# A Traffic Prediction Enabled Double Rewarded Value Iteration Network for Route Planning

Jinglin Li, *Member, IEEE*, Dawei Fu, Quan Yuan, Haohan Zhang, Kaihui Chen, Shu Yang, and Fangchun Yang, *Senior Member, IEEE*

***Abstract*—Effective route planning is the key to improving transportation efficiency. By leveraging the in-depth knowledge of road topology and traffic trends, experienced drivers (e.g., taxi drivers) can usually find near-optimal routes. However, existing online route planning services can hardly acquire this domain knowledge, so they just provide the fastest/shortest route based on current traffic conditions. These seemingly optimal routes may attract numerous vehicles and then become extremely congested. To solve this problem and actually improve transportation efficiency, we propose a double rewarded value iteration network (VIN) to fully learn the experienced drivers' routing decisions, which are based on their implicitly estimated traffic trends. First, the global traffic status and routing actions are chronologically extracted from large-scale taxicab trajectories. Then, to model the knowledge of traffic trends, a long short-term memory network is trained. Being expert at learning long-term planning involved functions, the VIN is utilized to model the policy function from both current and predicted future traffic status to an experienced driver's routing action. Finally, the performance of our proposed model is evaluated on real map and taxicab trajectories in Beijing, China. The experimental results demonstrate that the proposed model can achieve human like performance in most cases, with high success rate and less commuting time.**

***Index Terms*—Route planning, traffic prediction, double rewards, value iteration network.**

## I. Introduction

TRAFFIC congestion has long been a serious concern for big cities. Without comprehensive real-time traffic information, drivers can only make routing decisions based on their limited visions. These shortsighted and non-cooperative routing decisions inevitably deteriorate the resource utilization efficiency of road networks. In recent years, the ever-increasing capabilities of vehicular sensing, computing and communication provide an opportunity to improve urban transport [1], [2]. Specifically, the 5G enabled vehicular ad hoc network (5G-VANET) facilitates the real-time traffic information exchanging between vehicles and infrastructure [3], [4]. Furthermore, cloud and edge computing paradigms assist vehicles with complex information processing (e.g., data fusion, and traffic prediction) and real-time decision making capabilities [5]–[7]. Combined with these state-of-the-art technologies, big data and machine learning play an increasingly important role in reducing traffic congestion, improving road safety, and enhancing driving comfort [8], [9].

Though comprehensive real-time traffic information can be obtained, route planning remains challenging. Vehicles that plan the fastest/shortest routes by only considering current traffic conditions may fall into newly emerged congestions. This is because the currently optimal routes may attract numerous vehicles and be blocked in the near future [10]. Recently, some studies try to balance future traffic flows when planning routes [11], [12], which are effective only when sufficient drivers follow the central scheduling. In reality, route planning algorithm cannot meet all the personalized needs of drivers, so the actual routes usually deviate from the scheduling. In addition, it is almost impossible to use existing technology to thoroughly model all the complex factors (e.g., road topology, road condition, and social events) that impact on the traffic trends. However, experienced drivers can utilize these latent factors to infer traffic trends and plan near-optimal routes.

In this paper, we propose an online route planning model to provide time-saving routes by extracting latent routing knowledge, which leverages both current and estimated future traffic status, from massive trajectories of experienced drivers. Three key issues are focused in the model: i) future traffic status prediction; ii) routing knowledge learning from experienced drivers; and iii) route planning based on both current and future traffic status. Our preliminary work [13] has trained a value iteration network (VIN, hereinafter referred to as simple VIN) to give a policy function from current traffic status to a routing action. However, experienced drivers usually plan routes based on their implicitly estimated future traffic status. As this traffic prediction feature has not been considered in [13], its learned routing actions may deviate from the near-optimal routing behavior of experienced drivers. Therefore, in this paper, we innovatively propose a double rewarded VIN to fully learn the routing knowledge of experienced drivers. Specifically, long short-term memory network (LSTM) is used to predict future traffic status. Then, current and predicted traffic status are combined to serve as double rewards for VIN. Our contributions can be summarized as follows:

Manuscript received September 1, 2018; revised November 6, 2018 and December 16, 2018; accepted January 11, 2019. Date of publication January 16, 2019; date of current version May 28, 2019. This work was supported in part by the Natural Science Foundation of Beijing under Grant 4181002, and in part by the Natural Science Foundation of China under Grant 61876023. The review of this paper was coordinated by the Guest Editors of the Special Section on Machine Learning-Based Internet of Vehicles. *(Corresponding author: Quan Yuan.)*

J. Li, D. Fu, Q. Yuan, H. Zhang, K. Chen, and F. Yang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: jlli@bupt.edu.cn; fudawei@bupt.edu.cn; yuanquan@bupt.edu.cn; hhzhang@bupt.edu.cn; kayc@bupt.edu.cn; fcyang@bupt.edu.cn).

S. Yang is with the IVBU, Baidu, Beijing 100085, China (e-mail: yangshu629@gmail.com).

Digital Object Identifier 10.1109/TVT.2019.2893173

- **LSTM network for traffic prediction**. Precise future traffic status is important to generate time-saving routes. However, due to nonlinear and stochastic characteristics of complex transportation systems, traditional methods (e.g., regression analysis, Kalman filtering, time series analysis, and shallow neural network) do not perform well. Taking advantage of the long temporal dependency in traffic status series, we adopt LSTM to learn traffic patterns and predict short-term traffic status.
- **Double rewarded VIN**. In our preliminary work, a Markov decision process (MDP) is generated by current traffic status and routing actions. This MDP is processed by the simple VIN after $K$ iterations to plan routes. In this paper, we add a new MDP which is generated by future traffic status and current routing actions. By leveraging the proposed double rewarded VIN, these two MDPs are processed separately by two sub-VINs (i.e., current VIN and prediction VIN) in the first $K-1$ iterations, and aggregated in the last iteration to give actions as well as perform back propagation. Particularly, the combination of two sub-VINs is based on an intuition that current VIN focuses more on short-term planning, whereas prediction VIN focuses more on long-term planning
- **Network training and performance evaluation**. To validate the planning ability of the proposed model which contains LSTM module for traffic prediction and double rewarded VIN module for route generation, real road networks and taxi trajectories in Beijing city are used. We show that the proposed model performs better than Dijkstra, Gaussian process regression, and simple VIN. It can achieve human-like driving performance with high success rate and less commuting time.

The remainder of this paper is organized as follows. Section II reviews related research works on traffic prediction and route planning. Section III describes the learning model and problem statement for route planning. Section IV introduces preliminary knowledge of LSTM and VIN. Section V details the network structure and training algorithms. The effectiveness of the proposed model is demonstrated in Section VI. The paper ends in Section VII with conclusion and discussion.

## II. Related Work

### A. Traffic Prediction

Traffic flow prediction has a wide range of applications, including route planning, congestion management, and traffic forecast. Time series analysis and regression analysis are widely used methods for traffic flow prediction. Assuming that the time series of traffic status can be made to be "stationary" by differencing, Levin *et al.* [14] and Hamed *et al.* [15] used an autoregressive integrated moving average (ARIMA) model to predict traffic flow. By modeling the traffic flow per junction as a joint multivariate Gaussian distribution, Liebig *et al.* [16] adopted Gaussian process regression (GRP) to perform real-time traffic predictions.

These shallow models could give not far-off results under the condition of insufficient computing power. However, they can hardly describe the nonlinear and stochastic features of traffic flow, while deep neural networks can learn these features much better. Polson *et al.* [17] developed a deep learning model, which combines a linear model and a sequence of tanh layers, for predicting short-term traffic flow. To improve prediction accuracy, Lv *et al.* [18] adopted a stacked autoencoder model and trained the model in a greedy layerwise fashion. By converting spatio-temporal traffic dynamics into images, Ma *et al.* [19] leveraged convolutional neural network (CNN)-based method to learn traffic as images and predict large-scale traffic status. As LSTM unit can capture long time series information, Ma *et al.* [20] also utilized LSTM network to predict long-term traffic flow. Moretti *et al.* [21] proposed a hybrid model which combines multiple neural networks and traditional statistical approaches to provide traffic flow forecast.

