Neural Computing and Applications (2020) 32:10435–10449
https://doi.org/10.1007/s00521-019-04579-x

ORIGINAL ARTICLE

Bus travel time prediction based on deep belief network with back-propagation

Chao Chen[1,2] · Hui Wang[1] · Fang Yuan[1] · Huizhong Jia[1] · Baozhen Yao[1]

Received: 30 November 2018 / Accepted: 19 October 2019 / Published online: 2 November 2019

Abstract
In an intelligent transportation system, accurate bus information is vital for passengers to schedule their departure time and make reasonable route choice. In this paper, an improved deep belief network (DBN) is proposed to predict the bus travel time. By using Gaussian–Bernoulli restricted Boltzmann machines to construct a DBN, we update the classical DBN to model continuous data. In addition, a back-propagation (BP) neural network is further applied to improve the performance. Based on the real traffic data collected in Shenyang, China, several experiments are conducted to validate the technique. Comparison with typical forecasting methods such as *k*-nearest neighbor algorithm (*k*-NN), artificial neural network (ANN), support vector machine (SVM) and random forests (RFs) shows that the proposed method is applicable to the prediction of bus travel time and works better than traditional methods.

Keywords Bus travel time prediction · Multi-factor influence · Deep belief network · Machine learning models

1 Introduction

With the development of technologies and information systems, the intelligent transportation systems (ITSs) and advanced traveler information systems (ATISs) have been widely deployed in city region. In the field of public transportation, the delivery of real-time traffic information is actually the most visible applications of ITS [1]. Traditionally, this type of information is delivered in the form of arrival and departure times on digital boards at bus stops. More recently, passengers can obtain it via smartphone apps and in-vehicle screens. Currently, public transport can always serve passengers in an easy and comfortable way. However, due to some unpredictable factors, sometimes there is an early arrival or delay at a particular stop. For the lack of operational reliability, many travelers incline to choose private cars rather than public transport. Transit agencies realize that providing an accurate bus arrival time is valuable to attract more passengers and improve management or service level. They can adjust their bus schedules by applying a higher or lower speed in advance and ultimately achieve the goal of reducing the waste of bus resources.

At the same time, the availability of accurate bus information can also help passengers to efficiently schedule their departure time, reduce their waiting time and make smart choices for their travel [2, 3]. Given the bus arrival time information, passenger can choose suitable travel mode for their journeys. Figure 1 shows an example for illustrating the effect on providing accurate bus information at bus stop. In the example, a passenger attempts to travel from Stop A to Stop B. He/she actually has three choices on this trip. If the passenger has enough time, he/she can choose any one of the two bus routes (i.e., route no. 1 and no. 2). However, if the time is short, the passenger can take a taxi to the destination. Furthermore, if the passenger knows the bus arrival times of the next buses of the two bus routes (e.g., 09:00 and 09:05, respectively), he/she will wait for the next bus of route no. 2 rather than that of

✉ Baozhen Yao
yaobaozhen@dlut.edu.cn

[1] State Key Laboratory of Structural Analysis for Industrial Equipment, School of Automotive Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China

[2] Urban Planning Group, Department of the Built Environment, Eindhoven University of Technology, PO Box 513, Vertigo 8.16, 5600MD Eindhoven, The Netherlands

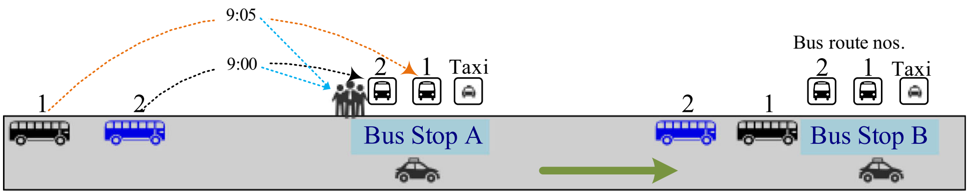


Fig. 1 An example for accurate bus information at bus stop

route no. 1. As a result, the waiting time of the passenger will be reduced.

