Research Article

An Online-Traffic-Prediction Based Route Finding Mechanism for Smart City

Xiaoguang Niu, Ying Zhu, Qingqing Cao, Xining Zhang, Wei Xie, and Kun Zheng

1School of Computer Science, Wuhan University, Wuhan 430000, China
2Computer School, Central China Normal University, Wuhan 430000, China
3Faculty of Information Engineering, China University of Geosciences, Wuhan 430000, China

Correspondence should be addressed to Xiaoguang Niu; xgniu@whu.edu.cn and Wei Xie; xw@ccnu.edu.cn

Received 23 October 2014; Revised 28 January 2015; Accepted 18 February 2015

Copyright © 2015 Xiaoguang Niu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Finding fastest driving routes is significant for the intelligent transportation system. While predicting the online traffic conditions of road segments entails a variety of challenges, it contributes much to travel time prediction accuracy. In this paper, we propose O-Sense, an innovative online-traffic-prediction based route finding mechanism, which organically utilizes large scale taxi GPS traces and environmental information. O-Sense firstly exploits a deep learning approach to process spatial and temporal taxi GPS traces shown in dynamic patterns. Meanwhile, we model the traffic flow state for a given road segment using a linear-chain conditional random field (CRF), a technique that well forecasts the temporal transformation if provided with further supplementary environmental resources. O-Sense then fuses previously obtained outputs with a dynamic weighted classifier and generates a better traffic condition vector for each road segment at different prediction time. Finally, we perform online route computing to find the fastest path connecting consecutive road segments in the route based on the vectors. Experimental results show that O-Sense can estimate the travel time for driving routes more accurately.

1. Introduction

With the development of intelligent transportation technology in smart cities, finding fast driving routes can be widely used for traffic flow coordinating [1] to optimize the plan of traffic management and urban computing [2–5]. It is beneficial to find fast driving routes, which is conducive to energy saving and traffic congestion coordination.

The existing fastest routes finding approaches usually look into the real-time traffic conditions or infer potential travel costs of road through mining historical trajectory data. However, these works mostly assume the static travel cost on each road, while ignoring the potential temporal dynamics. In fact, the driving time, calculated from the current travel time of each road segment, deviates much from the truth. The soundness of fastest route finding relates greatly to the prediction accuracy of travel time for vehicles on the driving road. Hence, it is necessary to predict the traffic conditions of road segments at an appropriate future moment.

The urban traffic prediction usually utilizes historical and current traffic flow information to predict road conditions for future moments [6]. Most existing methods present prediction trends via statistics of the current road time-dependent evolution or they only work on spatial relationships between various segments without consideration of the temporal sequence and temporal-spatial information rules. Although available spatial-temporal information is combined to model the traffic network pattern in some previous works, this information does not play out its full potential. Traffic networks involve complicated relations in time and space. Specifically, traffic flow often contains high-dimensional, nonlinear, and nonstationary random data. In a previous work [7], a trained temporal-spatial deep learning approach DeepSense was proposed, which extracted features from the high-dimensional, nonlinear, and random traffic flow in large scale taxi GPS traces. However, we found that the prediction does not work so well at certain times, especially when there are insufficient taxi GPS traces. Based on this finding, it
is proper to consider more factors such as environmental information containing temporal changes that influence the traffic flow of current road.

In this paper, we propose an online-traffic-prediction based route finding mechanism for smart city, namely, O-Sense. It comprehensively utilizes both temporal-spatial dynamic pattern in transportation network and temporally related environmental information to predict the travel cost of each route. Firstly, we analyze temporal and spatial information of taxi GPS traces via Restricted Boltzmann Machine (RBM) since it can process this information into low-dimensional features. Then the extracted low-dimensional data is put into a support vector machine (SVM) to gain a robust classifier that makes promise for the following prediction. Secondly, a linear-chain CRF predicts the future traffic state by giving the previous state in a temporal sequence and supplementary temporally related observations, since CRF is well known for being applied to mutually independent road segments. With auxiliary data utilized by CRF, prediction accuracy can be further enhanced through a dynamic weighted classifier fusion. Finally, using the results calculated from the predicted traffic conditions of each road, our system can perform the fast route selection for users.

The major contributions of this paper can be summarized into the following aspects.

1. We propose an online-traffic-prediction based route finding mechanism O-Sense, which innovatively adopts a trained temporal-spatial deep learning approach. Significant traffic features can be extracted from the information synthesized from high-dimensional, nonlinear, and random large scale taxi GPS traces.

2. Combining temporally related environmental resources which are utilized in CRF as supplementary and dynamic weighted classifier fusion, the prediction accuracy can be further improved. As a result, taking full advantage of temporal and spatial information for traffic forecasting, it can effectively reflect the dynamic pattern shown in transportation network.

3. Utilizing large scale taxi GPS traces and environmental information in Wuhan, we evaluate the mechanism from a systematic perspective. Experimental results manifest that O-Sense mechanism can achieve sound and robust performance in prediction accuracy.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 presents the proposed O-Sense mechanism. Section 4 describes the processes of the traffic flow condition prediction. The performance of O-Sense is evaluated in Section 5. Finally, we make a conclusion in Section 6.

2. Related Works

2.1. Route Finding. Many works have been proposed for the fastest route finding problem, most of which can be divided into two categories based on the characteristics of the travel cost of road segments. On one hand, some works infer potential travel costs of road segments by exploiting historical trajectory data [8]. In [9], Tsuyoshi and Sugiyama treat the potential cost of each road as a spatial proximity-regularized trajectory regression problem. Gonzalez et al. find the fastest route based on the important driving and speed patterns which are learned from the historical trajectory data [10]. However, these works with the assumption of static travel costs cannot reflect the dynamic variances according to the actual traffic conditions. On the other hand, some works take the temporal dynamics of road travel costs into consideration to compute the fastest route [11]. Zheng and Ni learn the temporal dynamics of road travel costs, while it depends on integrated trajectories which cannot provide enough information when there is insufficient traffic information [12]. However, the pattern of the temporal dynamics of latent costs for road segments is hard to be predicted, especially in the cases with insufficient trajectory data.

2.2. Traffic Flow Condition Prediction. Most of the existing traffic flow condition prediction approaches merely utilize time-dependent traffic flow evolution rule for prediction. In [13], the autoregressive integrated moving average model (ARIMA) simply relies on historical traffic flow data of forecasted point taking temporal variation into consideration for prediction. The traffic flow prediction model based on $m_{th}$-order Markov chain counts the transition probability of traffic flow state with historical data of forecasted road [14]. Coupled with current traffic situation and state transition probability, traffic condition in a period of time can be predicted. However, by using historical and real-time traffic patterns, the establishment of model in time sequence and the forecast duration is prolonged to some extent. Thus, there exists a lot of uncertainty and it is a statistical method. The variance-entropy-based clustering approaches [15] are used to estimate the travel time distribution in different time slots with historical data of object sites, which simply applies the idea of clustering analysis and generality.

There are also other methods that use spatial information in the transportation network to analyze the trend of traffic flow. Castro et al. [16] employ adjacent road information to forecast future traffic conditions, which does not take into account the influence of distant segments and temporal-spatial correlation of predicted road. Markov logic networks [17] have also been adopted to predict the traffic conditions at simultaneous locations in different future time.