### B. Route Planning

For the rapidly growing mobility demands in this era of unprecedented global urbanization, computing and communication technologies are highly beneficial to the improvement of people's travel experience. As a core issue in the transportation field, route planning is mainly divided into traditional methods and deep learning based methods.

Liebig *et al.* [16] and Yamashita *et al.* [22] proposed using centralized server to integrate multi-source information and then feedback to the driver for route decision-making. Some studies [11], [23]–[25] adopted route optimization schemes for adding additional goals such as user preferences, and response to emergency situations. These schemes only give optimal routes based on current traffic status, so if all drivers rely on the schemes to plan routes, they may fall into newly emerged congestion caused by traffic gathering. However, as automated driving gradually becomes commercial, it is possible to control all the behavior of vehicles. It means that future traffic status can be balanced by scheduling vehicles. Talebpour *et al.* [26] shown that controlling autonomous vehicles by central management would significantly improve the efficiency of transportation system. Sundar *et al.* [27] adopted a route planning solution for gaining the optimal cost when an automated driving team needs to arrive some destinations given fixed gas stations. Hiraishi [28] innovatively proposed a method that uses the sitting-pressure distribution of passenger as a reference factor for autonomous vehicle route planning.

In the era of deep learning, more effective route planning can be achieved. Nazari *et al.* [29] and Zolfpour-Arokhlo *et al.* [30] used deep reinforcement learning to quickly give an approximate optimal routing solution. Brahmbhatt *et al.* [31] proposed a convolutional neural network based algorithm for navigating in large cities using locally visible street view images. However, most existing methods give optimal routes based on current traffic status without considering future traffic trends, which are influenced by factors such as dynamic traffic demand, road condition, weather, and social events. Although some recent methods (e.g., [16]) planned routes based on predicted short-term traffic status, the neglect of long-term traffic status cause frequent re-routing and route jitter. To solve this problem, game

theory-based methods [32], [33] consider others' routing decisions to balance future traffic flow, but individuals without exhaustive cognition and enough experience can hardly get efficient routes. In this paper, we try to plan routes effectively and coordinately by learning experienced drivers' knowledge.

## III. System Model and Problem Statement

This paper aims to generate time-saving routes by learning experienced drivers' routing actions. As experienced drivers (e.g., taxi drivers) have an in-depth knowledge of traffic trends and relevant complex factors, their routing behavior can be adaptive to the dynamic road traffic. In other words, experienced drives can usually generate near-optimal routes by leveraging their implicitly estimated future traffic status. Therefore, learning route planning from experienced drivers is a feasible way to improve transportation efficiency. In addition, we have found that VIN can get better generalization performance than traditional neural networks such as fully connected network (FCN), CNN [13] and Bayesian model [34]. To this end, we propose a VIN based learning model for route planning in Fig. 1, which considers the influence of dynamic traffic on routing behavior. The proposed model consists of data preprocessing module, prediction module, double rewarded VIN module, and decision module.

The data preprocessing module uses the city map which is partitioned into disjoint grids. Trajectory profiles of experienced drivers are fed into two information pipelines. One pipeline merges speed data to produce traverse time map (TTM, defined in Section V), and the other discretizes each trajectory into routing actions (decision sequence) in a grid-world view. In the prediction module, to imitate experienced drivers' knowledge on traffic trends, historical TTMs are fed into LSTM network to predict a future TTM. As for the double rewarded VIN module, current and predicted future TTMs, aligning with routing actions by timestamps, are fed into current VIN and prediction VIN, respectively. Meanwhile, the combination of these two sub-VINs is learned with an intuition that current VIN contributes more to the short-term planning, whereas prediction VIN contributes more to the long-term planning. The double rewarded VIN is taught to make a routing action given two TTMs. The structures of TTMs and actions are co-designed so that VIN can be trained end to end. In the decision module, VIN is capable of generating a time-saving route using learned value when faced with a new scenario (i.e., TTM). The system plans routes every $\Delta T$ minutes for drivers, which is shown in Fig. 1, and any dynamic events are reflected in next traffic status when planning routes. Therefore, the response time of dynamic events is less then $\Delta T$ minutes.

## IV. Preliminary

In this section, we briefly introduce LSTM and VIN, which are the basic components of our prediction enabled doubled rewarded VIN.

### A. Long Short-Term Memory Network

LSTM is a recurrent neural network (RNN) composed of several LSTM units. Original RNN relies on hidden state to save previous information, so it can exhibit dynamic temporal behavior for a time series. However, original RNN suffers from vanishing/exploding gradient which makes it difficult to capture long-term time correlations. To solve this problem, a common LSTM unit is composed of a cell, an input gate $i_t$, an output gate $o_t$ and a forget gate $f_t$. Three gates can be computed by following equations,

$$i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1}\right), \tag{1}$$

$$f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1}\right), \tag{2}$$

$$o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1}\right), \tag{3}$$

where $x_t$ is the input data for this unit; $h_{t-1}$ is previous unit's hidden state; $\sigma(\cdot)$ denotes sigmoid function which can map any real number to $(0, 1)$; and $W^{(i)}$, $W^{(f)}$, $W^{(o)}$, $U^{(i)}$, $U^{(f)}$ and $U^{(o)}$ are parameters for LSTM network which need to be learned by training.

New cell state $c_t$ is updated by Eq. (4) using: old cell state $c_{t-1}$, with forget gate $f_t$ to determine the degree of forgetting earlier information; and new candidate vector $\widetilde{c_t}$ (defined in Eq. (5), which is a mix of input and previous output), with input gate $i_t$ to determine the degree of using input data. The cell is responsible for "remembering" values over arbitrary time intervals. LSTM uses cell states to transfer information, which is suitable for processing and predicting important events with long intervals and delays in time series. Finally, LSTM unit uses the cell state and output gate to output $h_t$ by Eq. (6).

$$c_t = f_t \circ c_{t-1} + i_t \circ \widetilde{c_t}, \tag{4}$$

$$\widetilde{c_t} = \tanh\left(W^{(c)}x_t + U^{(c)}h_{t-1}\right), \tag{5}$$

$$h_t = o_t \circ \tanh(c_t), \tag{6}$$

where $W^{(c)}$ and $U^{(c)}$ are parameters for new cell, and $\circ$ means matrix element-wise multiplication.

### B. Value Iteration Network

A standard model for sequential decision making and planning is an MDP [35]. The MDP consists of a state set $\mathcal{S}$ ($s \in \mathcal{S}$), an action set $\mathcal{A}$ ($a \in \mathcal{A}$), a reward function $R(s, a)$, and a transition matrix $P(s'|s, a)$ that encodes the probability of the next state $s' \in \mathcal{S}$ given the current state $s$ and action $a$. A policy $\pi(a|s)$ describes an action distribution for every state. The goal of the MDP is to find a policy that obtains high rewards in the long term. Formally, the value $V^\pi(s)$ of a state under policy $\pi$ is the expected discounted sum of rewards when starting from state $s$,

$$V^\pi(s) = \mathbb{E}^\pi\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\Big|s_0 = s\right] \tag{7}$$

where $\gamma \in (0, 1)$ is a discount factor, $\mathbb{E}^\pi$ is an expectation over trace of states and actions $(s_0, a_0, s_1, a_1, \ldots)$, in which actions are selected according to $\pi$, and states evolve according to $P(s'|s, a)$. The optimal value function $V^*(s) \doteq \max_\pi V^\pi(s)$ is the maximal long-term reward possible from state $s$. A policy