As the objective of bus travel time prediction is to provide such information, this problem has been one of the hottest issues in ITSs. It should be noted that providing bus travel time prediction precisely in areas with litter external influence, such as rural areas, is easy in a way. However, the problem becomes much more complex in urban areas. The motivation of this paper is to improve the accuracy of prediction yielded by current bus travel time prediction models.

In the past decades, by using historical data or online data (obtained by the global positioning system), various forecasting models and techniques have been proposed to predict bus travel time. These techniques include historical average model [4–6], statistical model [7–11], nonparametric regression model [2, 12–14], machine learning model [2, 3, 15–24] and hybrid model [20, 25, 26].

Historical average model is used to predict the current and future bus travel time within a given period of time by averaging the historical travel times. Chung and Shalaby [6] presented a school bus arrival time prediction model, which combined historical moving average model with an operational strategy. Their results indicated the proposed model was powerful in real-life application. However, a noteworthy feature of this type of model is the requirement of stable traffic pattern. The reliability of the model would greatly decrease if the traffic pattern has large variations [27, 28]. It should be noted that in recent research, this type of model is presented only for comparison purposes.

Statistical model can be further divided into two main categories: time-series model and regression model. With respect to time-series model, such model assumes that the future state of a bus depends on the trend of past several states of the same bus. This model usually causes a short time lag between the prediction value and real data. The commonly used time-series models include autoregressive moving average (ARMA) [29], generalized autoregressive conditional heteroscedasticity model [7] and seasonal ARIMA model [8, 30]. Williams and Hoel [4] compared the performances of several methods for traffic flow rate prediction. These methods include seasonal ARIMA, random walk, historical average and deviation from historical average. Their results showed the seasonal ARIMA could provide the best forecasts for all performance statistics. It should be noted that the prediction accuracy also decreases significantly when the relationship between real-time and historical data is complicated [16]. The other kind of statistical model is the regression model. This kind of model is developed to verify the effect of different factors on bus travel time. In the model, factors such as bus dwell time, traffic conditions and travel distance are treated as independent variables, while bus travel time is the dependent variable. Observed that the travel times in current and future status have linear relationship, at the same time, the slope and intercept of this relationship change subject to the time of day, Rice and Van Zwet [9] proposed a linear regression with time-varying coefficients to predict travel time on freeways. However, the prediction accuracy of regression model heavily depends on the selection of independent variables [16].

Recently, due to the absence of estimating parameters, *k*-NN has been extensively employed in travel time prediction. It is actually a kind of nonparametric regression model. Based on *k*-NN, You and Kim [12] proposed a hybrid travel time forecasting model to predict link travel times in congested road networks. By using historical and real-time data, Chang et al. [14] developed a dynamic model to predict bus multi-interval path travel time. Results in their paper showed that *k*-NN method was effective in the aspects of both accuracy and computing time. Yu et al. [2] proposed four models, including SVM, ANN, *k*-NN and linear regression, to predict bus arrival time at the bus stop with multiple routes. Their results also indicated *k*-NN was an effective method for bus travel time prediction. However, nonparametric regression model becomes costly in execution time when the sample size is large [2, 14].

As a subfield of artificial intelligence, machine learning methods are widely used in transportation. Of various machine learning methods, ANN and SVM are the most widely used models in bus travel time prediction. The ability of ANN for solving complex nonlinear problems has been proved in many applications [31, 32]. Chien et al. [16] developed two ANN-based models to predict bus arrival time, namely link-based ANN and stop-based ANN. To further improve the prediction accuracy of both models, an adaptive algorithm was developed. Results indicated that the adaptive algorithm could actually improve the performances. Mazloumi et al. [17] proposed an integrated framework that consists of two ANNs to predict both the average and variance of travel times. The proposed method was validated by using the data collected in Melbourne, Australia. However, the structure of ANN (i.e., input variables, hidden-layer size, learning rate, etc.) depends on the experience of researchers [33–36].

Characterized by the capacity control of the decision function, the use of the kernel functions and the sparse solution, SVM has also been widely used in the field of prediction. By using SVM-based model and the data of preceding buses, Yu et al. [18] predicted the bus arrival time for a bus route. Their results showed SVM was a powerful method for bus arrival time prediction. To further improve the performance of the SVM, Yu et al. [19] firstly applied the Grubbs' test method to remove outliers from the input data. Then, an SVM-based model with forgetting factors was introduced. Results showed that the improved SVM model outperformed the standard SVM.

