tuple $q^i = (t_s^i, s^i, d^i)$, where: (1) t_s^i is the starting time of the map query q^i; (2) s^i is the source location of q^i; (3) d^i is the destination. For simplicity, the superscript is removed without confusion in the rest of this paper.

Specifically, for the road segment l, our goal is to maximize the conditional probability of observing the future traffic speed slot $V^f = (v_{t+1}^l, v_{t+2}^l, \ldots, v_{t+w}^l)$, given the previous traffic speed slot $V^p = (v_1^l, v_2^l, \ldots, v_t^l)$ and the map query records Q:

$$p_\theta(V^f | V^p, Q) = \prod_{m=1}^{w} p_\theta(v_{t+m} | v_1, v_2, \ldots, v_{t+m-1}, Q_{<=t}) \quad (1)$$

where $Q_{<=t} = \{q^i | t_s^i <= t\}$, w is the prediction horizon and θ is the model parameter. Given K road segments, our training objective is to maximize the following log likelihood w.r.t. the model parameter θ:

$$\arg\min_\theta \; -\frac{1}{K} \sum_{k=1}^{K} \log p_\theta(V^f | V^p, Q). \quad (2)$$

3.2 Overview of Dest-ResNet

Before presenting Dest-ResNet, we clarify three unique challenges of modeling crowd map queries for traffic speed prediction, which motivate the design of Dest-ResNet:

1) **Spatiotemporal variation**. The raw map query data exhibits considerable spatiotemporal variations. For example, a set of users' query records for the same destination (e.g., the Capital Gym) can be issued at different source locations and at different times by different users. The spatiotemporal characteristics are beneficial to traffic speed prediction since a burst of queries for the same destination in a short duration in general leads to a traffic jam.

2) **Spatial impact.** A set of queries for the same destination from different source locations have different impacts on the traffic speed of road segments towards the destination. For examples, the more users issued queries for a destination with similar routes, and the heavier traffic jam can be caused across the corresponding road segments towards the destination. On the contrary, other limited queries for the destination will not cause heavy traffic across related road segments since they will take different routes respectively. How to effectively and efficiently model the spatial impact between road segments and queries is challenging.

3) **Temporal causal correlation.** The map queries have innately foreseeable characteristics for traffic speed prediction. For example, a set of queries for the Capital Gym issued before 15:00 can be utilized to foresee the counts of users arriving at the Capital Gym after 15:00 (i.e., the temporal causal correlation) since users usually inquire the traffic before heading towards their destinations. How to effectively capture the temporal causal correlation between traffic speed and queries is very challenging.

From the challenges as mentioned above, we argue that people's activities in an urban city usually burst in specific geographical regions during given time periods (e.g., attending a concert in the Capital Gym at weekend evening), which means that there are latent spatiotemporal hotspots that lead to the bursts. Therefore, how to identify a spatiotemporal hotspot (e.g., what kinds of activities are going to trigger a traffic jam in a given time duration) is very important. In Sec. 4.1, we design a hotspot discovery method to detect the spatiotemporal hotspots and address the spatiotemporal variation. The detected hotspots are used as basic units in the later traffic speed prediction.

A set of queries for a particular destination from different source locations will lead to complicated interactions among the road segments towards the destination. To capture the intricate interactions among road segments from source locations to the destinations, a query modeling method is devised to encode the spatial impacts of road segments mentioned in queries at a given time in Sec. 4.2.

After discovering the spatiotemporal hotspots and modeling the hotspot queries, we propose Dest-ResNet to effectively capture the temporal causal correlations between traffic speed and queries in Sec. 5. Dest-ResNet is a sequence learning framework that jointly deals with two sequences in different modalities, i.e., the traffic speed sequence and the query sequence. The main idea of Dest-ResNet is to learn to explain and amend the errors in the unimodal learning with the fusion of two modalities using a residual network.

4 HOTSPOT QUERY MODELING

The sudden explosion of people's activities in a particular geographical region of an urban area during a given period, namely hotspot, can be reflected by the burst of corresponding map queries issued within a short period. In this section, we first introduce the discovery of spatiotemporal hotspots and then the modeling of the spatial interactions between queries and road segments.

4.1 Hotspot Discovery

As shown in Figure 2, the discovery of hotspots consists of following steps:

1) **Grid-based map segmentation.** The map is first partitioned into a $X \times Y$ grid map. The width and height of a grid are both about 1 kilometer. For each query $q = (t_s, s, d)$, coordinates of the source location (x_s, y_s) and the destination (x_d, y_d) can be calculated based on the grid map.

2) **Arrival time estimation.** After one user issues a query for a destination, the user's arrival time t_d is estimated w.r.t the queried destination according to the query mode with a speed of 30km/h (by

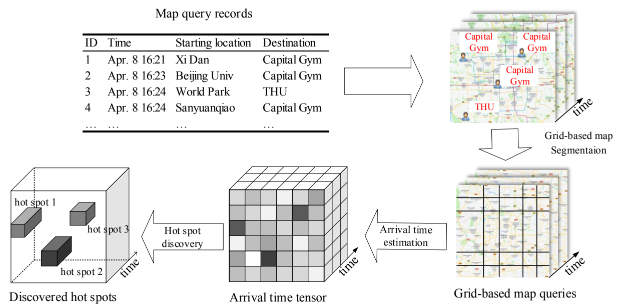


Figure 2: The flowchart of the discovery of hotspots. A set of map queries are segmented into geographical grids. Then we estimate the arrival time when one user triggers a query for a destination and construct the arrival time tensor. A hotspot discovery algorithm is utilized to discover the hotspots from the arrival time tensor.

car), 20km/h (by bus), 10km/h (by bike) or 3.6km/h (by walk). As a result, a query q can be represented as $q = (t_s, t_d, s, d, x_s, y_s, x_d, y_d)$.