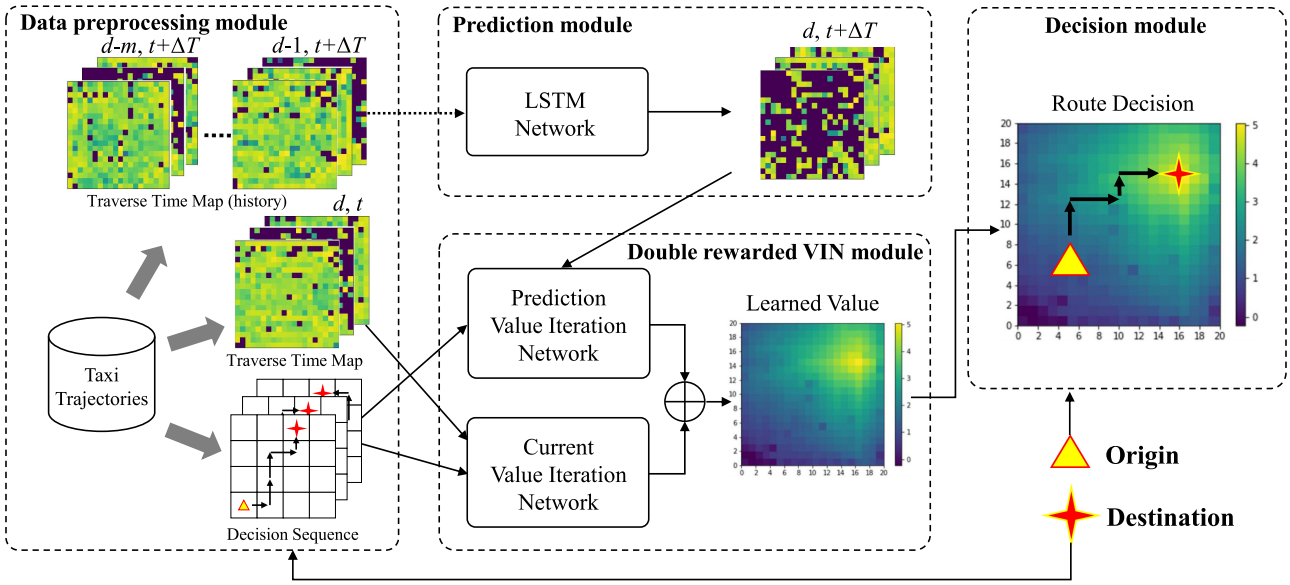



Fig. 1. A learning model for route planning. It consists of four modules: data preprocessing module, prediction module, double rewarded VIN module, and decision module.

$\pi^*$ is said to be optimal if $V^{\pi^*}(s) = V^*(s)$ for $\forall s$. A popular algorithm for calculating $V^*$ and $\pi^*$ is the value iteration (VI),

$$V_{n+1}(s) = \max_a Q_n(s,a), \ \forall s, \tag{8}$$

where

$$Q_n(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V_n(s'). \tag{9}$$

The value function $V_n$ in VI converges to $V^*$ as $n \to \infty$, from which an optimal policy can be derived as

$$\pi^*(s) = \arg\max_a Q_\infty(s,a). \tag{10}$$

It is interesting to adapt MDP and VI into our grid-world scenario and exploit them to learn the optimal routing policy. We build on the work from Tamar *et al.* [36] who studied how explicit planning can be incorporated in neural network structure, but they did not consider the case in large-scale city with dynamic traffic status.

## V. Traffic Prediction Enabled Double Rewarded Value Iteration Network

### A. Network Structure

*Definition 1 (Grid Map):* Our route planning is coarse-grained therefore the urban map is modeled as a $X \times Y$ grid-based graph $(G, E)$, where the grids are denoted as vertex set $G$, and adjacent grids in 8 directions of each grid are reachable through virtual edge set $E$. The set of grids is denoted by $G = \{g_{0,0}, g_{0,1}, \ldots, g_{x,y-1}, g_{x,y}, \ldots, g_{X-1,Y-1}\}$, where subscripts $x$ and $y$ depict grid's location in two-dimensional coordinates. An edge weight $e_{g,g'}$ $(g, g' \in G)$ represents the consumed time driving from grid $g$ to its adjacent grid $g'$.

*Definition 2 (Traverse Time Map):* Considering that one vehicle in grid $g$ could transit to 8 adjacent grids, the traverse time map $\mathcal{E}$ should consist of 8 layers, i.e., $\mathcal{E} = \{E^{l_1}, \ldots, E^{l_8}\}$. For any $e_g^l \in E^l, l \in \{l_1, \ldots, l_8\}$, it represents the traverse time from grid $g$ to its adjacent grid in the $l$th direction. The TTM $\mathcal{E}$ can be obtained by merging historical GPS-based trajectories. Our layered model is an approximation, we hope that the transition among grids could somehow reflect road topology and TTM could reflect traffic flow speed. Fortunately, this approximation works fine if appropriate grid granularity is selected.

Based on the processing of TTM, the structure of our traffic prediction enabled double rewarded VIN is depicted in Fig. 2. The whole network is driven by experienced drivers' trajectories. First, to predict next TTM in period $t + \Delta T$, TTMs in the same period of last $m$ days are fed into LSTM, where $t$ is current time of the day, and $\Delta T$ is the time interval between two successive TTMs. The LSTM network contains two LSTM layers and three fully-connected layers (FCL). To be adapted to VIN, each TTM is transformed to a 9-layer reward map by normalizing and combining an additional layer which represents destination. Then, the double rewards transformed from current TTM and next TTM are fed into current VIN and prediction VIN, respectively. These two sub-VINs are trained separately in the first $K-1$ iterations to learn short-term and long-term planning. The value map can be learned after the $K$th iteration, which combines two sub-VINs. Finally, time-saving routes are generated by iteratively choosing next optimal state in the learned value map.

### B. Building MDP

*Definition 3 (Driving Action Sequence):* Each experienced driver's trajectory is mapped into the grid map and discretized into driving action sequence set $\{a_1, \ldots, a_i, \ldots\}$, where $a_i \in \mathcal{A} = \{l_{1\sim 8}\}$ represents a choice among 8 actions.

Historical TTMs are used to predict next TTM by LSTM network. After aligning current TTM, predicted TTM and action according to timestamps, we build two MDPs including double rewards (i.e., current reward and future reward) with current

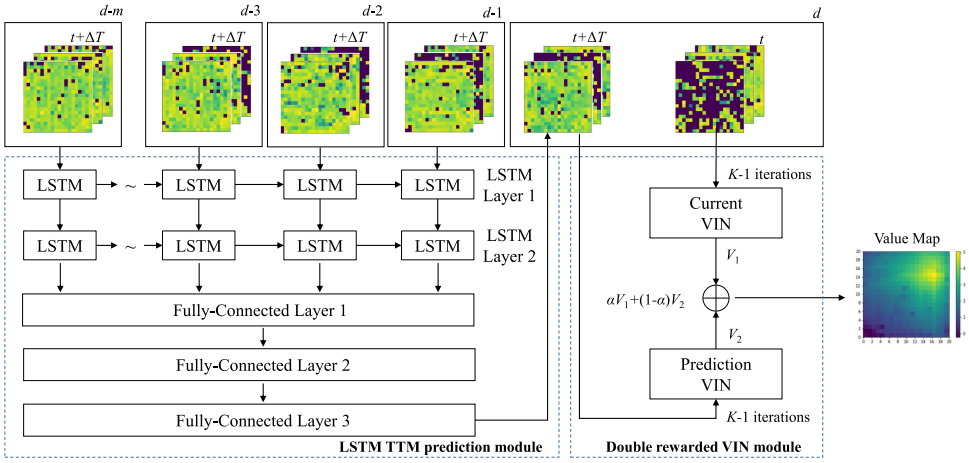



Fig. 2. Network structure of the traffic prediction enabled double rewarded VIN.

action, which grasp the essence of route planning. An MDP $M$ consists of a state set $\mathcal{S}$ ($\mathcal{S} = G$, $s = g \in G$), an action set $\mathcal{A}$ ($a \in \mathcal{A}$), a reward function $R(s,a) = R(g,a) = e_g^a \in E^a \subset \mathcal{E}$, and a transition matrix $P(s'|s,a) = P(g'|g,a)$ ($s' = g' \in G$) that encodes the probability of the next grid given the current grid and action.