However, ANN and SVM are complicated because of the large number of parameters needed to be adjusted [3]. In considering coping with the bad effect of parameters selection, attention has been focused on an emerging type of machine learning method, RFs. As an ensemble learning algorithm, RFs model can easily explain the importance of thousands of variables. Through the procedure of random selection of features and training samples, it has been proved to be efficient in avoiding overfitting. Some applications of RFs can be found in bus travel time prediction. Gal et al. [20] proposed a combination method of queuing theory and RFs to predict bus travel time. In their paper, RFs were used to detect the outliers in historical data. Yu et al. [3] developed a RFs model to predict bus travel time. To avoid the influence of massive data, a preselection method for the training set, near neighborhoods, was proposed in their paper. Finally, their model was calibrated and validated by using the data of two bus routes in China.

To avoid the drawback of single prediction model, several types of hybrid models have also been studied. These methods include ANN model with Kalman filter-based algorithm [25], SVM model with Kalman filter-based algorithm [26], queuing theory with RFs algorithm [20] and so on. Results showed that hybrid models could certainly obtain better performances than those single prediction models.

Among all varieties of prediction models, the accuracy of the bus travel time prediction heavily depends on historical and real-time traffic data. Nowadays, these traffic data are now undergoing rapid growth with the widespread of traffic sensor technologies. In addition, factors, which have influences on bus travel time, commonly have complicated relationships. Finding a reliable method that can identify the relationship among these influencing factors is a challenging problem. However, the above-mentioned bus travel time prediction methods are mainly based on shallow learning architecture. It is difficult for them to easily and precisely identify the relationship among these influencing factors with the boom of information.

In recent years, deep learning models have drawn a lot of attentions. By adopting multilayer or deep architectures, deep learning models can extract inherent features of data from the lowest level to the highest level even if the data set is large. Currently, there are some applications of deep learning models in traffic flows or travel time prediction [37–46]. The first traffic flow prediction model based on deep architecture was proposed by Lv et al. [37]. In their paper, a stacked autoencoder model was developed to learn traffic features from the data collected in California freeway systems. Experimental results showed the prediction method based on deep architecture was superior to BPNN, SVM and RBFNN models [37]. Siripanpornchana et al. [38] developed an urban freeway travel time prediction model based on deep learning architectures. Different from Lv et al. [37] that used autoencoder model, the concept of DBN, which comprises a stack of restricted Boltzmann machines (RBM), was used in their paper. However, to our knowledge, literature about bus travel time prediction using deep learning models is relatively scarce. In addition, the relationship among factors affecting the bus travel time is usually more complex than that affecting the traffic flows. Concerning the good features learning ability of deep learning models, this paper attempts to propose a new bus travel time prediction model based on the concept of DBN. In the aspect of model inputs, previous studies indicate that the traffic condition is regarded as the main factor influencing the accuracy of bus travel time prediction. Data that can reflect the traffic condition can be summarized as: (1) travel speed; (2) travel time; (3) distance; (4) emergency situation; and (5) dwell time. In this paper, due to data availability, both bus running time and bus dwell time (the number of passengers waiting to board) of the preceding vehicles (temporal and spatial data) are used to reflect the traffic condition and to predict the travel time of next bus.

DBN is one of the widely used deep learning-based methods. It has been proved for its capacity to extract nonlinear characteristics [42, 47]. However, the classical DBN is developed to learn features from binary data rather than continuous data. Considering the data used in this paper are continuous, a variation of RBM in DBN, called Gaussian–Bernoulli RBM (GBRBM), is employed. The basic units of DBN are improved to make the model have the ability of extracting and learning continuous data features. To further improve the performance of DBN, a shallow learning architecture, named BP neural network model, is also adopted to predict bus travel time in a supervised fashion.

This paper seeks to make three contributions to previous literature. Firstly, this paper attempts to develop a bus travel time prediction model based on the concept of DBN.