Note that some users may not go to the destination that they have searched. However, it is more likely that they will go to their searched destinations if the queries are related to a public hotspot, which is mostly event-driven and indicates the strong intention of traveling from a large group of people. Otherwise, without the intention, the burst of search queries in the map app will not happen. For example, there are many queries issued for the Capital Gym as a destination and the estimated arrival time is around 18:00 on April 8, 2017, in our collected map query dataset. Since the number of such queries is much more than average historical data, we claim that such queries are "hotspot" queries (in this case the hotspot is "Fish Leong Concert" indeed).

3) **Arrival time tensor construction.** Given all queries $C = \{q^i | i = 1, 2, \dots, N\}$, the arrival time tensor is constructed $\boldsymbol{D} = \{d_{x,y,t}\}$, where $x = 1, 2, \dots, X$, $y = 1, 2, \dots, Y$, $t = 1, 2, \dots, T$ and T is the total timestamps. $d_{x,y,t}$ is defined as

$$d_{x,y,t} = |\{q^k | x_d^k = x, y_d^k = y, t_d^k = t\}| \tag{3}$$

where $|\cdot|$ denotes the cardinality of a set.

4) **Hotspot discovery.** Before we discuss how to identify the "hotspot", we firstly introduce the definition of "moment".

Definition 4.1 (Moment). A tuple $m = (x, y, t)$ is a moment if

$$d_{x,y,t-\Delta t} > 0 \tag{4}$$

$$d_{x,y,t} - d_{x,y,t-\Delta t} > \zeta \tag{5}$$

$$\frac{d_{x,y,t} - d_{x,y,t-\Delta t}}{d_{x,y,t-\Delta t}} > \eta \tag{6}$$

Where Δt is time duration (i.e., a week), ζ and η are hyper-parameters that control the significance of the moment. .

Definition 4.2 (hotspot). A tuple $h = (x, y, t_s, t_d)$ is a hotspot if

$$t_d - t_s > \epsilon \tag{7}$$

$$\forall t \in [t_s, t_d], \quad m = (x, y, t) \in \mathcal{M} \tag{8}$$

$$m = (x, y, t_s - 1) \notin \mathcal{M} \wedge m = (x, y, t_d + 1) \notin \mathcal{M} \tag{9}$$

Where $\mathcal{M}$ denotes all the moments and ϵ is a hyper-parameter that controls the length of each hotspot.

From the definitions of moment and hotspot query, the hotspot is actually a local maximum interval of arrival time tensor $\boldsymbol{D}$. Along the time axis t of $\boldsymbol{D}$, for each tuple (x, y), we can find out all the moments via Def. 4.1. For all the moments, the moments are then merged into hotspots using Def. 4.2.

4.2 Hotspot Query Spatial Modeling

In this section, we introduce how to encode the impacts of the hotspot queries. The modeling of the query depends on the query counts and the spatial geographical region that the query will influence. The query counts feature $\boldsymbol{\Phi}_c$ and query spatial feature $\boldsymbol{\Phi}_s$ are concatenated as the feature of query $\boldsymbol{\Phi}$.

Given a hotspot w.r.t a destination d, note that the queries for d come from different places, thus each query has different impacts on different road segments towards the destination d. Given all the queries $Q = \{q^i | i = 1, 2, \dots, N\}$ and road segments $\mathcal{L} = \{l^i | i = 1, 2, \dots, K\}$, where K is the total number of road segments. The query counts feature $\boldsymbol{\Phi}_c(l, t)$ and the query spatial feature $\boldsymbol{\Phi}_s(l, t)$ on the road segment l at time t can be calculated by Algorithm 1.

In Algorithm 1, we argue that the query only influences a small range around the destination at the arrival time. Therefore, for each query, only the road segments within 1km (line 8, 9) towards the destinations are taken into consideration. The function *dist* in line 12 calculates the Euclidean distance of a point and a segment. The function f in line 13 is a decreasing function whose input is the distance d_l and output is the impact $f(d_l)$ that the query q imposes on the road segment l at time t_d. For simplicity, the exponential function $f(x) = exp(-\frac{x}{\sigma})$ is applied, where σ is the impact factor.