Once MDP of routing has been specified, standard planning algorithm can be used to obtain the value function $V^*$. In our algorithm, double rewarded VIN module needs $K$ iterations to generate action. Double rewards are trained separately in the first $K-1$ iterations, and they will be aggregated in the last iteration. An additional property of $V^*$ in route planning scenario is that the optimal decision $\pi^*(s)$ at a state $s$ only depends on a subset of the value of $V^*$, since

$$\pi^*(s) = \arg\max_a \left[ R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right]. \quad (11)$$

The transition has a local connectivity structure, the states for which $P(s'|s,a) > 0$ is a small subset of $\mathcal{S}$, i.e., only 8 adjacent grids or ego grid $s'$ are reachable to $s$.

### C. Training Algorithm of LSTM

The algorithm for prediction module is shown in Algorithm 1. The network needs TTMs in last $m$ days to predict next TTM, these data are organized as $pair_t = (input_t, future_t)$. Particularly, only 8 (i.e., traverse time to 8 adjacent grids) out of 9 layers in each TTM need to be considered in the prediction module. In order to adapt to the input structure of LSTM network, we transform each TTM which is represented by a $X \times Y \times 8$ tensor to a $8XY \times 1$ tensor. Therefore, $input_t$ is $m$ tensors of size $8XY \times 1$, which are TTMs in the same period of previous $m$ days, $future_t$ is one $8XY \times 1$ tensor which represents the next TTM we need.

The whole network contains two LSTM layers and three FCLs which can increase nonlinearity and improve fitting accuracy than pure LSTM network. In this network, when tensor flows out of LSTM layers, $8XY \times 1$ tensor is increased to $Z_1$ dimensions to enhance expressiveness and then fed into FCLs. The first two FCLs contain $Z_1$ and $Z_2$ units, respectively, and the last FCL contains $8XY$ units. In order to gradually reduce dimension to the target, the numbers of units in three FCLs follow $Z_1 \geq Z_2 \geq 8XY$. So, final output is $8XY$ dimensions and can be easily transformed to an $X \times Y \times 8$ tensor which represents next TTM.

**Algorithm 1:** Training LSTM.

```
Initialize data set D_LSTM = {pair_1, pair_2, ..., pair_T}
Initialize LSTM network with weights θ
for epoch = 1, N do
    Initialize mini-batch number num = T/batch_size
    for i = 0, num do
        j = i + batch_size
        pr_data = pair_{i:j}
        Perform a gradient descent step Δθ on loss =
          RMSE{pr_data.future, LSTM(pr_data.input; θ)}
          with respect to LSTM parameters θ ← θ + Δθ
```

At last, dataset $\mathcal{D}_{LSTM} = \{pair_1, pair_2, \ldots, pair_T\}$ is divided into small batches and then fed into LSTM network which uses mini-batch gradient descent to train.

### D. Training Algorithm of VIN

We co-design data structure and VIN, which incorporates a differentiable planning computation. The main observation is that each iteration of value iteration could be seen as passing previous value function $V_n$ and reward function $R$ through a convolution layer and a max-pooling layer. Therefore, by recurrently applying convolution and max-pooling one time, one iteration of value iteration is performed. Current TTM and predicted TTM generated from prediction module are trained separately in the first $K-1$ iterations and combined in the last iteration. Near-optimal routing policy could be derived from a converged value function $V_n$ if $n$ is big enough. Particularly, the sub-VIN based on current TTM focuses on short-term planning while the sub-VIN based on predicted TTM focuses on long-term planning, so they are trained separately in the first $K-1$ iterations to derive short-term and long-term value maps. These two value maps are fused into a final value map

**Algorithm 2:** Training double rewarded VIN.

Initialize Data set $\mathcal{D}_{VIN} = \{triad_1, triad_2, \ldots, triad_k\}$
Initialize action-value function $V$ with weights $\theta$
**for** $epoch = 1, N$ **do**
  Initialize mini-batch number $num = k/batch\_size$
  **for** $i = 0, num$ **do**
    $j = i + batch\_size$
    $fd\_data = triad_{i:j}$
    Perform a gradient descent step $\Delta\theta$ on $loss = Cross\_Entropy\{fd\_data.a, V(fd\_data.TTM, fd\_data.s; \theta)\}$ with respect to VIN parameters $\theta \leftarrow \theta + \Delta\theta$

in the last iteration to generate routing actions which considers both current and near-future traffic status.

From this idea, we propose a double rewarded VIN module, as shown in Fig. 3. The input to the module is current and future multi-channel reward map $R$ (i.e., TTM after normalization). The rewards are fed into a convolution layer $Q$ with $\|\mathcal{A}\| = 9$ channels. Each channel in this layer corresponds to $Q(s, a)$ for a particular routing action $a$. This layer is then max-pooled along the 9-action channel to produce the next-iteration value function layer $V$. The next-iteration value function layer $V$ is then stacked with multi-channel reward map, and fed back into the convolutional layer and max-pooling layer. In VIN module, `Convolutions` and `Pooling` are used to compute $Q(s, a)$ in value iteration process. Specifically, two groups of `Convolutions` intend to extract the features of reward map and value map separately, then `Pooling` fuses these features into the $Q$-function for 8 next-step actions. Finally, `Max-pooling` operation produces new value for every state through choosing maximum value of $Q(s, a), \forall a \in \mathcal{A}$.

The algorithm for training VIN is presented in Algorithm 2. The initial neural network learns by batches. As using historical data of arbitrary length as inputs is difficult, our VIN instead works on a fixed length representation of historical triads. The drivers' experiences at each timestamp are organized as $triad_t = (TTM_t current, TTM_t predicted, a_t)$ in a dataset $\mathcal{D}_{VIN} = \{triad_1, \ldots, triad_t, \ldots, triad_k\}$. The timestamp is coarse-grained and counted periodically. The dataset $\mathcal{D}_{VIN}$ is shuffled and divided into many mini-batches. During the inner loop of the algorithm, we apply mini-batch gradient descent to update network parameters $\theta$.

VIN is trained by comparing predicted action, which is the direction with maximum value in the learned value map, with the real action of drivers. Here, a cross-entropy loss function is employed. Finally, VIN uses learned value map to determine a route from source to destination by choosing maximum value of actions step by step. Once multi-channel reward and VIN are co-designed, the training data can flow from TTM to routing action straightforwardly. This enables VIN to be trained end to end.

## VI. Experiment

In this section, we use real trajectories in Beijing city to demonstrate the performance of the proposed traffic prediction enabled double rewarded VIN.

### A. Data Preprocessing

*Grid map:* The grid map $(G, E)$ is built from Beijing map within the 4th Ring Road. The detailed information of map is trivial for coarse-grained navigation, therefore only grid model is built in our experiment. Each grid has its position and traverse directions.

*Taxi trajectory data:* We extract routes from real taxi trajectory data in Beijing. The data package includes over 400,000 taxis' trajectories in November 2012. For each day the data package contains full-scale GPS trajectories during 24 hours. In order to learn general pattern of routing actions with respect to traffic status, only the data between 8:00 and 20:00 in weekdays are used because the data in other time periods do not represent general pattern. Since trajectory is too detailed to use, it is simplified to grid-based route by trajectory-grid-matching, i.e., mapping trajectory into sequence of grids, and tagging each grid with arriving/leaving time. Since the goal of this paper is to learn (near) optimal route in city, and "no-passenger" taxis probably have no optimal route patterns, we only utilize the trajectories of taxicabs which are in "carrying-passenger" state.