It is expected that an accurate prediction of bus travel time can be obtained. By using the prediction results, the anxiety and waiting time of passengers can be effectively reduced. Secondly, the basic units of DBN are improved to make the model have the ability of extracting and learning continuous data (traffic) features. Thirdly, to improve the performance of DBN, a shallow learning architecture, named BP neural network model, is also adopted to predict bus travel time in a supervised fashion. Experimental results show that the proposed method has superior performance in prediction accuracy.

The remainder of this paper is organized in the following way. Section 2 presents the bus travel time prediction model and the deep learning method based on the concept of DBN. Section 3 describes the details of our experiments and results. Finally, conclusions and future works are given in Sect. 4.

2 Methodologies

2.1 Bus travel time prediction model

Bus travel time prediction model is to predict the bus travel time between two consecutive bus stops. The framework of the prediction is illustrated in Fig. 2. The bus travel time on the target segment i (distance between bus stop B and C) need to be predicted when the target bus k arrives at the bus stop B. To obtain accurate prediction results, it is essential to find the appropriate factors that have influence on the travel time. As described in the literature, bus travel time between bus stops has relations with the bus dwell time and traffic conditions [3]. Yu et al. [3, 18] pointed out the traffic condition can be reflected by the running time of preceding buses and the most recent data provides the most reliable information. However, the number of passengers waiting to board (bus dwell time) also has effect on the bus running time since buses need more time to speed up or slow down. There are two general approaches to forecast bus travel time in previous studies. The first one is only to use the bus running time of preceding buses (dwell time is not explicitly expressed) [3, 18, 26], and the other approach is to use both bus running time and bus dwell time [27]. In addition, in an urban road network, road links do not exist in isolation. Traffic speed on both upstream and downstream links can affect the traffic speed of the current road. Therefore, in this paper, the average bus travel times of preceding buses on the current segment, upstream segment and downstream segment are all considered. Similarly, to find the appropriate factors to estimate bus dwell time and avoid the errors brought by unpredictable factors, the dwell time, the average dwell time and the variance of dwell time of targeted bus at the starting stop are also considered. It should be noted that the preceding buses in this paper only refer to the buses that just finished the travel between two consecutive stops and leave for the next stop.

Assuming that k represents the target bus for predicting, i represents the segment of target bus route. The input variables considered in this paper are illustrated as follows:

$$\hat{t}_{i,k} = f\left(t_{i-1,k}, t_{i,k-1}, t_{i+1,k-1}, dt_{i,k}, \overline{dt}_i, \sigma(dt)_i^2\right) \quad (1)$$

where $\hat{t}_{i,k}$ is the predicted travel time of bus k on segment i and $t_{i-1,k}$ represents the running time of bus k on segment $i-1$. It shows the traffic conditions on the upstream of segment i. $t_{i,k-1}$ means the running time of bus $k-1$ (i.e., the preceding bus) on segment i, expressing the traffic conditions on the current segment i, $t_{i+1,k-1}$ is the running time of bus $k-1$ (i.e., the preceding bus) on segment $i+1$, which represents the traffic conditions on the downstream of segment i, $dt_{i,k}$ shows the dwell time of bus k at the starting stop of segment i and $\overline{dt}_i$ and $\sigma(dt)_i^2$ represent the average and variance of dwell time at the starting stop of segment i, respectively. The prediction model proposed in this paper is to find the relationship among these influencing factors.

2.2 Deep belief network

DBN is a kind of artificial neural network inspired by imitating the process of cognition and inference of human brain [41]. It has become one of the most universal deep learning models and has been widely used in many fields. As an arbitrary undirected graphical model, DBN can be described as a generative energy-based model that has random multilayer units connecting the input layer and the output layer. It is constituted by a stack of RBMs, which provide DBN with the capacity of feature learning and