Algorithm 1: QueryModeling Calculate the query feature

Input: A set of queries $Q = \{q^i | i = 1, 2, \dots, N\}$, a set of road segments $\mathcal{L} = \{l^i | i = 1, 2, \dots, K\}$

Output: The query counts feature $\boldsymbol{\Phi}_c$, the query spatial feature $\boldsymbol{\Phi}_s$

```
1  Initialization: Φc ← 0, Φs ← 0
2  for each q ∈ Q do
3      // q = (ts, td, s, d, xs, ys, xd, yd)
4      // return the longitude and latitude of a location
5      lonlat_s ← lonlat(s)
6      lonlat_d ← lonlat(d)
7      seg ← segment(lonlat_s, lonlat_d)
8      // return the set of road segments L within 1km
9      L ← near_road_segment(lonlat_d)
10     for each l ∈ L do
11         lonlat_l ← lonlat(l)
12         d_l ← dist(lonlat_l, seg)
13         Φs(l, td) ← Φs(l, td) + f(d_l)
14         Φc(l, td) ← Φc(l, td) + 1
15 return Φc, Φs
```

5 DEST-RESNET

In this section, we firstly introduce the Seq2Seq network for traffic speed prediction. Then we propose our Dest-ResNet.

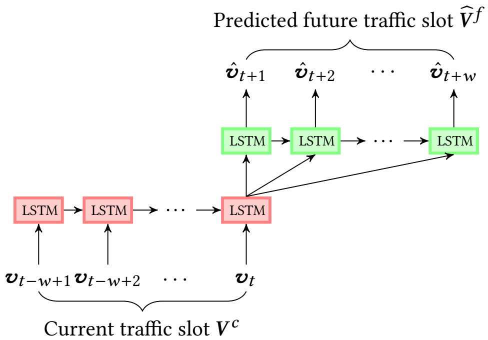


Figure 3: The structure of the Seq2Seq network *Seq2Seq*. The bottom LSTM layer (colored red) encodes information in the current traffic speed slot V^c while the second LSTM layer (colored green) decodes the hidden representation of V^c to predict the future traffic speed slot $\widehat{V}^f$.

5.1 Seq2Seq Network

For each hotspot, given the historical traffic speed of selected k road segments, we aim to forecast their future traffic speed. Specifically, given the current traffic speed slot $\boldsymbol{V}^c = (\boldsymbol{v}_{t-w+1}, \boldsymbol{v}_{t-w+2}, \ldots, \boldsymbol{v}_t)$ of k road segments, we predict their future traffic speed slot $\widehat{V}^f = (\hat{\boldsymbol{v}}_{t+1}, \hat{\boldsymbol{v}}_{t+2}, \ldots, \hat{\boldsymbol{v}}_{t+w})$, where $\boldsymbol{v}_t = (v_t^1, v_t^2, \ldots, v_t^k)^T$ is the traffic speed of k road segments at time t, w is the prediction horizon.

As shown in Figure 3, a sequence to sequence (Seq2Seq) network is applied to model the traffic speed. It consists of two LSTM[11] layers. The bottom LSTM layer (colored red) encodes information in the current traffic speed slot $\boldsymbol{V}^c$ while the second LSTM layer (colored green) decodes the encoding information of $\boldsymbol{V}^c$ to predict the future traffic speed slot $\widehat{V}^f$.

5.2 Dest-ResNet

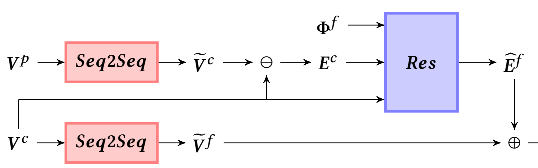


Figure 4: The structure of the proposed Dest-ResNet. The network *Seq2Seq* is used for traffic speed prediction while the network *Res* is utilized to predict the residual.

The map queries issued by users at a certain time can be utilized to foresee the traffic speed around the queried destination after a while. Considering a set of queries triggered earlier than 15:00 whose destinations are all around the Capital Gym, the arrival time towards the destinations can be estimated as described in Sec. 4.1. The number of queries implies the information of how many individuals would arrive at the Capital Gym at the estimated arrival time, which would be beneficial to traffic speed prediction around the Capital Gym. Note that here only the map queries for the Capital Gym issued earlier than 15:00 is utilized to guarantee the foreseeable characteristic of map queries.

Additionally, there would be some residuals between the ground truth traffic speed and the predicted traffic speed when the Seq2Seq network is applied for traffic speed prediction individually. Besides the travel distance, some social factors in our real-world may cause the residuals, such as festivals, drivers' habits and so on, especially for hotspot traffic speed prediction. Therefore, learning the residuals with social factors (e.g., map query) can be utilized to boost the performance of traffic speed prediction.

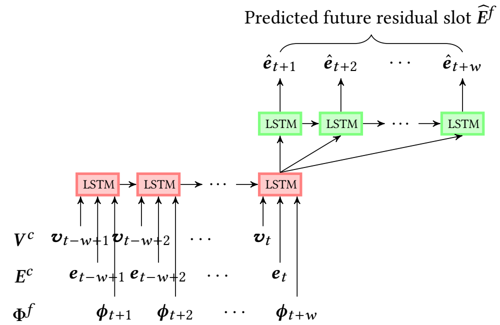


Figure 5: The structure of the residual network *Res*. The bottom LSTM layer(colored red) encodes information in the current traffic speed slot V^c, the current residual slot E^c and the future query slot Φ^f. While the second LSTM layer (colored green) decodes the hidden representation to predict the future residual slot $\widehat{E}^f$.