*Traverse time map:* For a certain grid $g$, a traverse direction $l_{1\sim8}$ and a fixed period $[t, t + \Delta T]$, we extract traverse time of all passing by taxis, and average them to get one "reward grid" $e_g^{l,t} = -average\ time$ in TTM. Some grids' values $e_g^{l,t}$ are blank because of lacking trajectory data, their values are set to a large negative reward indicating that grid $g$ in $l$th direction is likely impossible. To get good performance in mini-batch gradient descent, TTM is normalized into $[-1, 0]$ by

$$\mathrm{TTM}_{\mathrm{norm}} = \frac{\mathrm{TTM} - \mathrm{TTM}_{\max}}{\mathrm{TTM}_{\max} - \mathrm{TTM}_{\min}}. \tag{12}$$

For illustration purpose, traffic status data on November 8, 2012 from 8:00 to 20:00 with a two-hour interval are shown in Fig. 4. The traverse time of all directions of a grid is added to show the big difference of traffic status in different grids and different time periods.

For prediction module, we use four 8-layer TTMs to predict future TTM. Then, future TTM and current TTM are fed into VIN module to generate routes. For VIN module, we need to add one more layer called destination layer besides 8-layer TTM for both current and future TTM. Destination layer express the destination $(x_d, y_d)$ information which indicates a big reward for arriving destination. Finally, $l_{1\sim8}$ and $l_0$ are concatenated into a 9-layer TTM, which contains both global status as well as destination information. As Fig. 5 shows, two 9-layer TTMs which represent current TTM and predicted TTM are calculated and fed to VIN, which later gives out a learned value map. By gradually moving from dark grid (far from destination) to bright yellow grid (destination), one can find routes that VIN has learned.

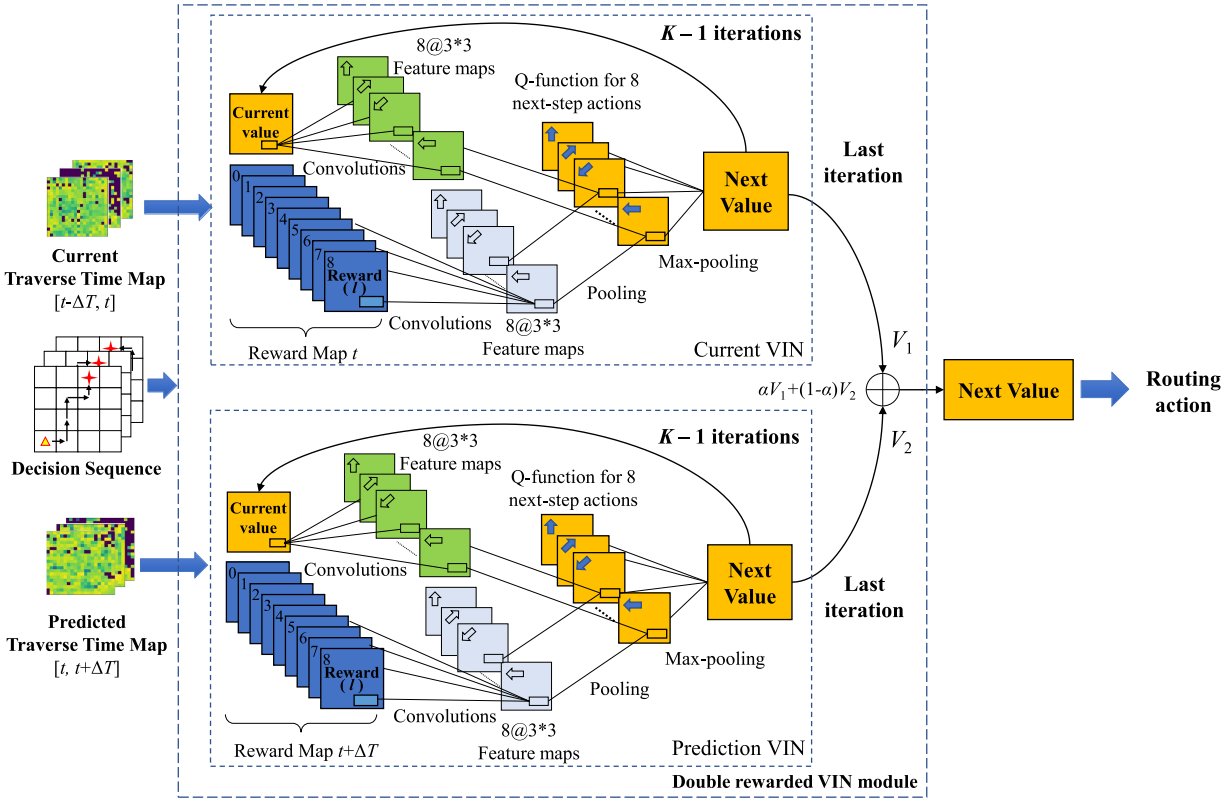



Fig. 3. Double rewarded value iteration network module.

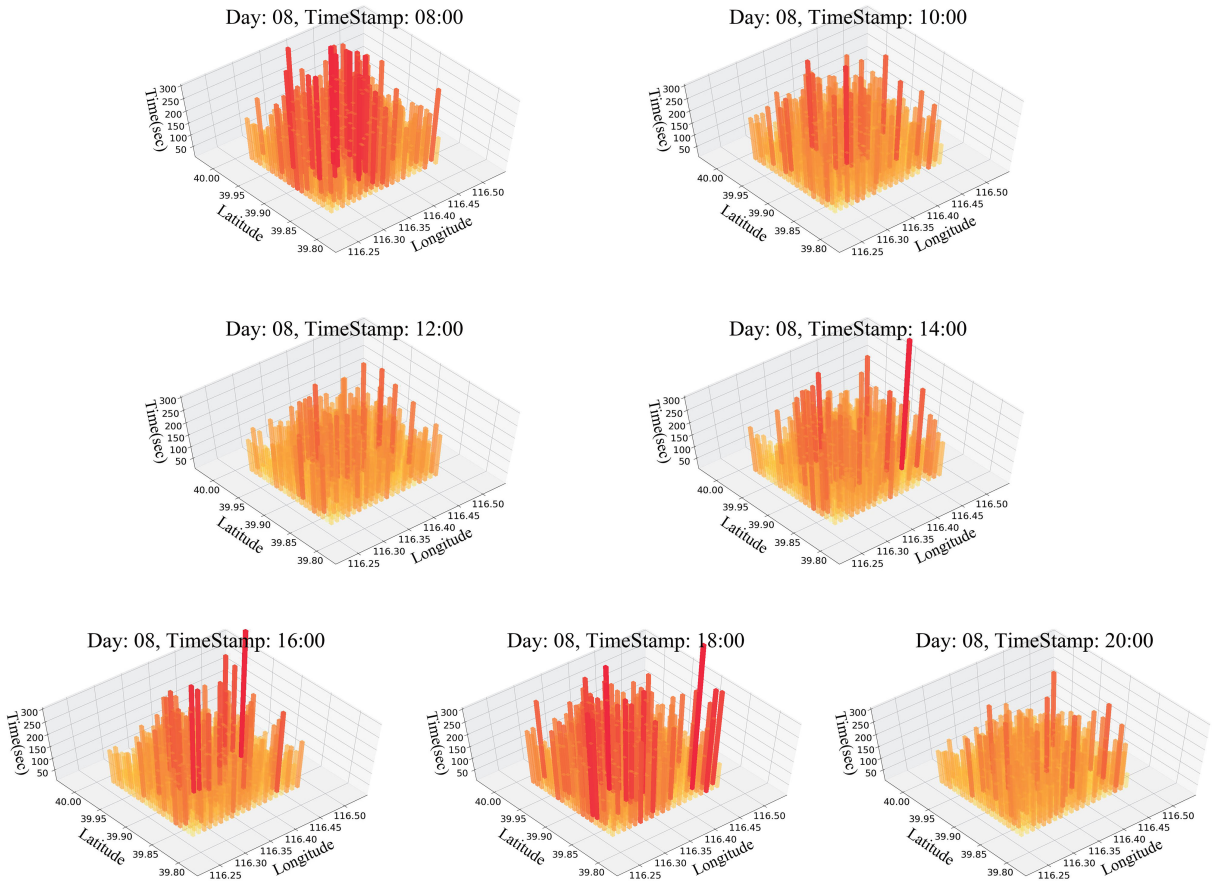



Fig. 4. The traverse time maps on November 8, 2012 from 8:00 to 20:00.