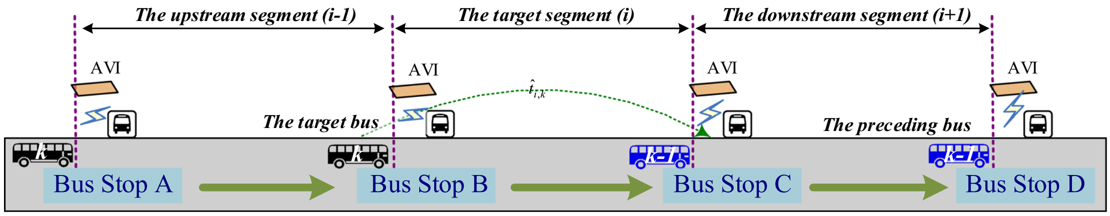


Fig. 2 The framework of bus travel time prediction

feature extraction. There are two layers existing in the RBM. One layer is constituted by visible units, while the other is composed of hidden units [42]. Visible units are used to represent observable data, while hidden units are used to capture dependencies among observed variables. An illustration of an RBM is shown in Fig. 3.

In this figure, $v_i(i = 1, \ldots, 3)$ is the unit in the visible layer, while $h_i(i = 1, \ldots, 4)$ is the unit in the hidden layer, and W represents the weight matrix.

However, the classical RBM was developed to solve problems only with binary data, limiting the applications for problem solving. Considering the traffic data are continuous and several variations of RBM have been presented in previous literature, in this paper, a variation of RBM that can deal with continuous data, called GBRBM, is introduced. In this model, the binary visible units are replaced by linear units with Gaussian noise [40]. Note that GBRBM is an energy-based model, in which hidden-layer variables are used to describe the probabilistic distribution of visible-layer variables. Thus, given m visible units and k hidden units, the probabilistic distribution over variables can be defined as follows [47]. It is actually a derivation of the energy function of a standard RBM.

$$E(v, h|\theta) = -\sum_{i=1}^{m}\frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{k} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{k}\frac{v_i}{\sigma_i}W_{ij}h_j \tag{2}$$

where $\theta = \{W, a, b\}$ is the structure parameter of GBRBM, a_i and b_j are the bias vectors for visible and hidden units, respectively, w_{ij} is the weight related to the connection between visible unit v_i and hidden unit h_j and σ_i is the standard deviation of Gaussian noise, which corresponds to the visible unit v_i.

Based on Eq. (2), the probabilistic distribution for each pair of visible unit and hidden unit (v, h) can be described as:

$$P(v, h|\theta) = \frac{e^{-E(v,h|\theta)}}{\sum_{v,h} e^{-E(v,h|\theta)}} \tag{3}$$

Thus, the probability of the vector in visible layer can be calculated by summing all probabilities of the vector in hidden layer, and the process can be described by:

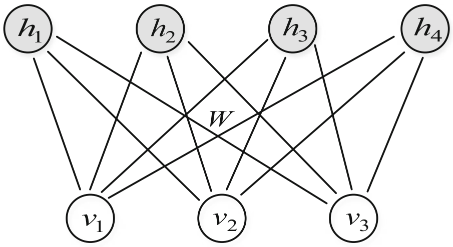


Fig. 3 The structure of RBM

$$p(v) = \frac{\sum_h e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}} \tag{4}$$

Since units in a layer only have connections to the units in another layer, the events are independent of each other. The conditional probabilities can be calculated by:

$$P(v_i|h, \theta) = N\left(a_i + \sigma_i \sum_{j=1}^{k} W_{ij}h_j, \sigma_i^2\right) \tag{5}$$

$$P(h_j = 1|v, \theta) = \sigma\left(b_j + \sum_{i=1}^{m}\frac{v_i}{\sigma_i}W_{ij}\right) \tag{6}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the most used activation function named sigmoid function. $N(\mu, \sigma^2)$ stands for the Gaussian distribution with mean μ and variance σ^2. In GBRBM, the deviation is also known as noise level, and in practice, it is usually set to 1.