Figure 4 shows the structure of Dest-ResNet. Given the previous traffic speed slot $\boldsymbol{V}^p = (\boldsymbol{v}_{t-2w+1}, \boldsymbol{v}_{t-2w+2}, \ldots, \boldsymbol{v}_{t-w})$, the current traffic speed slot $\boldsymbol{V}^c$ and the future query slot $\boldsymbol{\Phi}^f = (\boldsymbol{\phi}_{t+1}, \boldsymbol{\phi}_{t+2}, \ldots, \boldsymbol{\phi}_{t+w})$, the future traffic speed slot $\widehat{V}^f = (\hat{\boldsymbol{v}}_{t+1}, \hat{\boldsymbol{v}}_{t+2}, \ldots, \hat{\boldsymbol{v}}_{t+w})$ can be obtained by:

$$\widetilde{\boldsymbol{V}}^c = \boldsymbol{Seq2Seq}(\boldsymbol{V}^p) \tag{10}$$

$$\boldsymbol{E}^c = \boldsymbol{V}^c - \widetilde{\boldsymbol{V}}^c \tag{11}$$

$$\widehat{\boldsymbol{E}}^f = \boldsymbol{Res}(\boldsymbol{V}^c, \boldsymbol{E}^c, \boldsymbol{\Phi}^f) \tag{12}$$

$$\widetilde{\boldsymbol{V}}^f = \boldsymbol{Seq2Seq}(\boldsymbol{V}^c) \tag{13}$$

$$\widehat{\boldsymbol{V}}^f = \widetilde{\boldsymbol{V}}^f + \widehat{\boldsymbol{E}}^f \tag{14}$$

where (1) $\widetilde{\boldsymbol{V}}^c$ is the estimated current traffic speed slot; (2) $\boldsymbol{E}^c$ is the current residual slot; (3) $\widehat{\boldsymbol{E}}^f$ is the predicted residual slot; (4) $\boldsymbol{Res}$ is a residual network shown in Figure 5; (5) $\widetilde{\boldsymbol{V}}^f$ is the estimated future traffic speed slot; (6) $\widehat{\boldsymbol{V}}^f$ is the predicted future traffic speed slot. Note that the future query slot $\boldsymbol{\Phi}^f$ is calculated using Algorithm 1 according to the "future" arrival time of the queries.

To make a long story short, the main parts of Dest-ResNet are the Seq2Seq network ***Seq2Seq*** and residual network ***Res***. The Seq2Seq network is applied to predict the traffic speed given historical traffic speed data only. While the residual network is tasked with learning to predict the errors between ***Seq2Seq*** prediction and ground truth in the next horizon w. The queries, which are long-term foreseeable, are used to explain and predict the residuals. These predicted residuals $\widehat{E}^f$ can then be utilized to amend the future traffic speed slot $\widehat{V}^f = \widetilde{V}^f + \widehat{E}^f$ from the estimated future traffic speed slot $\widetilde{V}^f$.

6 EXPERIMENTS

6.1 Datasets

Table 1: Statistics of the query dataset.

Items	Numbers
Filtered queries	114,658,750
Query words	17,210,732
Average distance/query	12km
Average travel time/query	46 minutes

6.1.1 ***Data Pre-processing***. Our experiments are based on two real-world datasets: 1) traffic speed dataset, which is used for traffic speed prediction. It is collected in Beijing, China from Baidu Map, with a time period from April 1, 2017, to May 31, 2017. This dataset contains about 800 thousands of road segments. The traffic speed of each road segment is recorded per minute. Since the traffic speed data is from real-world urban areas, the traffic speed lights would have a significant impact on the traffic speed, leading to the traffic speed varies greatly. For instance, the traffic speed may differ 20km/h between two consecutive minutes. To make the traffic speed predictable, for each road segment, the traffic speed per 5 minutes is average and zero-phase digital filtering [9] is utilized to smooth the traffic speed; and 2) map query dataset, which is used for hotspot discovery and traffic speed prediction. The spatial and temporal ranges of this dataset are the same as the traffic speed dataset with the statistics in Table 1.

There are two modes of map queries in the Baidu Map, "location search" (searching for a certain place) and "route search" (searching a route from one place to another). Each query records the user ID (anonymized), search timestamp, coordinate of the current location, coordinate and query word of the source location (in "route search"), coordinate and query word of the destination. Note that if the query mode is "location search", the source location is the same as the current location. It is preprocessed as follows:

- To eliminate redundancy, only the last query can be retained if a single user created several queries in 10 minutes.
- It is assumed that the users are more likely to go to their searched destinations if they are currently close to the searched source locations. Thus the queries whose current locations are 2km away from the source locations are eliminated.
- Since the searched source locations are within 2km from current locations, the starting time is estimated according to the distance between source locations and current locations, with a speed of 3.6km/h (by walk).
- To address the time variations, the starting (and arrival) times of the queries are converted to $[0, 17, 568)$ by calculating its offset (every 5 minutes) w.r.t. 12:00 AM, April 1, 2017.