The methods mentioned above merely model traffic flow trends with respect to one aspect. Temporal-spatial information has not been made use of to depict the characteristics of nonlinearity and randomness in the flow from an angle of the whole network. A Bayesian network approach [18] intended for traffic flow forecast integrates information of adjacent links and its spatial-temporal information in a transportation network. Sun and Zhang develop a selective random subspace predictor (SRSP) model [19] utilizing traffic flows of some most closely correlated links ranked by the measurement of Pearson correlation coefficient in the subspace to forecast...
the given road link. Although considering spatial-temporal information, these approaches did not extract the characteristics of high-dimension, nonlinearity, and randomness effectively.

3. O-Sense Overview

3.1. Preliminary

**Definition 1** (route). A route $R$ is a group of consecutive road segments, $R: r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_n$, where $r_i$ is the $i$th road segment ($1 \leq i \leq n$). In route $R$, the end point of one road segment is the start point of its directly succeeding road segment; that is, $r_i$.end = $r_{i+1}$.start.

**Definition 2** (taxi trajectory). A trajectory is a time series of GPS points for a trip, where there is a geospatial coordinate set and a timestamp at each consecutive point.

**Definition 3** (road segment/link). A road link or segment is defined as a directed edge consisting of a direction symbol, two terminal points, and a length between crossroads.

**Definition 4** (traffic flow condition). We select traffic flow to denote traffic state of road segments. Given different speed situations and flow limits on distinct road segments, classified traffic state based on absolute speed is obviously inaccurate. According to the degree of traffic congestion, we can categorize the traffic conditions into five states, represented by $\Omega = \{\text{congesting (Cg)}, \ \text{slow (Sl)}, \ \text{normal (Nm)}, \ \text{moderate (Md)}, \ \text{unimpeded (Un)})\}$.

In detail, for the road segment $R$ from 0 o'clock to 24 o'clock, a traffic flow $o_i$ is collected every time interval $\Delta t$ (e.g., 15 minutes), thus obtaining a series of traffic flow observations $O$ (e.g., $o_1, o_2, o_3, \ldots, o_{96}$). In order to categorize these series to prementioned five states, we adopted $K$-means algorithm to cluster them into 5 subsets. By utilizing the average values of these 5 subsets, we can acquire their corresponding traffic states.

3.2. Framework. In order to learn online road travel costs for selecting the fastest routes for users, we propose an online-traffic-prediction based route finding mechanism, namely, O-Sense. As shown in Figure 1, to implement our mechanism for fastest route finding entails three procedures, namely, preprocessing, comprehensive feature learning, and route computing.

3.2.1. Preprocessing. Spatial trajectories collected by GPS-equipped vehicles are mapped onto a road network using a map-matching algorithm [20] and then stored into a taxi traces database. Environmental information is extracted to be used as temporally related features, as described in detail in Section 4.

3.2.2. Comprehensive Feature Learning. This process contains three parts: temporal-spatial feature deep learning, temporally related supplementary feature learning, and dynamic weighted classifier fusion. Firstly, to learn temporal-spatial features, we apply deep learning approach with preextracted temporal-spatial traffic flow information from taxi traces database. The trained temporal-spatial features are used to train RBM model, which is capable of predicting irregular and stochastic factors in traffic systems. By means of this trained model, the dimensionality and redundancy of the input features are reduced to a proper extent. The extracted features can be more effectively classified by a classification engine SVM. This engine is designed to utilize the extracted
Input: A directed graph $G = \{V, E\}, V = \{1, 2, \ldots, n\}$, Departure time $t$

Output: The shortest distance from 1 to each node in $G$

Procedure:
1. $X = \{1\}; Y = V - \{1\}; \lambda[1] = 0$
2. for $x = 1$ to $n - 1$
3. for $y = 1$ to $n$
4. if $x$ is adjacent to $y$ then $r_{xy}[60] = \{r_{xy}[t], r_{xy}[t + 2], \ldots, r_{xy}[t + 120]\}$
5. end if;
6. end for;
7. end for;
8. for $y = 2$ to $n$
9. if $y$ is adjacent to 1 then $\lambda[y] = r_{1y}[t]$;
10. else $\lambda[y] = \infty$;
11. end if;
12. end for;
13. $u = 1$;
14. for $j = 1$ to $n - 1$
15. $y \in Y, \lambda[y]$ is the least among all values;
16. $X = X \cup \{y\}$;
17. $Y = Y - \{y\}$;
18. for every edge $(y, w)$
19. if $w \in Y$ and $\lambda[y] + r_{wy}[t + r_{wy}[t]] < \lambda[w]$ then
20. $\lambda[w] = \lambda[y] + r_{wy}[t + r_{wy}[t]]$;
21. end if;
22. end for;
23. $t = t + r_{wy}[t]$;
24. $u = y$;
25. end for;

Algorithm 1: Online route computing algorithm.

features to better train the expected prediction. Secondly, in order to learn supplementary temporally related features, CRF approach is trained to estimate the temporal state sequence of traffic flow with extracted environmental information. Therefore, the temporal state sequences obtained from CRF are matched for future traffic state prediction. The last technique adopted is dynamic weighted classifier fusion for comprehensively learning the whole temporal-spatial information and temporally related features. After real-time traffic flow data is preprocessed and temporally related features of forecasted time are preextracted, we can get an eventual prediction result. Details are presented in Section 4.

3.2.3. Online Route Computing. Given the prediction of each road segment at disparate time, a detailed route computing approach is performed in O-Sense to seek an optimal route. This approach utilizes the traffic condition of road segments in real-time for route computing. The process of online route computing is illustrated in Algorithm 1. In this paper, we consider the transportation networks exhibit the "FIFO" (First-In-First-Out) property; that is, if A and B visit node $n1$ at two different times $t1$ and $t2$, where $t1 \leq t2$, then A always arrives at node $n2$ before B for any arc $(n1, n2)$. Our solution is composed of the following steps.

Step 1. To find the fastest route with giving a starting point, a destination, and departure time, O-Sense firstly continually predicts a group of traffic states for each road segment that connects two points every two minutes for the following specific time period (e.g., for the following two hours). As the traffic condition division is based on specific clustering method, each traffic condition state takes a speed value as the clustering center. Hence, we can utilize five speed values to represent five states and they form a speed vector. Utilizing the speed vector to represent predicted traffic states, the weights of each road segment at different moments are obtained.

Step 2. O-Sense chooses the optimal predicted value from the predicted speed vector of each road segment dynamically as the weight of corresponding road segment, which is closest to the actual value according to the time when the user arrives at the road segment. We can find the time-dependent fastest route using a modified Dijkstra algorithm [21, 22].