For a given training data $X = \{x^{(t)}\},\ t = 1, 2\ldots, C$ (C is the number of sample size), data should be firstly standardized due to the disunity of weights and measures. After that, the structure parameters of GBRBM can be estimated by minimizing the negative log-likelihood. In this paper, a stochastic approximation method, namely contrastive divergence (CD) algorithm, is introduced to estimate the expected values. The structure parameters of GBRBM can be updated by the following equations:

$$\Gamma = \log \prod\nolimits_m p(v_m) \tag{7}$$

$$\Delta w_{ij} = \varepsilon\frac{\partial\Gamma}{\partial W_{ij}} = \varepsilon\left\langle\frac{v_i h_j}{\sigma_i^2}\right\rangle_d - \left\langle\frac{v_i h_j}{\sigma_i^2}\right\rangle_m \tag{8}$$

$$\Delta a_i = \varepsilon\frac{\partial\Gamma}{\partial a_i} = \varepsilon\left\langle\frac{v_i}{\sigma_i^2}\right\rangle_d - \left\langle\frac{v_i}{\sigma_i^2}\right\rangle_m \tag{9}$$

$$\Delta b_j = \varepsilon\frac{\partial\Gamma}{\partial b_j} = \varepsilon\left\langle\frac{h_j}{\sigma_i^2}\right\rangle_d - \left\langle\frac{h_j}{\sigma_i^2}\right\rangle_m \tag{10}$$

where ε is the learning rate and $\langle.\rangle_d$ and $\langle.\rangle_m$ represent the expected values of the training data and model, respectively. Details of how GBRBM is derived from the classical RBM are given in "Appendix."

In the DBN model, features of the observation data are extracted from the hidden units, and the obtained features then served as the input of another GBRBM. After stacking GBRBMs in such way, a high-dimensional feature vector can be obtained in the final GBRBM. However, GBRBM is originally used for unsupervised learning. One method for considering the supervised learning is to combine an additional supervised learning algorithm. In this paper, the output of the final GBRBM (i.e., high-dimensional feature vector) is used as the input of a BP neutral network, and the output of the BP neutral network is used as the final result (i.e., predicted bus travel time). The multilayer structure of the proposed model makes it can process highly structured data and learn features automatically. We report hereafter the pseudo-code of the procedure of training the DBN model. And we use the following symbols: T is the number of layers; m and k are, respectively, the number of visual units and hidden units; N is the number of iterations; ε is learning rate; W is the set of weights of GBRBMs to be estimated; $\theta = \{W, a, b\}$ is the structure parameter of GBRBM; Ψ represents the inherent features of data; ETF(data) extracts and transforms the features across the hidden layers; UWE(Ψ) is the procedure which updates the weights of GBRBM; ψ is the prediction results of DBN model; ψ^* is the best results of DBN model; BP(ψ^*) is the procedure of BP neural network; and Δ denotes the final solutions. In addition, the structure of DBN model for bus travel time prediction is shown in Fig. 4.

Algorithm 1: Deep Belief Network with Back Propagation

```
//Initialization//
Set $T \in [1,4]$, $m \in [150,200]$, $k \in [150,200]$, $N \in [100,500]$, $\varepsilon \in [0.0005,0.005]$.
Initialize the weights of GBRBMs $W$ .
//Model Training//
Data normalization.
For each combination of $(T, m, k, N, \varepsilon)$ do
    $\Psi \leftarrow ETF(data)$          //based on Eq. (2)-(6)
    While $N \neq \varnothing$ do
        $\theta \leftarrow UWE(\Psi)$.   //based on Eq. (7)-(10)
        $N = N - 1$
    End do
    Save the prediction results as $\psi$
    If $\psi < \psi^*$
        $\psi^* \leftarrow \psi$
    End if
End for
$\Delta = BP(\psi^*)$
$\Delta$ provides the final prediction results
```

3 Case study

3.1 Data descriptions

In this paper, the proposed model is tested by real-world data collected in Shenyang, the capital city of Liaoning Province in China. Shenyang has a highly developed urban transit network that comprises more than 222 bus routes. Providing accurate bus information is vital for transit agencies to attract more passengers. In Shenyang, each bus stop is equipped with a video detector. It is originally used to identify abnormal conditions. However, by using video recognition technology, the data of bus travel time and dwell time can also be obtained.