6.1.2 ***Hotspot Discovery***. Let $\Delta t = 2,016$ ($12 \times 24 \times 7$, a week), $\zeta = 100$, $\eta = 0.2$, $\epsilon = 12$(an hour). Along the time axis t in arrival time tensor D, for each tuple (x, y), we find out all the moments using Def.4.1. all the moments are merged into hotspots using Def.4.2. At last, 932 hotspots are discovered.

Table 2 shows some hotspots that are discovered from the query dataset. Several kinds of hotspots are presented, including concerts, forums, places of interest and anniversaries. For each hotspot, there are many slightly different query words (e.g., "South Gate of the Capital Gym" and "Park of the Capital Gym"). One can find that the query counts are much more than that in last week, and the top 1 query word is highly related to the hotspot. Not only that, more than 80% of the query counts comes from the top 1 query word.

6.1.3 ***Correlation Analysis***. As the first step towards predicting traffic speed with map query data, the correlation between map query counts and the traffic speed in the same hotspot is tested. For each hotspot, the average traffic speed of the k ($k = 5$) road segments with a resolution of 5 minutes is used. And we stack the traffic speed and map query counts of all the hotspots, respectively. As both of the variables are both non-linear, the standard linear Pearson correlation coefficient is not appropriate for our study. Spearman's rank correlation coefficient is applied, and the result $\rho = -0.57$ with a P-value= $4.64 \times e^{-14}$ indicate that there is a strong negative correlation between the average traffic speed and the query counts, making it promising to predict traffic speed with the map query data.

6.2 Hotspot Traffic Speed Prediction

6.2.1 ***Baselines***. On one hand, for the traditional machine learning methods, due to the computation complexity of ARIMA [2], it is not suitable for a large-scale dataset. Thus we compared our method with the state-of-the-art approaches: RF [16] and SVR [14].

On the other hand, for the deep learning approaches, we compared our method with two RNN methods (LSTM and GRU [6]) in terms of traffic speed prediction. Furthermore, different variants of our Dest-ResNet have been trained for traffic speed prediction. Dest-ResNet considers the query features and the residual network. The query features consist of two kinds of features, the query counts feature (shortly, "C"), and the query spatial feature (shortly, "S") extracted with Algorithm 1. In addition, "R" represents the residual network ***Res*** for simplicity. In a word, we compare Dest-ResNet with the following methods:

- Random forests (RF) [16]: RF is a traditional machine learning method for regression, often used for traffic speed prediction;
- Support vector regression (SVR) [14]: SVR is a version of SVM for regression, widely used for traffic speed prediction;
- Seq2Seq: The Seq2Seq network ***Seq2Seq*** whose input is the current traffic speed slot and output is the future traffic speed slot;
- Gated Recurrent Unit (GRU) [6]: GRU is a variant of RNN. We compare Dest-ResNet with GRU which replaces the LSTM layers in the Seq2Seq network with GRU layers;
- Seq2Seq+P: The same as the Seq2Seq network, except for the input which consists of previous traffic speed slot and current traffic speed slot;
- Dest-ResNet(C): "S" and "R" are removed from our Dest-ResNet, and only "C" is retained;
- Dest-ResNet(R): "C" and "S" are removed from our Dest-ResNet, and only "R" is retained;
- Dest-ResNet(S): "C" and "R" are removed from our Dest-ResNet, and only "S" is retained;
- Dest-ResNet(CR): "S" is removed from our Dest-ResNet, and "C" and "R" are retained;
- Dest-ResNet(CS): "R" is removed from our Dest-ResNet, and "C" and "S" are retained;
- Dest-ResNet(RS): "C" is removed from our Dest-ResNet, and "S" and "R" are retained.

Table 2: Examples of the discovered hotspots from the query dataset, where Time, Grid, QC_cur, QC_last, Top1 query word, Top1_qc, and Description represents the start time and end time, grid coordinates, query counts in the current time period, query counts in the same period of last week, top 1 query word, top 1 query counts and the description of each hotspot, respectively.

Time	Grid	QC_cur	QC_last	Top1 query word	Top1_qc	Description
2017-04-08 14:00-20:00	(26, 39)	3431	417	Capital Gym	2724	Fish Leong Concert
2017-04-11 08:00-10:00	(24, 38)	447	93	Beijing Shangri-La Restaurant	304	IBM Data Scientist Forum
2017-04-15 08:00-16:00	(13, 47)	4551	2202	Beijing Botanical Garden	3849	Spring outing
2017-04-15 16:00-20:00	(21, 34)	2173	207	Letv sports center	1831	Chou Chuan-huing Concert
2017-04-30 08:00-18:00	(22, 47)	7283	3607	Summer Palace	7149	Summer Palace (May Day)
2017-04-30 08:00-18:00	(26, 46)	3691	1582	Tsinghua University	3102	106th Anniversary of THU