The fastest path connecting consecutive road segments is eventually found as we arrive at the destination. Owning to the different time costs of road segments, if we start at different time at the starting of the route even though given the same start point and destination, we may find disparate fast routes.
Example 1. Seen from Figure 2, if starting at current time $t_{ct}$ from road intersection A, the road segment $r_{ab}$, of which the time cost $r_{ab}(t_{ct})$ is the least in all road segments connecting A is chosen. Then the label of other nodes in the graph is updated, which denotes the shortest distance from the departure point to each node. When arriving at B, the time stamp at that time is $t_{ct} + r_{ab}(t_{ct})$. We compare the label of nodes C and E, that is, the travel time of going through $r_{bc}$ and $r_{be}$ ($r_{bc}[t_{ct} + r_{ab}(t_{ct})] < r_{be}[t_{ct} + r_{ab}(t_{ct})]$) in this example. Then turn to E with updating the labels. Keeping searching like this, the fastest route from A to G starting at current time $t_{ct}$ is $A \rightarrow B \rightarrow E \rightarrow F \rightarrow G$. However, if starting at $t_{ct} + t_{lr}$, where $t_{lr}$ denotes some time later than $t_{ct}$, the route $A \rightarrow B \rightarrow C \rightarrow F \rightarrow G$ now becomes the fastest rough route since when we arrive at B, the time stamp at that time is becoming $t_{ct} + t_{lr} + r_{ab}[t_{ct} + t_{lr}]$ and the travel time of going through $r_{bc}$ is larger than that of $r_{be}$; that is, $r_{bc}[t_{ct} + t_{lr} + r_{ab}[t_{ct} + t_{lr}]] < r_{be}[t_{ct} + t_{lr} + r_{ab}[t_{ct} + t_{lr}]]$. Hence, our approach utilizes the traffic condition of road segments in real-time (when actually driven) for route computing.

4. Comprehensive Feature Learning Based Prediction

This section details the methodology of the comprehensive feature learning based online traffic condition prediction mechanism, which consists of temporal-spatial feature deep learning, temporally related supplementary feature learning, and dynamic weighted classifier fusion-based comprehensive prediction.

4.1. Temporal-Spatial Traffic Feature Deep Learning. To simplify and reduce the dimension of the input data in order to efficiently learn the significant traffic features, O-Sense employs the temporal-spatial feature deep learning approach which includes three main steps: feature preextraction, feature learning, and training a SVM model.

4.1.1. Feature Preextraction. The correlation coefficient approach is first used to select the most correlated road segments between adjacent road segments and the predicted point, and then extracted features of these road segments are input into PCA to reduce data dimension and remove redundant information for better prediction.

Correlation Coefficient Approach. Considering $N$ samples $\{x_i(k), y_j(t_{ct}+t_i)\}$ ($i, j = 1, 2, \ldots, N$) in a spatial accessible space, $x_i(k)$ ($k = 1, 2, \ldots, s$) represents the traffic flow of road segment $i$ at time stamp $k$ and $\overline{x}$ represents the mean value. And $y_j(t_{ct}+t_i)$ stands for the traffic flow of road segment $j$ at predicted time ($t_{ct}+t_i$) and $\overline{y}$ stands for the mean value, where $t_{ct}$ is the current time and $t_i$ means the duration of the prediction. The Pearson correlation coefficient $r_{ij}(k)$ between $x_i(k)$ and $y_j(t_{ct}+t_i)$ is defined as follows:

$$
r_{ij}(k) = \frac{\sum_{i=1}^{N} (x_i(k) - \overline{x}) (y_j(t_{ct}+t_i) - \overline{y})}{\sqrt{\sum_{i=1}^{N} (x_i(k) - \overline{x})^2} \sum_{i=1}^{N} (y_j(t_{ct}+t_i) - \overline{y})^2}
= \left( N \sum_{i=1}^{N} x_i(k) y_j(t_{ct}+t_i) - \sum_{i=1}^{N} x_i(k) \sum_{i=1}^{N} y_j(t_{ct}+t_i) \right)
\sqrt{N \sum_{i=1}^{N} x_i(k)^2 \sum_{i=1}^{N} y_j(t_{ct}+t_i)^2 - \left( \sum_{i=1}^{N} x_i(k) \sum_{i=1}^{N} y_j(t_{ct}+t_i) \right)^2}
$$

Figure 2: Example of online route computing.
constructed as follows: the covariance matrix of traffic flow in road segments can be contributed by each eigenvalue is only needed to extract part of the main components. The contribution rate of each eigenvalue is

\[ \Delta \begin{pmatrix} \mathbf{x}_1(k) \\
\mathbf{x}_2(k) \end{pmatrix} = \mathbf{v}_1 + \mathbf{v}_2 \]

where \( \mathbf{v}_1 \) and \( \mathbf{v}_2 \) are the first and second components, respectively.

It means higher correlation if the absolute value of spatial and temporal correlation coefficient is closer to 1. Then information of the most \( m \) correlated road segments and corresponding time stamp will be chosen as extracted features, that is, six parameters (\( \mathbf{F}_m, \mathbf{c}_m, \Delta_{mx}, \mathbf{d}_m, \mathbf{w}_m, \mathbf{l}_m \)). \( \mathbf{F}_m, \mathbf{c}_m \) denote the average speed and the certain state of the \( m \)th road segment at the corresponding time stamp. \( \Delta_{mx} \) and \( \mathbf{d}_m \) denote the interval of the time stamp and geodistance between the predicted road segment and its \( m \)th most correlated road segment, respectively. \( \mathbf{w}_m \) and \( \mathbf{l}_m \) denote the width and length of the \( m \)th most correlated road segment.

**Principle Component Analysis.** After extracting features by correlation coefficient approach, Principle Component Analysis (PCA) is adopted to reduce data dimension and compress the data volume efficiently. The features of \( m \) most correlated road segments construct the original data matrix. Consider

\[ \mathbf{C} = \begin{bmatrix} c(1) & c(2) & \cdots & c(6) \end{bmatrix}^T \in \mathbb{R}^{6 \times m}, \]

where \( c(i) = [c_1(i) c_2(i) \cdots c_6(i)] \in \mathbb{R}^m \). In order to overcome such problems as the disunity of the original data, the raw data is normalized as follows:

\[ x_j(i) = \frac{c_j(i) - e(c_j)}{\delta(c_j)} \quad \forall i, j, \]

\[ \mathbf{X} = [\mathbf{x}(1) \mathbf{x}(2) \cdots \mathbf{x}(6)]^T \in \mathbb{R}^{6 \times m}, \]

where \( \delta^2(c_j) = (1/5) \sum_i^6 (c_j(i) - e(c_j))^2 \) is the sample variance and \( e(c_j) = (1/6) \sum_i^6 c_j(i) \) is the sample mean. Then the covariance matrix of traffic flow in road segments can be constructed as follows:

\[ \mathbf{S} = \frac{1}{m-1} \sum_j \left( \mathbf{x}_j - \langle \mathbf{x} \rangle \right) \left( \mathbf{x}_j - \langle \mathbf{x} \rangle \right)^T, \]

where \( \langle \mathbf{x} \rangle = (1/m) \sum_j \mathbf{x}_j \). All eigenvalues of the covariance matrix \( \mathbf{R} \) are found through the matrix computation and the sequence is \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_6 \geq 0 \).