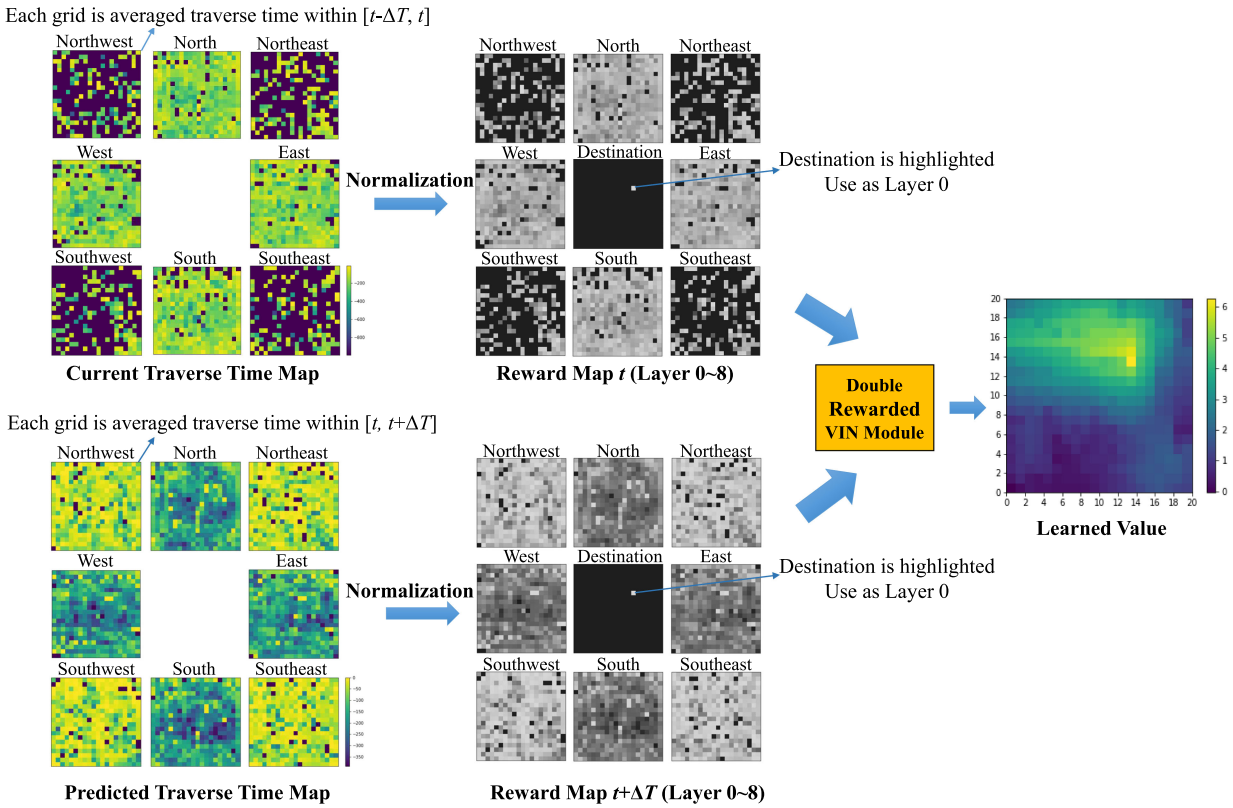



Fig. 5. Prepare reward map for VIN.

TABLE I
LSTM TRAINING PARAMETERS

| | |
|---|---|
| Learning rate | 1 |
| # layers | 2 |
| # steps | 4 |
| Hidden size | 4000 |
| First FCL size | 3500 |
| Second FCL size | 3200 |
| Third FCL size | 3200 |
| Keep probability | 1 |
| Learning rate decay | 0.93 |
| Batch size | 10 |
| Vocal size | 3200 |

### B. Prediction Module

For prediction module, besides LSTM network adapted in our prediction enabled double rewarded VIN model, Gaussian process regression method proposed in [16] is used to compare prediction results.

In prediction module, we first divide Beijing map into a $20 \times 20$ grid-world and compute TTMs at an interval of $\Delta T = 20$ minutes, so the input data fed into LSTM network are four $20 \times 20 \times 8$ tensors which represent TTMs. Then, LSTM network can generate a TTM for next 20 minutes. When input data tensors have passed prediction module, we can have two tensors which represent current and future TTMs. Training parameters for LSTM network are shown in Table I.

The data from 21 weekdays and 36 $\left(= \frac{12 \text{ Hours} \times 60}{20}\right)$ time slices per day can generate available data of 17 $(= 21 - 4)$ days (data of previous 4 days are used to predict the data of today). Training phase uses first 14 days for training and test phase uses last 3 days for prediction. The performance of LSTM and Gaussian process regression is compared using relative error

$$error = \frac{|\text{real time} - \text{predicted time}|}{\text{real time}}, \tag{13}$$

and loss (i.e., mean squared error), shown in Fig. 6 and Table II. It can be seen that LSTM performs much better than Gaussian process regression.

### C. VIN Module

For the double rewarded VIN module, the state space $\mathcal{S}$ is a $20 \times 20$ grid-world. The reward $R$ in this space can be represented by two $20 \times 20 \times 9$ TTMs including destination layer. The transition matrices $P$ are defined as $3 \times 3$ convolution kernels in value iteration model, exploiting the fact that transitions in grid-based map are local. The parameters of double rewarded VIN are shown in Table III.

### D. Generalizing Results

To have an intuitive sight for the route planning ability of the proposed model, some typical cases in Fig. 7 (a)–(d) are visualized to show generalizing results of our proposed model (LSTM-VIN) and real routes of experienced drivers. For LSTM-VIN, the trajectories in Fig. 7 (a)–(c) successfully reach destinations with varies route choices, but the trajectory in Fig. 7 d fails. The routing failure, which is repeatedly oscillating, is

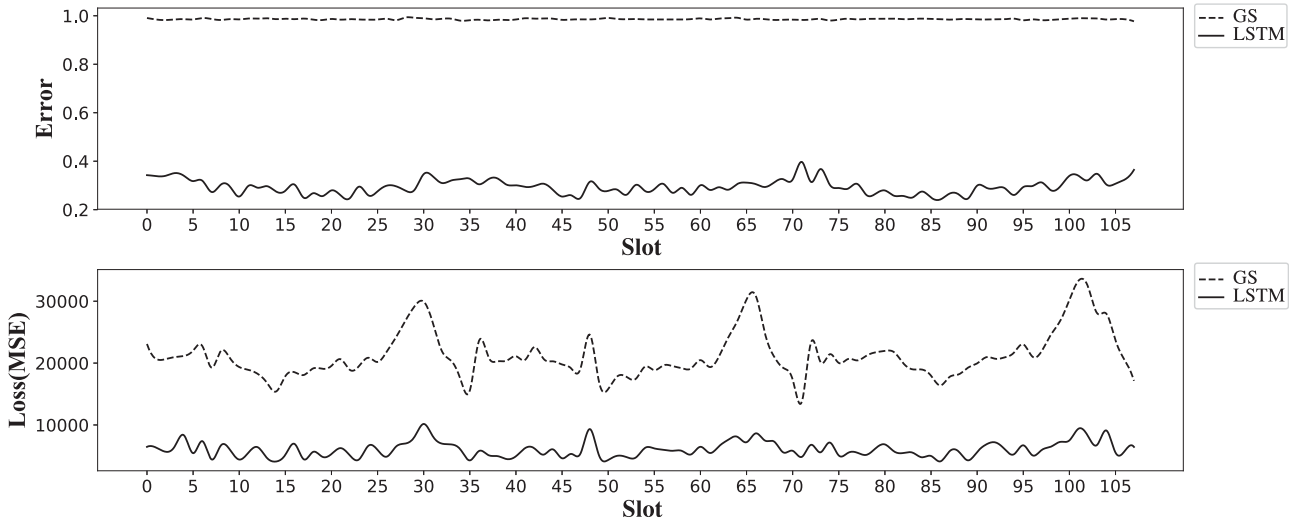



Fig. 6. Prediction module results.