6.2.2 ***Parameter Settings.*** Given the current traffic speed slot $\boldsymbol{V}^c = (\boldsymbol{v}_{t-w+1}, \boldsymbol{v}_{t-w+2}, \ldots, \boldsymbol{v}_t)$, note that the standard RF and SVR can only predict the traffic speed $\boldsymbol{v}_{t+1}$ at time step $t+1$, whose goals are slightly different from our sequence to sequence model. So on the testing stage, we treat prior forecasts as "observations" and utilize them for subsequent forecasts, that is, we firstly use these models to output $\hat{\boldsymbol{v}}_{t+1}$, then we use $\boldsymbol{v}_{t-w+2}, \ldots, \boldsymbol{v}_t$ and the predicted $\hat{\boldsymbol{v}}_{t+1}$ to output $\hat{\boldsymbol{v}}_{t+2}$. Finally, we have the predicted future traffic speed slot $\widehat{\boldsymbol{V}}^f = (\hat{\boldsymbol{v}}_{t+1}, \hat{\boldsymbol{v}}_{t+2}, \ldots, \hat{\boldsymbol{v}}_{t+w})$. Allowing for that the traffic patterns of different kinds of hotspots vary, each hotspot discovered is classified into 16 classes according to its top 1 query word. For each kind of hotspot, we choose 70% (30%) as our training (test) set. The training (test) set consists of 656 (276) hotspots. For each hotspot, we set $w = 12$ and $k = 5$ in our experiments, in other words, we predict the traffic speed of 5 road segments for next hour meanwhile. These 5 road segments are selected according to the their distances to the center of the hotspot queries' destinations. As a result, the total number of the traffic speed samples is 238,830 and the corresponding number of queries is 2,336,114. We set $\sigma = 100$m in Algorithm 1.

The parameter settings of RF and SVR are motivated by [7], which is also a recent traffic prediction work. For RF, the number of trees in the forests and the maximum depth are both 10. For SVR, we set $C = 1$, $\epsilon = 0.1$ and use RBF kernel. The scikit-learn [25] is used to implement RF and SVR. For Seq2Seq (or GRU), we use the Seq2Seq network in Figure 3 which consists of two LSTM (or GRU) layers. This network is pre-trained for finetuning the other models. For Seq2Seq+P, we use the same structure as the Seq2Seq network except for the length of input is $2w = w + w$ (w for previous traffic speed slot and w for current traffic speed slot). For Dest-ResNet(C), Dest-ResNet(S) and Dest-ResNet(CS), we use the counts (and/or spatial) feature of the query to fine-tune the Seq2Seq network. Specifically, we use another sequence to sequence network to learn the representation of query slot $\Phi^f = (\boldsymbol{\phi}_{t+1}, \boldsymbol{\phi}_{t+2}, \ldots, \boldsymbol{\phi}_{t+w})$, and concatenate the representation of query slot and the representation of current traffic speed slot $\boldsymbol{V}^c$ before decoding them into $\widehat{\boldsymbol{V}}^f = (\hat{\boldsymbol{v}}_{t+1}, \hat{\boldsymbol{v}}_{t+2}, \ldots, \hat{\boldsymbol{v}}_{t+w})$. For Dest-ResNet(R), Dest-ResNet(CR), and Dest-ResNet(RS), we fine-tune Seq2Seq, Dest-ResNet(C) and Dest-ResNet(S) with the residual network ***Res***, respectively.

All the networks are trained with the stochastic gradient using the Adam optimizer [15] with a learning rate of 0.001. The squared loss and linear activation function $a(x) = x$ are utilized to train all the weights. The dimension of hidden units in LSTM (or GRU) is tuned by using a grid of parameter settings (8, 16, 32, . . .) without dropout and the best performance is reported. All the networks are implemented based on the publicly available framework Keras [5] with TensorFlow [1] as the backend, and are trained on a single NVIDIA GeForce GTX TITAN X GPU with 12GB memory. It takes the model about 2 hours to train and less 1 minute to test.

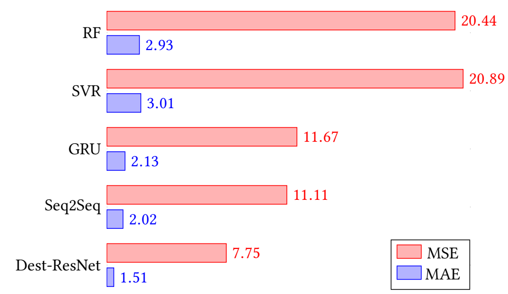


Figure 6: Best performance of different methods for traffic speed prediction. Lower MSE (MAE) means better performance. (Best viewed in the electronic version)

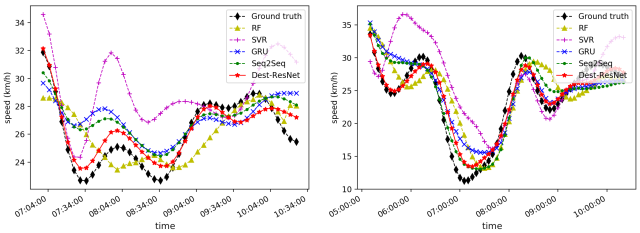
6.2.3 ***Evaluation Metrics.*** Two widely used evaluation metrics for traffic speed prediction are used, namely mean square error (MSE) and mean absolute error (MAE), which are defined as

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{v}_t - \hat{\boldsymbol{v}}_t)^2 \quad (15)$$

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|\boldsymbol{v}_t - \hat{\boldsymbol{v}}_t| \quad (16)$$

where $\boldsymbol{v}_t$ and $\hat{\boldsymbol{v}}_t$ are the ground truth traffic speed and the predicted traffic speed at time t, respectively.