As principle components contain most information, we only need to extract part of the main components. The contribution rate of each eigenvalue is

\[ \frac{\lambda_i}{\sum_{k=1}^6 \lambda_k} = \frac{\lambda_i}{6}. \]

The cumulative contribution rate of the preceding \( r \) principle components is computed as \( \sum \alpha_r \). When this contribution rate is larger than a constant value \( \psi \), we only need to extract \( r \) principle components which can reflect main information of all variables. Consider

\[ \sum \alpha_r = \frac{\sum_{k=1}^r \lambda_k}{\sum_{k=1}^6 \lambda_k}. \]

Finally the sample can be projected on the selected \( r \) feature vectors and a \( r \times m \) feature matrix is obtained for feature learning.

4.1.2. Feature Learning with RBM. The main idea of deep learning approach RBM in O-Sense is that a plenty of preextracted temporal-spatial traffic data are processed into informative low-dimensional features, and then these learned features are used to train an efficient classifier SVM.

RBM is a generative stochastic neural network that can learn a probability distribution (e.g., \( p(v, h) \)) over the input set, which is good at reasoning about and predicting irregular and stochastic behavior in the traffic flow. As shown in Figure 3, RBM is in the shape of a bipartite graph with no intralayer connections. The hidden unit activations \( h \) (low-dimensional data) are mutually independent given the visible unit activations \( v \); that is, when constraints are given on \( v \), all hidden units are conditionally independent; that is, \( p(h \mid v) = p(h_1 \mid v) \cdots p(h_6 \mid v) \), and vice versa. Given visible layer \( v \), hidden layer \( h \) can be obtained through \( p(h \mid v) \); meanwhile the value of units is gained by getting the hidden units \( p(v \mid h) \). Through adjusting the parameters, the visible layer \( v \) obtained from hidden layer can approximately equal the original input layer \( v \). From this perspective, the outputs of hidden units are another representation of the visible units; namely, the original high-dimensional traffic information has been transformed into low-dimensional representational data.

The standard type of RBM has binary-valued hidden and visible units and consists of a matrix of weights in which \( w_{ij} \) denotes the connection weight between hidden unit \( h_i \) and visible unit \( v_j \). \( a_i \) denotes the bias of \( v_i \), and \( b_j \) denotes the bias of \( h_j \). Given parameters \( \theta = (w_{ij}, a_i, b_j) \), the energy of a configuration \( (v, h) \) is defined as

\[ E(v, h; \theta) = -\sum_{j=1}^m a_j v_j - \sum_{j=1}^m b_j h_j - \sum_{i=1}^6 \sum_{j=1}^m h_j w_{ij} v_j. \]

This energy function is analogous to that of a Hopfield network [23]. As in general Boltzmann machines, probability distributions over hidden and visible vectors are defined in terms of the energy function. Consider

\[ P(v, h) = \frac{1}{Z} e^{-E(v, h)}, \]

where \( Z \) is a partition function defined as the sum of \( e^{-E(v, h)} \) over all possible configurations. Similarly, the marginal probability of a visible vector of Booleans is the sum over all possible hidden layer configurations. Consider

\[ P(v) = \frac{1}{Z} \sum_h e^{-E(v, h)}. \]
Then the probability under the condition that $h_j$ equals 1 and the probability under the condition that $v_i$ equals 1 are given as follows:

$$P(h_j = 1 | v_i; \theta) = \sigma\left(b_j + \sum_{i=1}^{m} w_{ij} v_i\right),$$  

$$P(v_i = 1 | h_i; \theta) = \sigma\left(a_i + \sum_{j=1}^{a} w_{ij} h_j\right),$$  

where $\sigma(x) = 1/(1 + e^{-x})$ denotes the logistic sigmoid function and the parameters $\theta = (w_{ij}, a_i, b_j)$ can be learned by gradient-based contrastive divergence algorithm [24].

4.1.3. Training a SVM Model. After input data is compressed by RBM, more representative features are extracted as the inputs of SVM for better classification effect. Through training a SVM classifier, we can get 5 classifications of traffic flow state and their probabilities construct a vector. Consider

$$P(t) = \left[ p \left( \text{state} = Cg \right), p \left( \text{state} = Sl \right), p \left( \text{state} = Nm \right), p \left( \text{state} = Mo \right), p \left( \text{state} = Un \right) \right].$$  

4.2. Temporally Related Supplementary Feature Learning. Observing from the data distribution, there are enough taxi GPS traces at day, which is, however, insufficient at midnight. When traffic information is insufficient at some certain times, the deep learning approach which only utilizes big data for traffic condition prediction cannot make enough effect [25]. Through supplementary feature learning, environmental information relevant to traffic condition for each road segment can be well taken into account and then make the prediction become more profound and accurate. This approach basically involves three main processes, environmental feature extraction, CRF real-time estimation, and sequence segments matching.

4.2.1. Environmental Feature Extraction. The traffic condition can be reflected by environmental information. Apparently, most of the road noise is emitted by vehicles. The observed patterns of noise can reflect under which traffic conditions it was produced [26, 27]. A high rainfall intensity also influences the traffic condition [28]. Owning to the reduced visibility and pavement friction, the effective travel times are greater than the mean travel time as expected. In addition, the rainfall intensity will also influence the travelers' path choice. If the effective travel time on one path is less sensitive to the rainfall as compared to other paths, most travelers choose or switch to the path with the probability of a higher rainfall intensity. A high wind speed results in a more hazardous trip and travel times are longer and less reliable, which leads to some travelers deferring trips or cancelling them [29]. The impact of temperature and PM2.5 is not very clear, but a good traffic condition is more likely when the temperature is high and PM2.5 is low.

When there is little traffic information at midnight, environmental information especially road noise can well reflect traffic condition. Road noise is mainly affected by traffic flow at night, which is not the same as during the day that contains interference factors. Hence, the sparse period of traffic information especially at midnight is quite suitable for being supplied with environmental information.

We select some environmental features that related to the traffic, road noise, temperature, wind speed, PM2.5, and rainfall denoted by a 5-dimensional vector $E = [e_1, e_2, e_3, e_4, e_5]$, where $e_i (i = 1, 2, \ldots, 4, 5)$ is the value of each feature.

4.2.2. CRF Real-Time Estimation. The characteristic mode and temporal evolution rule of the traffic flow state are investigated by using linear-chain CRF method [30], considering the supplementary features such as $E$ of the predicted road segment.

A linear-chain CRF is a discriminative probability undirected graphical learning model. The advantage of CRFs over hidden Markov models (HMM) is the relaxation of the independent assumptions between features. HMM is necessarily local in nature because they are constrained to binary transition and emission feature functions, which forces each state to depend only on the current label and each label to depend only on the previous label; however, CRF can use more global features. Additionally, CRF can obtain the global optimal value with the global normalization of all features. Building the special case of a linear-chain CRF,
features are restricted to depend on only the current and previous labels instead of arbitrary labels throughout.