TABLE II
RESULTS IN PREDICTION MODULE

| Method | Error | Loss |
|---|---|---|
| GS | 0.986 | 21174.8 |
| LSTM | 0.296 | 6052.0 |

TABLE III
VIN TRAINING PARAMETERS

| | |
|---|---|
| # value iterations | 30 |
| Convolution kernel size | 3×3 |
| # epoch | 200 |
| Batch size | 64 |
| Learning rate | 0.001 |
| Channels in input layer | 9 |
| Channels in Q layer (actions) | 8 |

mostly because of the imprecise value map derived from sparse trajectory data.

The experimental results show that LSTM-VIN can reach 50.9% top-1 accuracy, 73.0% top-2 accuracy, 81.1% success rate and 62.2% save time rate.

*Accuracy:* Top-1 accuracy and top-2 accuracy are used to measure experimental performance, where top-k accuracy means that the correct class is in the top-k probabilities for it can be regarded as "correct". The top-1 accuracy is not high because of intrinsic uncertainty of route planning. Even experienced drivers in the same place could choose different routes to the same destination. In Beijing, the main roads are usually in $east-west$ or $north-south$ direction, hence there are usually more than one route to the same destination. In other words, the existing Bayes error [37] of route planning makes it slightly difficult to get high top-1 accuracy. On the other hand, the high top-2 accuracy indicates that our LSTM-VIN has successfully learned most driving patterns considering alternative choices.

*Success rate:* Given a global status TTM, a full trail from initial state is predicted by iteratively choosing the optimal next states. A trail is successful if it reaches destination via the predicted trajectory. Success rate is the ratio of the number of successful trails to that of all trails. The high success rate demonstrates that LSTM-VIN is able to plan route in coarse-grained urban model.

*Saved time rate:* By randomly selecting real taxi trajectories from the dataset, we can build a sample set $\mathbb{T} = \{(O_1, D_1, time_1), (O_2, D_2, time_2), \ldots, (O_i, D_i, time_i), \ldots\}$, and use VIN to generate corresponding new trajectory set $\mathbb{T}' = \{(O_1, D_1, time'_1), (O_2, D_2, time'_2), \ldots, (O_i, D_i, time'_i), \ldots\}$. Note that we only consider successful trajectories. We give "TTM invariant assumption" that changing one's trajectory does not make difference on global TTM, so that TTM can be used to estimate total time-consuming of new trajectory. More specifically, applying VIN to a trajectory $\mathcal{T}_i = (O_i, D_i, time_i)$ would produce a new trajectory $\mathcal{T}'_i = (O_i, D_i, time'_i)$, which saves time $\Delta time_i = time_i - time'_i$. Saved time rate (STR) is calculated from whole sample set $\mathbb{T}$:

$$STR = \frac{\sum_{\mathcal{T}_i \in \mathbb{T}} \Delta \text{time}_i}{\sum_{\mathcal{T}_i \in \mathbb{T}} \text{time}_i}. \tag{14}$$

### E. Comparative Experiments

To demonstrate the performance of our LSTM-VIN, following three algorithms are compared.

*Dijkstra algorithm:* It uses TTM which represents current traffic status to generate routes for drivers with shortest time.

*Simple VIN network [13]:* It utilizes simple VIN without prediction module to generate time-saving routes. This method learns a policy function which maps current global traffic status to experienced drivers' actions.

*Gaussian regression algorithm [16]:* It uses Gaussian process regression to predict traffic status and then to generate routes. However, the authors mainly emphasized the prediction part without explaining how to plan routes. Therefore, we use VIN network to generate routes based on the predicted future traffic status.

Some generalizing routes are depicted in Fig 7 e-h. It can be seen that LSTM-VIN usually generates stable and reasonable

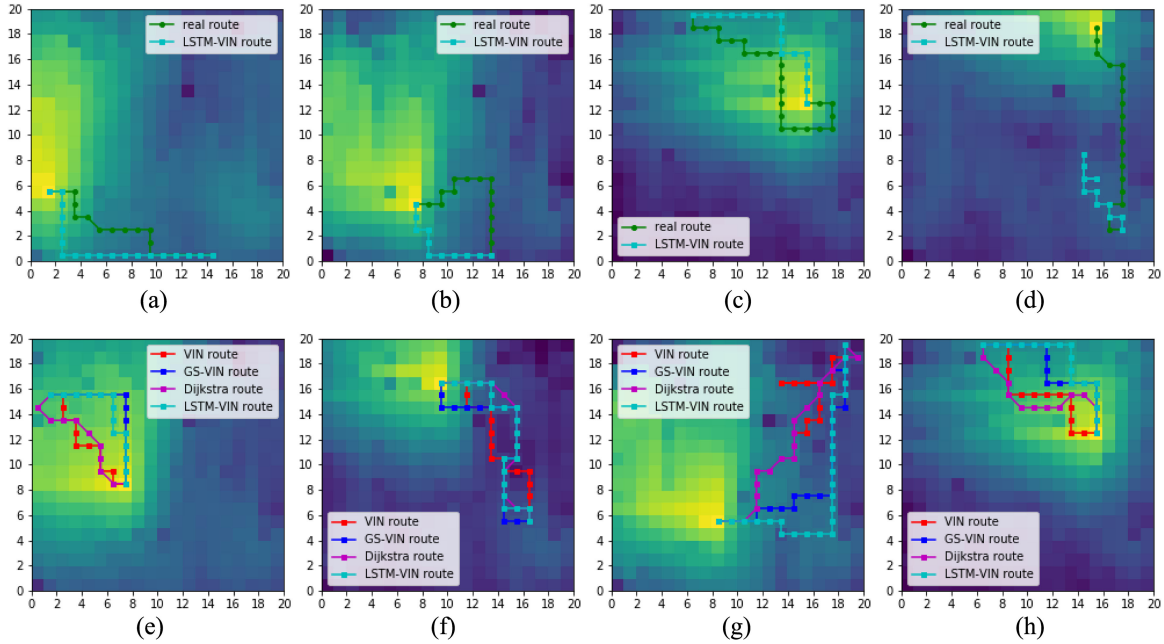



Fig. 7. Routes of LSTM-VIN and comparative experiments.

TABLE IV
PERFORMANCE OF LSTM-VIN AND COMPARATIVE EXPERIMENTS

| Method | Top-1 Accuracy | Top-2 Accuracy | Success Rate | Saved Time Rate (only for success trails) |
|---|---|---|---|---|
| LSTM-VIN | 50.9% | 73.0% | 99.6% | 62.2% |
| Simple-VIN | 49.5% | 72.1% | 98.4% | 60.8% |
| Dijkstra | - | - | 100.0% | 46.0% |
| GS-VIN | 50.3% | 72.8% | 99.7% | 61.6% |

routes, whereas the routes generated by simple VIN may oscillate (e.g., Fig. 7 g) due to the change of traffic status. Moreover, Dijkstra method cannot generate optimal time-saving routes, because it fails to take traffic trends into consideration. As for GS-VIN, it generates routes which are similar to LSTM-VIN because the only difference between them is traffic prediction method. Specifically, the numerical results are shown in Table IV. We can find that LSTM-VIN has the best performance for top-1 accuracy, top-2 accuracy and saved time rate. As for success rate, we have speculated the reason why LSTM-VIN works not so well. Though LSTM-VIN has adequate expressiveness and complexity, but we do not have sufficient data to train this network to learn all experienced drivers' patterns. In the future, we will collect more data to solve this problem.