6.2.4 ***Results.*** Firstly, the effectiveness of our proposed Dest-ResNe for traffic speed prediction is verified. Figure 6 demonstrates the prediction performance of different methods in terms of MSE and MAE. It can be observed that deep learning based methods(e.g., GRU, Seq2Seq and Dest-ResNet) outperform traditional methods(e.g., RF and SVR). This is because deep learning based methods have stronger expressive power to encode the complicated traffic speed condition. It is shown that Seq2Seq is slightly better than GRU, the difference between them is the sequence layer (LSTM for Seq2Seq,



(a) **(b)**

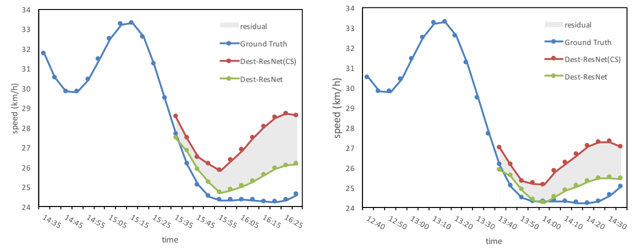
Figure 7: Traffic speed prediction with different methods. (Best viewed in the electronic version)

Table 3: Comparison of all the variants of our Dest-ResNet with the different number of hidden units. The results with the best performance are marked in bold.

Methods	Num units=8		Num units=16		Num units=32	
	MSE	MAE	MSE	MAE	MSE	MAE
Seq2Seq	14.34	2.41	11.11	2.02	11.32	2.02
Seq2Seq+P	13.98	2.35	10.97	1.97	10.86	1.95
Dest-ResNet(C)	12.80	2.35	9.13	1.90	9.25	1.75
Dest-ResNet(R)	14.26	2.36	9.84	1.83	8.54	1.78
Dest-ResNet(S)	11.56	2.20	8.76	1.81	8.76	1.68
Dest-ResNet(CR)	12.02	2.25	8.81	1.85	8.93	1.74
Dest-ResNet(CS)	11.74	2.21	8.35	1.74	8.71	1.65
Dest-ResNet(RS)	11.37	2.15	7.69	1.69	7.93	1.55
Dest-ResNet	**10.92**	**2.08**	**7.66**	**1.62**	**7.75**	**1.51**

GRU for GRU), so the LSTM is chosen as the sequence layer in our Seq2Seq network. Among all three deep learning based methods, our proposed Dest-ResNet performs the best. Compared to Seq2Seq, Dest-ResNet achieves 30% and 25% relative improvement in terms of MSE and MAE, respectively. This improvement can be explained by the fact that Dest-ResNet leverages the query features extracted from map query data as well as the residual network. Moreover, Figure 7 shows some qualitative traffic speed prediction samples of different methods. For simplicity, the performance of the variants is not presented.

Secondly, the effectiveness of the residual network "R" and query features are verified, which comprise of query counts feature "C" and the query spatial feature "S" extracted using Algorithm 1. As mentioned in Sec. 6.2.2, we tune the number of hidden units in the LSTM. Table 3 compares the performance of all the variants with the different number of hidden units. As the same as Dest-ResNet(R), Seq2Seq+P also use the previous and current traffic speed to predict future traffic speed. However, the improvement is limited with respect to Seq2Seq, which indicates the effectiveness of the residual network "R". The performances of Dest-ResNet(C), Dest-ResNet(R), and Dest-ResNet(S) are better than Seq2Seq, which demonstrate the effectiveness of the query counts feature "C", residual network "R", and the query spatial feature "S", respectively. Similar observations can be obtained compared to Dest-ResNet(CR), Dest-ResNet(CS) and Dest-ResNet(RS). As a result, our proposed Dest-ResNet which considers the query counts feature "C", residual network "R", and the query spatial feature "S" achieves the best performance. Furthermore, as the number of hidden units increases, the performance is improving. However, beyond 32 hidden units, there are diminishing returns — overfitting occurs with a large number of hidden units.



(a) **(b)**

Figure 8: Traffic speed prediction with Dest-ResNet(CS) and Dest-ResNet. The shadow area is the predicted residual using the residual network. It is utilized to amend the future traffic speed (in green) from the estimated future traffic speed (in red). (Best viewed in the electronic version)

Finally, the effectiveness of the residual network is demonstrated in Figure 8, where we show two samples of traffic speed prediction with/without the residual network. The shadow area, which is predicted by the residual network, is utilized to amend the future traffic speed (in green) from the estimated future traffic speed (in red).

7 CONCLUSION

We have studied the problem of how to model the map query and utilize it to boost hotspot traffic speed prediction. Towards this end, we firstly propose a hotspot discovery module to detect the representative spatiotemporal hotspots from massive map query data, addressing the spatiotemporal variations. Furthermore, an algorithm is devised to model the intricate spatial impact between queries and road segments. At last, we propose Dest-ResNet which integrates the sequence learning from different modalities. The residual network of Dest-ResNet encodes the information of map queries and utilize it to explain and amend the errors of the original traffic speed prediction, addressing the temporal casual correlation between the map queries and traffic speed. As a result, Dest-ResNet achieves 30% relative improvement over the state-of-the-art methods on real-world datasets from Baidu Map. As for future work, it is interesting to discover and recommend the hotspots for users, and provide traffic speed prediction with Dest-ResNet in practical applications.