As shown in Figure 4, an undirected graph $G$ consists of two kinds of nodes. The white nodes $Y = \{Y_1, Y_2, \ldots, Y_n\}$ represent traffic flow state variables at continuous time stamps, connected with supplementary features represented by blue nodes $X = \{X_1, X_2, \ldots, X_m\}$, $X_t = \{E, t\}$. Under observation sequence $x$, each random variable $Y_t$ obeys the Markov property; that is, $p(Y_t \mid X, Y_u, u \neq v) = p(Y_t \mid X, u \sim v)$ ($u$ and $v$ are adjacent nodes in $G$). Given observation sequence $x$ at time $t$, the probability of each possible state for $Y_t$ is defined as follows:

$$P(y \mid x, \lambda) = \frac{\exp \left( \sum_j \lambda_j f_j (y_{t-1}, y_t, x, t) + \sum_k \mu_k s_k (y_t, x, t) \right)}{\sum_t \exp \left( \sum_j \lambda_j f_j (y_{t-1}, y_t, x, t) + \sum_k \mu_k s_k (y_t, x, t) \right)},$$  

(12)

where $f_j (y_{t-1}, y_t, x, t)$ is the transfer feature function depending on the current time $t$ and previous time $t - 1$, $s_k (y_t, x, t)$ represents the state feature function depending on the time $t$, and $y_t$ denotes the traffic flow state at time $t$. When meeting the known deterministic feature conditions, the value of $f_j$ or $s_k$ is 1, otherwise 0. $\lambda_j$ and $\mu_k$ can be estimated from training data.

Assuming that there is a bunch of samples which are independent of each other in training dataset $Data = \{x^{(1)}, y^{(1)}\}$, maximum likelihood estimation learning $p(y \mid x, \lambda)$ is used to learn the parameters $\lambda$ by gradient descent algorithm [31]. Consider

$$L(\lambda) = \sum_x \frac{1}{y} \log \left[ \sum_y \exp \left( \sum_j \lambda_j f_j (y_{t-1}, y_t, x, t) \right) \right]$$

$$+ \sum_j \lambda_j f_j (y_{t-1}, y_t, x, t).$$  

(13)

By setting $s_k (y_t, x, t) = s_k (y_{t-1}, y_t, x, t)$, two feature functions are unified into $f_j (y_{t-1}, y_t, x, t)$, which includes an observation sequence $x$, the time $t$, the current state $y_t$, and the previous state $y_{t-1}$. Assuming that there are $K_1$ transfer feature functions and $K_2$ state feature functions under the given deterministic feature conditions, $K = K_1 + K_2$, $f_j (y_{t-1}, y_t, x, t)$ is represented as follows:

$$f_j (y_{t-1}, y_t, x, t)$$

$$= \begin{cases} t_j (y_{t-1}, y_t, x, t), & j = 1, 2, \ldots, K_1 \\ s_k (y_t, x, t), & j = K_1 + k; k = 1, 2, \ldots, K_2. \end{cases}$$

(14)

Assigning each feature function $f_j$ a weight $\lambda_j$ and given an observation sequence $x$, (12) can be transformed as follows:

$$P(y \mid x, \lambda) = \frac{\exp \left[ p(y \mid x, \lambda) \right]}{\sum_y \exp \left[ p(y \mid x, \lambda) \right]}$$

$$= \frac{\exp \left[ \sum_j \lambda_j f_j (y_{t-1}, y_t, x, t) \right]}{\sum_y \exp \left[ \sum_j \lambda_j f_j (y_{t-1}, y_t, x, t) \right]}.$$  

(15)

Through the above equation, the probabilities of five possible states for $Y_t$ can be informally denoted as a five-dimensional vector. Consider

$$P(t) = [p(y_t = Cg), p(y_t = Sl), p(y_t = Nm),$$

$$p(y_t = Mo), p(y_t = Un)].$$

(16)

4.2.3. Sequence Segments Matching. After applying CRF classifier to output a state vector of traffic flow at each time stamp, a state sequence segment can be obtained. Through finding similar state segments in history, we can obtain the state of traffic flow at the predicted time stamp according to the sequence matching with historical sequence segments. Since there may be no identical state sequence segments, measuring the similarity between sequence segments is necessary for sequence matching. Commonly used Euclidean distance is not suitable for the distance measurement of time-series sequences, which leads to the utilization of a nonlinear time alignment approach, namely, dynamic time warping (DTW).

In the process of state sequence segment matching, assuming that the current time is $t_{ct}$, a traffic state sequence segment obtained from CRF classifier previously is represented as $S = (s_{ct-n\Delta t}^{ct}, \ldots, s_{ct-\Delta t}^{ct}, s_{ct}^{ct})$ and a predicted traffic state after time $t_f$ can be denoted as $s_{ct+t_f}^{ct+t_f}$ DTW-based segment matching algorithm, shown in Algorithm 2, is employed to procure adequate matching segments through measuring DTW distances between the given state sequence segment and historical segments $S^* = (s_{ct-n\Delta t}^{ct}, \ldots, s_{ct-\Delta t}^{ct}, s_{ct}^{ct})$. In this algorithm, the predicted sequence segment is warped nonlinearly in time dimension to determine the similarity with historical segments. A constant threshold $\varepsilon$ is defined to filter adequate traffic flow state segments in history according to the DTW distances which are less than the chosen threshold; that is, $DTW^{ij}(t_{ct} \mid t_{el}) \leq \varepsilon$. Assuming there are he adequate historical segments chosen through DTW-based segment matching algorithm, so we can get the state
probability vectors of 5-dimensional at the predicted time $t_{ct} + t_l$, where $i_{th}$ probability vector is represented as

$$p^{(i)} (t = t_{ct} + t_l) = \begin{cases} p^{(i)} (s_i^* = Cg), & \text{if } s_i^* = Cg, \\ p^{(i)} (s_i^* = Sl), & \text{if } s_i^* = Sl, \\ p^{(i)} (s_i^* = Nm), & \text{if } s_i^* = Nm, \\ p^{(i)} (s_i^* = Mo), & \text{if } s_i^* = Mo, \\ p^{(i)} (s_i^* = Un), & \text{if } s_i^* = Un \end{cases} \quad (17)$$

In the above formula, $S^a(i)$ is the $i_{th}$ adequate historical sequence segment and $P^{(i)}(s_i^* = s)$ $(s \in \Omega)$ represents the probabilities of five possible states at the predicted time $t_{ct} + t_l$ according to the $i_{th}$ historical segment $S^a(i)$. Since $w^{(i)}$ is the weight of $i_{th}$ historical segment and can be calculated as

$$w^{(i)} = \frac{DTW^{(i)} [t_{ct}] [t_{ct}]}{\sum_{i=1}^{hs} DTW^{(i)} [t_{ct}] [t_{ct}]} \quad (18)$$

now, we can obtain the weighted vector at the predicted time $t (t = t_{ct} + t_l)$ as the final state result. Consider

$$P (t = t_{ct} + t_l) = \sum_{i=1}^{ha} w^{(i)} \cdot p^{(i)} (t = t_{ct} + t_l). \quad (19)$$

### 4.3. Dynamic Weighted Classifier Fusion

During the daytime, the taxi traffic operation characteristics, which provide ample data of arterial roads, can objectively reflect real road traffic conditions of the city from a certain extent. However, applying supplementary information especially road noise to predict the traffic condition may be affected by some interference factors such as man-made noise during the daytime. In the early morning hours, pure road noise is more likely to better reflect the road traffic conditions, which makes it possible to achieve higher prediction accuracy than what is achieved by insufficient taxi GPS traces. Obviously, two different approaches have various prediction accuracies during different time in a day.