The selected bus route for testing is route No. 232, which spans 10.7 km. The operation time of route No. 232 is from 4:50 am to 10:00 pm, and the departure interval is approximately 2.5 min. The reason for selecting this route is that there is a large passenger demand on the route every day. It connects Santaizi bus stop, located in the suburb of Shenyang, to Wanda plaza, the city center. It has 19 bus stops and the location of the selected route is illustrated in Fig. 5. The detailed information is also given in Table 1. Note that the two directions of the route are considered separately in this paper.

To obtain the data of bus running time and dwell time, vehicle recognition has been carried out on the video data during every whole day of July 17–21, 2017 (Monday to Friday). By using vehicle recognition technology, the bus no., bus dwell time and bus running time can be acquired. However, sometimes, several buses reach the bus stop at the same time, making vehicle recognition difficult to implement. To reduce the influence of recognition errors, automatic vehicle location (AVL) data obtained by the GPS unit of buses are also used. By mapping the bus location [48] with the vehicle video records and filtering out the outliers, 50,876 valid observations of bus route, No. 232, are finally obtained. Then, based on the running directions, these valid observations are divided into two groups. And for each group, data are also divided into two subsets: 80 percent of the data are used as the training set, while remaining 20 percent are used as testing data. The descriptive statistics for each route segment are given in Table 2.

Note that the north direction is the direction in which buses move from the city center to suburb, while the south direction is from suburb to city center. From Table 2, it can be observed that the bus travel time in the south direction varies from 30 to 392 s and the travel time in the north direction is from 58 to 343 s. The average bus travel time for both directions is almost the same, while the standard deviation (SD) of the north direction is less than that of the south direction. The comparisons arise because passenger demand in the south direction is obviously more volatile than that in the north direction.

3.2 Performance indexes

The performance of the prediction model is evaluated with three metrics: mean absolute error (MAE), mean absolute percent error (MAPE) and root mean squared error (RMSE). The three metrics could measure the differences between the predicted and actual observed value in different aspects. They are defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|t_{i,k} - \hat{t}_{i,k}\right| \tag{11}$$

$$\mathrm{MAPE}\ (\%) = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|t_{i,k} - \hat{t}_{i,k}\right|}{t_{i,k}} \times 100\% \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(t_{i,k} - \hat{t}_{i,k}\right)^2} \tag{13}$$

where $t_{i,k}$ is the observed travel time and n is the number of observations.

3.3 Model identification

In model identification, observations of the first four days, i.e., July 17–20, 2017, are set as the training set and the observations on July 21, 2017, are selected as the testing set. Firstly, sensitivity tests are conducted to select rational input variables. To reduce the effect of different units of