## VII. CONCLUSION

In this paper, we have proposed a traffic prediction enabled double rewarded value iteration network to plan time-saving routes. An LSTM network is used to learn experienced drivers' knowledge of traffic trends, and a double rewarded VIN is designed to learn experienced drivers' routing behavior base on both current and future traffic status. Extensive experimental results, based on real taxicab mobility traces in Beijing, have demonstrated that our proposed model can achieve human-like performance with high success rate and less commuting time. To fully leverage the proposed model, further research issues are discussed as follows.

**Spatial correlation can be used for traffic prediction.** In this paper, according to the input data structure of LSTM network, we have simply transformed a three-dimensional tensor representing traverse time map to a one-dimensional vector, which loses spatial correlation among traffic status. In the future, we will add CNN layers before LSTM layers to capture the spatial correlation.

**The road topology can be well utilized.** To simplify the model, grid-based routes are utilized in this paper. However, vehicles in the real world do not move grid by grid, so these data cannot actually reflect drivers' routing behavior. We will reconstruct drivers' routes via road topology and utilize graph convolution to realize value iteration network.

## REFERENCES

[1] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao, "A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2551–2566, Mar. 2017.

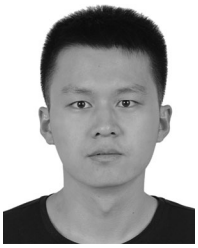
[2] M. A. S. Kamal, T. Hayakawa, and J. Imura, "Road-speed profile for enhanced perception of traffic conditions in a partially connected vehicle environment," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6824–6837, Aug. 2018.

[3] G. Luo *et al.*, "Cooperative vehicular content distribution in edge computing assisted 5G-VANET," *China Commun.*, vol. 15, no. 7, pp. 1–17, Jul. 2018.
[4] Q. Yuan, H. Zhou, Z. Liu, J. Li, F. Yang, and X. Shen, "CE-Sense: Cost-effective urban environment sensing in vehicular sensor networks," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2018.2873112.
[5] J. Wan, D. Zhang, S. Zhao, L. T. Yang, and J. Lloret, "Context-aware vehicular cyber-physical systems with cloud support: Architecture, challenges, and solutions," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 106–113, Aug. 2014.
[6] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.
[7] S. Wang, J. Xu, N. Zhang, and Y. Liu, "A survey on service migration in mobile edge computing," *IEEE Access*, vol. 6, pp. 23511–23528, 2018.
[8] J. Xie *et al.*, "A survey on machine learning-based mobile big data analysis: Challenges and applications," *Wireless Commun. Mob. Comput.*, vol. 2018, 2018, Art. no. 8738613.
[9] J. Li *et al.*, "An end-to-end load balancer based on deep learning for vehicular network traffic control," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 953–966, Feb. 2019.
[10] A. L. Kok, E. W. Hans, and J. M. J. Schutten, "Vehicle routing under time-dependent travel times: The impact of congestion avoidance," *Comput. Operations Res.*, vol. 39, no. 5, pp. 910–918, 2012.
[11] J. L. Adler and V. J. Blue, "A cooperative multi-agent transportation management and route guidance system," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 5/6, pp. 433–454, Oct.–Dec. 2002.
[12] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, "Real-time path planning based on hybrid-VANET-enhanced transportation system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1664–1678, May 2015.
[13] S. Yang, J. Li, J. Wang, Z. Liu, and F. Yang, "Learning urban navigation via value iteration network," in *Proc. IEEE Intell. Veh. Symp.*, Changshu, China, 2018, pp. 800–805.
[14] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transp. Res. Rec.*, no. 773, pp. 47–49, 1980.
[15] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, 1995.
[16] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik, "Dynamic route planning with real-time traffic predictions," *Inf. Syst.*, vol. 64, pp. 258–265, Mar. 2017.
[17] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.
[18] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.
[19] 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. 16, Apr. 2017, Art. no. 818.
[20] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.
[21] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, Nov. 2015.
[22] T. Yamashita, K. Izumi, and K. Kurumatani, "Car navigation with route information sharing for improvement of traffic efficiency," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, Washington, DC, USA, 2004, pp. 465–470.
[23] D. Bucher, D. Jonietz, and M. Raubal, "A heuristic for multi-modal route planning," in *Proc. Prog. Location Based Sevices*, Vienna, Austria, 2016, pp. 211–229.
[24] M. Dotoli, H. Zgaya, C. Russo, and S. Hammadi, "A multi-agent advanced traveler information system for optimal trip planning in a co-modal framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2397–2412, Sep. 2017.
[25] Z. Wang and S. Zlatanova, "Multi-agent based path planning for first responders among moving obstacles," *Comput. Environ. Urban Syst.*, vol. 56, pp. 48–58, Mar. 2016.
[26] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transp. Res. C, Emerg. Technol.*, vol. 71, pp. 143–163, Oct. 2016.
[27] K. Sundar, S. Venkatachalam, and S. Rathinam, "Analysis of mixed-integer linear programming formulations for a fuel-constrained multiple vehicle routing problem," *Unmanned Syst.*, vol. 5, no. 4, pp. 197–207, Aug. 2017.
[28] H. Hiraishi, "Route-planning based on a passenger condition for self-driving vehicles," in *Proc. IEEE 16th Int. Conf. Cogn. Inform. Cogn. Comput.*, Oxford, U.K., 2017, pp. 329–334.
[29] M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takáč, "Deep reinforcement learning for solving the vehicle routing problem," 2018. [Online]. Available: http://arxiv.org/abs/1802.04240v2
[30] M. Zolfpour-Arokhlo, A. Selamat, S. Z. M. Hashim, and H. Afkhami, "Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms," *Eng. Appl. Artif. Intell.*, vol. 29, pp. 163–177, Mar. 2014.
[31] S. Brahmbhatt and J. Hays, "DeepNav: Learning to navigate large cities," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 3087–3096.
[32] K. Lin, C. Li, G. Fortino, and J. J. P. C. Rodrigues, "Vehicle route selection based on game evolution in social internet of vehicles," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2423–2430, Aug. 2018.
[33] J. Liu and S. Ma, "Analysis of evolutionary game about the route choice of individual travel mode based on bounded rationality," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Maui, Hawaii, USA, 2008, pp. 622–626.
[34] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, "Variational Bayesian matrix factorization for bounded support data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 876–889, Apr. 2015.
[35] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 1054–1054, Sep. 1998.
[36] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel, "Value iteration networks," in *Proc. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 2154–2162.
[37] S. H. Yang and B. Hu, "Discriminative feature selection by nonparametric Bayes error minimization," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1422–1434, Aug. 2012.

**Jinglin Li** received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is currently an Associate Professor with the State Key Laboratory of Networking and Switching Technology, BUPT. His research interests are mainly in the areas of network intelligence, mobile Internet, the Internet of Things, and the Internet of Vehicles.

**Dawei Fu** is currently working toward the M.S. degree in computer science and technology with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include the Internet of Vehicle, mobile Internet, and intelligent transportation system.

**Quan Yuan** received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2018. He is currently working as a Postdoctoral Fellow with the State Key Laboratory of Networking and Switching Technology, BUPT. His current research interests include crowdsensing, connected vehicle, mobile Internet, and intelligent transportation system.



**Haohan Zhang** is currently working toward the M.S. degree in computer science and technology with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include Internet of Vehicle, mobile Internet, and intelligent transportation system.

**Kaihui Chen** received the B.S. degree from Yanshan University, Qinhuangdao, China, in 2017. He is currently working toward the M.S. degree with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests are in the areas of deep learning, intelligent city, and computer vision.

**Shu Yang** received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China. He is currently a R&D with Baidu, IVBU, Beijing, China. His research interests include autonomous driving and ITS.

**Fangchun Yang** received the Ph.D. degree in communications and electronic systems from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, BUPT. His current research interests include network intelligence, service computing, and the Internet of Vehicles. He is a fellow of the IET.