Comprehensively considering temporal-spatial features and supplementary features, a dynamic weighted classifier fusion is utilized to fuse two approaches to obtain a better result. We give scores to the prediction effects of two approaches every hour in one day, represented as $score^{aHR, bD}_{CRF}$ and $score^{bD}_{DL}$, where $aHR$ represents the $a_{th}$ hour in one day and $bD$ denotes that $b_{th}$ day in history. Sampling num times per hour, we compare predicted state $s_i$ with actual state $s_i^*\atop^{\text{gt}}$ at each sampling in one hour. $s_i^\atop^{\text{gt}}$ represents the prediction result of $j_{th}$ sample in one hour that the timestamp $t$ belongs to, while $s_i^\atop^{\text{gt}}$ is the corresponding ground truth, which is accurately defined in evaluation part. If $s_i^\atop^{\text{gt}}$ is congruent with the ground truth, a positive value considering the probability of this state is assigned to the score; otherwise the score is assigned negative. Consider

$$score^{aHR, bD}_{CRF} = \left\{ \begin{array}{ll} P^{(bD)} (s_i^\atop^{\text{gt}} = s_i) \cdot P_i^{(bD)} (s_i), & \text{if } s_i^\atop^{\text{gt}} \text{ is } s_i^\atop^{\text{gt}} \\ -P^{(bD)} (s_i^\atop^{\text{gt}} = s_i) \cdot P_i^{(bD)} (s_i), & \text{if } s_i^\atop^{\text{gt}} \text{ is not } s_i^\atop^{\text{gt}} \end{array} \right. \quad (20)$$
where \( \text{score}_{aHR}^{j|aHR, bD} \) represents the score of the \( j \)-th sample in the hour \( aHR \) on day \( bD \). The sum of all scores in the hour \( aHR \) of day \( bD \) is calculated as follows:

\[
\text{score}_{aHR, bD}^{CRF} = \sum_{j=1}^{\text{num}} \text{score}_{aHR}^{j|aHR, bD}.
\]

Similarly, we can also get the score \( \text{score}_{aHR}^{j|aHR, bD} \) and score \( \text{score}_{DL}^{j|aHR, bD} \) as follows:

\[
\begin{align*}
\text{score}_{DL}^{j|aHR, bD} &= \begin{cases} 
P_{DL}(s^j_t = s^j) , & \text{if } s^j_t \text{ is } s^j_{gt} \\
-P_{DL}(s^j_t = s^j) , & \text{if } s^j_t \text{ is not } s^j_{gt} 
\end{cases} \\
\text{score}_{aHR, bD}^{DL} &= \sum_{j=1}^{\text{num}} \text{score}_{DL}^{j|aHR, bD} .
\end{align*}
\]

When the expectation of scores in a given hour \( aHR \) of all historical days is larger, the prediction will perform better. However, if the variance of scores in a given hour is larger, we will get worse prediction effect. Assuming that there are \( ev \) days in the training dataset, prediction effects \( \alpha_{aHR} \) and \( \beta_{aHR} \) in two respective approaches considering expectations and variances at the corresponding hour of \( ev \) days in the history can be defined as follows:

\[
\begin{align*}
\alpha_{aHR} &= \frac{\text{score}_{aHR}^{aHR, bD}}{1 + \sum_{bD=1}^{ev} \left( \text{score}_{aHR}^{aHR, bD} - \text{score}_{aHR}^{DL, bD} \right)^2} \\
\beta_{aHR} &= \frac{\text{score}_{DL, bD}^{aHR, bD}}{1 + \sum_{bD=1}^{ev} \left( \text{score}_{DL, bD}^{aHR, bD} - \text{score}_{DL, bD}^{DL, bD} \right)^2} .
\end{align*}
\]

where \( \text{score}_{aHR}^{DL, bD} \) and \( \text{score}_{DL}^{DL, bD} \) denote the average values of scores in hour \( aHR \) all \( ev \) days in two different approaches.

In prediction process, we apply CRF approach and deep learning approach to learn features sequentially, generating two classifications with two probability vectors \( (P_{CRF} \) and \( P_{DL} \) of 5-dimensional. The prediction result of O-Sense (donated as \( P_{OS} \)) can be calculated by fusing \( P_{CRF} \) and \( P_{DL} \) with respective weights at time \( t \) belonging to hour \( aHR \), which can be represented as follows:

\[
P_{OS} = \frac{\alpha_{aHR}^2 \times P_{CRF}}{\alpha_{aHR}^2 + \beta_{aHR}^2} \times P_{CRF} + \frac{\beta_{aHR}^2 \times P_{DL}}{\alpha_{aHR}^2 + \beta_{aHR}^2} \times P_{DL} ,
\]

\[
aHR = (1, 2, \ldots, 24) .
\]

5. Evaluations

5.1. Datasets. The datasets of Wuhan city are used for the evaluation of our traffic flow prediction and route computing. A representative region is selected to verify the validity of our mechanism. The following three available datasets are used.

1. Taxi Trajectories: the trajectory datasets are generated by 30,000 taxis over a period of three months from January to March in 2013. Total distance of the dataset is about 450 million kilometers, and whole number of GPS points reaches 890 million. The interval sampling each time is 4.5 minutes with the distance of 500 meters between two consecutive points.

2. Road Network: the road network of Wuhan is adopted to perform the experiments. In Figure 5, a snapshot of the Wuchang district road network in Wuhan is displayed in the rush hour (5 pm).

3. Environmental data: from a public website, we collect environmental information which contains road noise, temperature, wind speed, PM2.5, and rainfall.

4. Ground truth: the actual traffic flow which can be known from camera sensor on the road is used as the ground truth to measure the prediction accuracy.

5.2. Performance

5.2.1. Evaluation of Temporal-Spatial Deep Learning Approach. Several parameters need to be determined for utilizing a temporal-spatial deep learning approach. Since different network structures influence the forecast results obviously, the parameter effect needs to be tested. One of the most important problems in designing a neural network is to determine the size of the network. 128 nodes are chosen to be the number of input layer nodes. As the number of input layer node changes, different weighted mean accuracy (WMA) can be obtained. As shown in Figure 6, the best choice is 64 nodes based on our experiments. If there are more nodes, training the RBM model will increase both the redundancy and computational expense. However, fewer nodes cannot make full use of neural network to depict the characteristics of high-dimension, nonlinearity, and randomness in the transportation network.