ACKNOWLEDGMENTS

We would like to thank the support from Baidu Map and ZJU-Imperial Joint Lab. This work was supported by the National Basic Research Program (973) of China (No. 2015CB352300), the National Natural Science Foundation of China (Nos. 61625107, U1611461, U1509206), the Key Program of Zhejiang Province, China (No. 2015C01027), the Chinese Knowledge Center for Engineering Sciences and Technology, and the Fundamental Research Funds for the Central Universities, China. Jingqing Zhang was funded by LexisNexis HPCC Systems Academic Program.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Mohammed S Ahmed and Allen R Cook. 1979. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. Number 722.

[3] Manoel Castro-Neto, Young-Seon Jeong, Myong-Kee Jeong, and Lee D Han. 2009. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert systems with applications* 36, 3 (2009), 6164–6173.

[4] H Chang, Youngjoo Lee, B Yoon, and Sanghoon Baek. 2012. Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences. *IET intelligent transport systems* 6, 3 (2012), 292–305.

[5] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras. (2015).

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[7] Xingyuan Dai, Rui Fu, Yilun Lin, Li Li, and Fei-Yue Wang. 2017. DeepTrend: A Deep Hierarchical Neural Network for Traffic Flow Prediction. *arXiv preprint arXiv:1707.03213* (2017).

[8] Gary A Davis and Nancy L Nihan. 1991. Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering* 117, 2 (1991), 178–188.

[9] Fredrik Gustafsson. 1996. Determining the initial states in forward-backward filtering. *IEEE Transactions on Signal Processing* 44, 4 (1996), 988–992.

[10] Jingrui He, Wei Shen, Phani Divakaruni, Laura Wynter, and Rick Lawrence. 2013. Improving Traffic Prediction with Tweet Semantics.. In *IJCAI*. 1387–1393.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[12] Shan-Huen Huang and Bin Ran. 2003. *An application of neural network on traffic speed prediction under adverse weather condition*. Ph.D. Dissertation. University of Wisconsin–Madison.

[13] Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie. 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems* 15, 5 (2014), 2191–2201.

[14] Xuexiang Jin, Yi Zhang, and Danya Yao. 2007. Simultaneously prediction of network traffic flow based on PCA-SVR. *Advances in Neural Networks–ISNN 2007* (2007), 1022–1031.

[15] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Guy Leshem and Yaacov Ritov. 2007. Traffic flow prediction using adaboost algorithm with random forests as a weak learner. In *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 19. 193–198.

[17] Moshe Levin and Yen-Der Tsao. 1980. On forecasting freeway occupancies and volumes (abridgment). *Transportation Research Record* 773 (1980).

[18] Binbing Liao, Siliang Tang, Shengwen Yang, Wenwu Zhu, and Fei Wu. 2018. Multi-modal sequence to sequence learning with content attention for hotspot traffic speed prediction. In *Pacific Rim Conference on Multimedia*. Springer.

[19] Binbing Liao, Jingqing Zhang, Chao Wu, Douglas McIlwraith, Tong Chen, Shengwen Yang, Yike Guo, and Fei Wu. 2018. Deep sequence learning with auxiliary information for traffic prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[20] Xinyue Liu, Xiangnan Kong, and Yanhua Li. 2016. Collective Traffic Prediction with Partially Observed Traffic History using Location-Based Social Media. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2179–2184.

[21] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2015. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2015), 865–873.

[22] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* 54 (2015), 187–197.

[23] Fabio Moretti, Stefano Pizzuti, Stefano Panzieri, and Mauro Annunziato. 2015. Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling. *Neurocomputing* 167 (2015), 3–7.

[24] Jungme Park, Dai Li, Yi L Murphey, Johannes Kristinsson, Ryan McGee, Ming Kuang, and Tony Phillips. 2011. Real time vehicle speed prediction using a neural network traffic model. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, 2991–2996.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, Oct (2011), 2825–2830.

[26] Shiliang Sun, Changshui Zhang, and Guoqiang Yu. 2006. A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems* 7, 1 (2006), 124–132.

[27] Quang Thanh Tran, Zhihua Ma, Hengchao Li, Li Hao, and Quang Khai Trinh. 2015. A multiplicative seasonal ARIMA/GARCH model in EVN traffic prediction. *International Journal of Communications, Network and System Sciences* 8, 04 (2015), 43.

[28] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. *Transportation Research Part C: Emerging Technologies* 13, 3 (2005), 211–234.

[29] Billy Williams. 2001. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. *Transportation Research Record: Journal of the Transportation Research Board* 1776 (2001), 194–200.

[30] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.

[31] Dawen Xia, Binfeng Wang, Huaqing Li, Yantao Li, and Zili Zhang. 2016. A distributed spatial–temporal weighted model on MapReduce for short-term traffic flow forecasting. *Neurocomputing* 179 (2016), 246–263.

[32] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. 2011. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 12, 4 (2011), 1624–1639.

[33] Weizhong Zheng, Der-Horng Lee, and Qixin Shi. 2006. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of transportation engineering* 132, 2 (2006), 114–121.