As shown in Figure 7, road segment \( I_5 \) chosen as the forecasted road segment denotes the traffic flow from upstream link \( F \) to downstream road \( I \). As shown in Table 1, in order to predict the object traffic flow of \( I_5(t_{ct} + T_{predict}) \) at time \( t_{ct} + T_{predict} \), \( M (M = 32) \) can be calculated through number of input nodes in RBM most closely correlated traffic flow with different time indexes are selected from all the following available traffic flows: \( \{A_1(t_{ct}), A_2(t_{ct} - 1), \ldots, A_4(t_{ct} - T), A_5(t_{ct} - 1), \ldots, A_4(t_{ct} - T), \ldots, I_5(t_{ct} - 1), \ldots, I_5(t_{ct} - T) \} \), and

![Figure 5: The road network of Wuchang district in Wuhan.](image)}
Table 1: Most correlated traffic flows with road segment $I_3(t + T_{\text{predict}})$.

| Traffic flow | Correlation |
|--------------|-------------|
| $J_1(t)$     | 0.971911    |
| $F_3(t)$     | 0.967859    |
| $H_2(t)$     | 0.967743    |
| $I_3(t)$     | 0.967213    |
| $A_3(t-1)$   | 0.966754    |
| $A_3(t)$     | 0.966512    |
| $B_4(t)$     | 0.964985    |
| $A_3(t-2)$   | 0.966754    |
| $A_3(t)$     | 0.966512    |
| $B_4(t-1)$   | 0.964985    |
| $G_1(t-1)$   | 0.964985    |
| $A_4(t)$     | 0.962859    |
| $B_4(t)$     | 0.962859    |
| $G_1(t)$     | 0.962859    |
| $A_4(t-1)$   | 0.962859    |
| $B_4(t-1)$   | 0.962859    |
| $G_1(t)$     | 0.962859    |
| $A_4(t-2)$   | 0.962859    |
| $B_4(t-1)$   | 0.962859    |
| $G_1(t)$     | 0.962859    |
| $A_4(t-1)$   | 0.962859    |
| $B_4(t-1)$   | 0.962859    |

Table 2: Eigenvalue and contribution rate of the covariance matrix.

| Eigenvalue | Contribution rate |
|------------|-------------------|
| 2.2928     | 58.03%            |
| 0.8010     | 20.27%            |
| 0.6322     | 16.00%            |
| 0.1594     | 4.03%             |
| 0.0406     | 1.03%             |
| 0.0250     | 0.63%             |

so forth. In this paper, we choose 5 min as the time unit and $T = 20$.

In the PCA approach, we choose $\psi$ as 95 percent and 6 eigenvalues of the covariance matrix can be obtained as shown in Table 2.

After computing the cumulative contribution rate, we find that the cumulative contribution rate of the previous four principle components has achieved 98.33%.

Table 3: Principle component structure of traffic flow.

| Principle component | 1   | 2   | 3   | 4   | 5   | 6   |
|---------------------|-----|-----|-----|-----|-----|-----|
| $F_m$               | -0.0072 | -0.0116 | -0.0631 | -0.2166 | 0.0764 | -0.7433 |
| $c_m$               | 0.2076 | -0.7524 | -0.5373 | 0.2606 | 0.1545 | -0.0683 |
| $A_{mx}$            | 0.2358 | -0.4925 | 0.3296 | -0.5935 | -0.4530 | 0.1713 |
| $d_{mx}$            | 0.3975 | -0.1156 | 0.5612 | -0.0020 | 0.5919 | -0.2532 |
| $w_m$               | 0.5540 | 0.0862 | 0.1869 | 0.4959 | -0.1112 | 0.2889 |
| $l_m$               | 0.4671 | 0.2517 | -0.1530 | 0.1169 | -0.5413 | -0.4488 |

To express the predicted state and its probability more clearly, we define five numbers for each state; for example,
5.2. Evaluation of Temporally Related CRF Approach. Figure 9 demonstrates the prediction result of temporally related CRF approach with the prediction time $\varphi = 15$ min. According to the calculation, the prediction precision of CRF approach is nearly 0.75. Comparing the results of two approaches to predict separately in Figures 8 and 9, it can be inferred that the effect of deep learning approach is much better than CRF approach in most cases. However, temporally related CRF approach has better prediction effect on compensating for sparse data in deep learning approach. It takes full advantage of supplementary information especially road noise that provides effective information to improve the prediction accuracy when in the midnight.

5.2.3. Evaluation of Dynamic Weighted Classifier Fusion. In the following experiments, Figure 10 presents the expectations and the standard deviations of scores in the same hour varied in these two approaches. Two observations can be found in Figure 10(a). First, deep learning approach in the daytime has good performance using ample temporal-spatial traffic features. Second, the performance of CRF is higher when there is sparse data in the traffic network. As shown in Figure 10(b), CRF approach performs unstably in daytime on the account of the supplementary features such as road noise mainly caused by many active sources rather than just the vehicles in daytime.

The dynamic weights of two different approaches in the same hour of one day are showed in Figure 11, which reveals that we can make more use of the advantages of two classifiers through fusing two approaches with the variety of time. Figure 12 presents better results obtained by fusion in different time of one day. For example, between 1 to 5 am in the morning, due to the sparse data in the traffic network, deep learning approach may have a low degree of recognition. However, CRF can obtain higher prediction precision during the period of time by extracting temporally related supplementary features. Hence, more accurate results can be gained by O-Sense which combines two approaches.

Figure 13 shows the precision, the recall, and the F1-measure of three approaches. As is shown in Figures 13(a) and 13(b), under the same prediction time, O-Sense can obtain 8% higher precision and recall than existing deep learning approaches. In Figure 13(c), the F1-measure, which considers both the precision and the recall, also reveals that O-Sense can obtain high accuracy. O-Sense comprehensively uses temporal-spatial traffic flow information and temporally
related supplementary features in the traffic network. We can conclude that the effect of O-Sense is closer to the deep learning approach. Moreover, combining temporally related supplementary features used by CRF approach to overcome the problem of sparse data in deep learning, the prediction result of O-Sense can be further improved.

5.2.4. Evaluation of Online Route Computing. As is shown in Figure 14, two routes are selected to evaluate the performance of our mechanism. Route 1 starts from the Wuhan University to Hankou Railway Station and route 2 starts from Wuhan University to Guangda Masion. Route 1 has longer path length and costs much more time for travel than that of route 2, which can be seen in Figure 4.

Comparing Figures 14(a) and 14(b), there are several observations as follows: Firstly, the traffic conditions almost remain unchanged at about 6 am and 12 am on route 1 and route 2. O-Sense and STR can both relatively accurately estimate the travel cost of the routes at that time. Secondly, the prediction performance of O-Sense is much better than STR in the time period containing different traffic conditions on route 1 (about 8 am and 4:30 pm). It can be inferred that starting at 8 am in the morning, driving on route 1 needs more than one hour to complete the whole path. The traffic condition of route 1 varies from congestion to slow then to normal. Hence, the total travel cost of the whole route depends on the real-time traffic condition of road segments when the user actually drives on them. Nevertheless, STR approach considers that the traffic conditions of road segments are always the same as their initial traffic condition at the departure time. It leads to the estimation that the travel cost by STR approach is much lower than the real travel cost.
longer than the ground truth. The situation at 4:30 pm is similar to the situation at 8 am, which can be seen as the estimated travel cost by STR is much shorter than ground truth. It can be inferred that the traffic condition on route 1 becomes congested after driving for about 30 minutes. The actual travel cost of the remaining path is much longer than that computed by the initial condition of each road segment in STR approach. Thirdly, route 2 takes less time than route 1 and the traffic conditions in that time period do not change too much; hence the STR performs better effects on route 2 at about 8 am and 4:30 pm.

Figure 15 shows the variation of driving speed at different places when the participant drives along route 1. STR keeps the speed of each road segment the same as that of the starting time for its static estimation policy, while O-Sense can estimate the real-time speeds for all the places on the route based on the predicted arrival time. When the user drives from the downtown area at 8 am, the traffic condition is congested. After the user drives about 4/6 distance of the whole path, the traffic flow shows high variation over time and traffic condition on current road segment becomes unimpeded. The traffic condition can be estimated by both STR and O-Sense on account of the rush hour at the beginning of traveling. Nevertheless, with the driving distance increasing, the traffic condition of the driving area becomes unblocked. Owning to the online traffic condition prediction of current road segment, O-Sense can perceive this variation and estimate the travel cost more accurately.
6. Conclusions

This paper proposes an online-traffic-prediction based route finding mechanism, namely, O-Sense, for fastest route finding with large scale taxi GPS traces and environmental information. In O-Sense, a temporal-spatial deep learning approach with preprocessed traffic-related features can effectively extract nonlinear, random, and high-dimensional characteristics from the traffic flow changes. Utilizing environment features as supplementary, CRF classifier can further reflect the dynamic patterns in the transportation network. A dynamic weighted classifier fusion approach is used to obtain a better prediction result, which helps to find the fastest route based on online route computing. We adopt trajectory datasets generated by 30,000 taxis over a period of three months in Wuhan to evaluate our approach. The experiment results showed that O-Sense can improve the travel cost estimation accuracy effectively.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partially supported by National Key Basic Research Program of China “973 Project” (no. 2011CB707106), Development Program of China “863 Project” (no. 2013AA-122301), National Natural Science Foundation of China “NSFC” (nos. 41127901-06, 61303212, and 61373169), the Program for Changjiang Scholars and Innovative Research Team in University (no. IRT1278), and the Natural Science Foundation of Hubei Province of China (no. 2014CFB191).

References

[1] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, “T-finder: a recommender system for finding passengers and vacant taxis,” IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pp. 2390–2403, 2013.
[2] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, “Urban computing with taxicabs,” in Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp ’11), pp. 89–98, ACM, September 2011.
[3] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li, “Towards mobility-based clustering,” in Proceedings of the 16th ACM SIGKDD
[4] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xie, “Discovering spatio-temporal causal interactions in traffic data streams,” in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1010–1018, August 2011.

[5] F. Zhang, D. Wilkie, Y. Zheng, and X. Xie, “Sensing the pulse of urban refueling behavior,” in Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp ’13), pp. 13–22, September 2013.

[6] A. Thiagarajan, L. Ravindranath, K. LaCurts et al., “VTrack: accurate, energy-aware road traffic delay estimation using mobile phones,” in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys ’09), pp. 85–98, November 2009.

[7] X. Niu, Y. Zhu, and X. Zhang, “DeepSense: a novel learning mechanism for traffic prediction with taxi GPS traces,” in Proceedings of the IEEE Global Communications Conference (GLOBECOM ’14), pp. 2745–2750, Austin, Tex, USA, December 2014.

[8] J. Zheng and L. M. Ni, “Modeling heterogeneous routing decisions in trajectories for driving experience learning,” in Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp ’14), pp. 951–961, September 2014.

[9] I. Tsyuoshi and M. Sugiyama, “Trajectory regression on road networks,” in Proceedings of the 25th AAAI Conference on Artificial Intelligence, pp. 203–208, August 2011.

[10] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag, “Adaptive fastest path computation on a road network: a traffic mining approach,” in Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB ’07), pp. 794–805, September 2007.

[11] E. Kanoulas, Y. Du, T. Xia, and D. Zhang, “Finding fastest paths on a road network with speed patterns,” in Proceedings of the 22nd International Conference on Data Engineering (ICDE ’06), pp. 1–10, April 2006.

[12] J. Zheng and L. M. Ni, “Time-dependent trajectory regression on road networks via multi-task learning,” in Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI ’13), pp. 1048–1055, July 2013.

[13] M. van der Voort, M. Dougherty, and S. Watson, “Combining Kohonen maps with ARIMA time series models to forecast traffic flow,” Transportation Research Part C: Emerging Technologies, vol. 4, no. 5, pp. 307–318, 1996.

[14] J. Yuan, Y. Zheng, X. Xie, and G. Sun, “Driving with knowledge from the physical world,” in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 316–324, August 2011.

[15] J. Yuan, Y. Zheng, X. Xie, and G. Sun, “T-drive: enhancing driving directions with taxi drivers’ intelligence,” IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 1, pp. 220–232, 2013.

[16] P. S. Castro, D. Zhang, and S. Li, “Urban traffic modelling and prediction using large scale taxi GPS traces,” in Pervasive Computing, vol. 7319 of Lecture Notes in Computer Science, pp. 57–72, Springer, Berlin, Germany, 2012.

[17] M. Lippi, M. Bertini, and P. Frasconi, “Collective traffic forecasting,” in Machine Learning and Knowledge Discovery in Databases, vol. 6322, pp. 259–273, Springer, Berlin, Germany, 2010.

[18] S. Sun, C. Zhang, and G. Yu, “A Bayesian network approach to traffic flow forecasting,” IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 1, pp. 124–133, 2006.

[19] S. Sun and C. Zhang, “The selective random subspace predictor for traffic flow forecasting,” IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 2, pp. 367–373, 2007.

[20] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, “An interactive-voting based map matching algorithm,” in Proceedings of the 11th IEEE International Conference on Mobile Data Management (MDM ’10), pp. 43–52, May 2010.

[21] J. Yuan, Y. Zheng, C. Zhang et al., “T-drive: driving directions based on taxi trajectories,” in Proceedings of the 18th International Conference on Advances in Geographic Information Systems ACM SIGSPATIAL (GIS ’10), pp. 99–108, November 2010.

[22] B. C. Dean, Continuous-time dynamic shortest path algorithms [Master thesis], Massachusetts Institute of Technology, 1999.

[23] A. Jagota, “Approximating maximum clique with a Hopfield network,” IEEE Transactions on Neural Networks, vol. 6, no. 3, pp. 724–735, 1995.

[24] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Computation, vol. 14, no. 8, pp. 1771–1800, 2002.

[25] X. Glorot, A. Bordes, and Y. Bengio, “Domain adaptation for large-scale sentiment classification: a deep learning approach,” in Proceedings of the 28th International Conference on Machine Learning (ICML ’11), pp. 513–520, July 2011.

[26] D. R. Johnson and E. G. Saunders, “The evaluation of noise from freely flowing road traffic,” Journal of Sound and Vibration, vol. 7, no. 2, pp. 287–309, 1968.

[27] A. Calixto, F. B. Diniz, and P. H. T. Zannin, “The statistical modeling of road traffic noise in an urban setting,” Cities, vol. 20, no. 1, pp. 23–29, 2003.

[28] W. H. K. Lam, H. Shao, and A. Sumalee, “Modeling impacts of adverse weather conditions on a road network with uncertainties in demand and supply,” Transportation Research Part B: Methodological, vol. 42, no. 10, pp. 890–910, 2008.

[29] T. H. Maze, M. Agrawal, and G. Burchett, “Whether weather matters to traffic demand, traffic safety, and traffic operations and flow,” Transportation Research Record, no. 1948, pp. 170–176, 2006.

[30] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” in Proceedings of the 8th International Conference on Machine Learning (ICML ’01), pp. 282–289, June 2001.

[31] L.-K. Liu and E. Feig, “A block-based gradient descent search algorithm for block motion estimation in video coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 4, pp. 419–422, 1996.