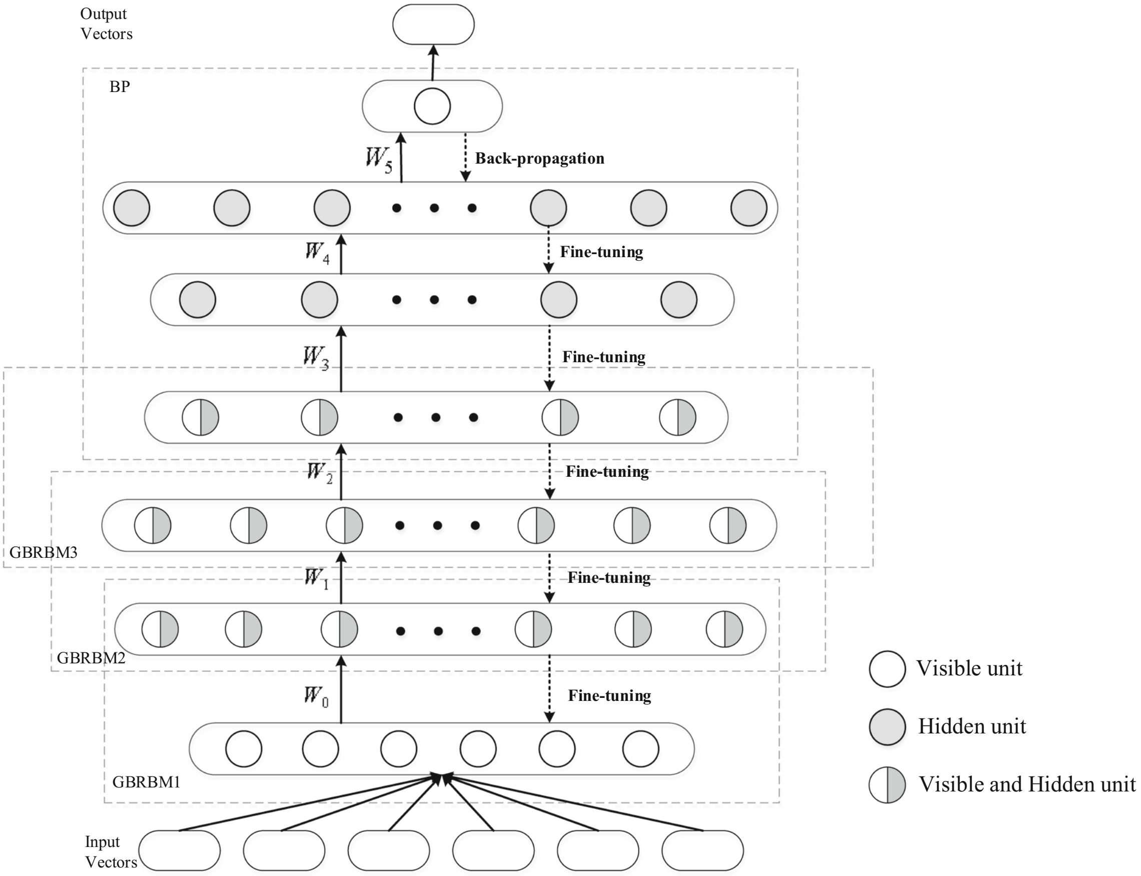


Fig. 4 The structure of DBM for bus travel time prediction

measurement, data normalization is conducted and the normalization method is shown in Eq. (14).

$$x_i^{\text{normal}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{14}$$

where x_i^{normal} is the normalized travel time, x_i is the raw sample data and $x_{\min}$ and $x_{\max}$ are the minimum and maximum value of sample data, respectively.

In this paper, six models with different input variables are calibrated and listed in Table 3. Table 3 also provides the average MAEs of the prediction results for both directions. It can be seen that the performance of the sixth model is the best. This indicates that taking different aspects of traffic conditions into consideration can truly improve the accuracy of bus running time prediction. Thus, the variables, $t_{i-1,k}, t_{i,k-1}, t_{i+1,k-1}, dt_{i,k}, \overline{dt}_i, \sigma(dt)$, are all selected as the input variables of the DBN model in this paper.

Finding a DBN with well features learning capacity is difficult since several numerical parameters, such as the number of layers (T), the number of units (m for visual units, k for hidden units), the number of iterations (N) and the learning rate (ε), need to be predetermined. In this paper, the range of these parameters is set as $T \in [1, 4]$, $m \in [150, 200]$, $k \in [150, 200]$, $N \in [100, 500]$ and $\varepsilon \in [0.0005, 0.005]$, respectively. Then, a tenfold cross-validation [3] and a grid search are used to identify the optimal parameter values. The results of $(T, m, k, N, \varepsilon)$ are $(3, 160, 180, 340, 0.001)$. Finally, a three-layer BP neural network with six hidden units is used to predict bus travel time. Since the data have been processed through standardization, the final predicted bus travel time should be obtained according to Eq. (15).

$$x_i^{\text{inv}} = x_i^{\text{normal}} \times (x_{\max} - x_{\min}) + x_{\min} \tag{15}$$

3.4 Numerical results

In this section, to validate the proposed DBN model, four other prediction models, including k-NN, ANN, SVM and RFs, are also employed. To have an equitable comparison, the same training and testing sets are used for all prediction models, and the input variables of all four models are the same as the ones of the DBN model. By sensitivity tests, the value of the parameter k is set as 3 for the k-NN model. In addition, in the ANN model, a standard three-layer ANN is used and the number of hidden units is set as 5. There are two hyper parameters, C and ε, existing in the SVM model. By using a tenfold cross-validation [3] and a grid search method, the value of the two parameters is set as 2^5 and 0.2, respectively. Beyond that, the radial basis function (RBF) kernel function is also used in the SVM model.

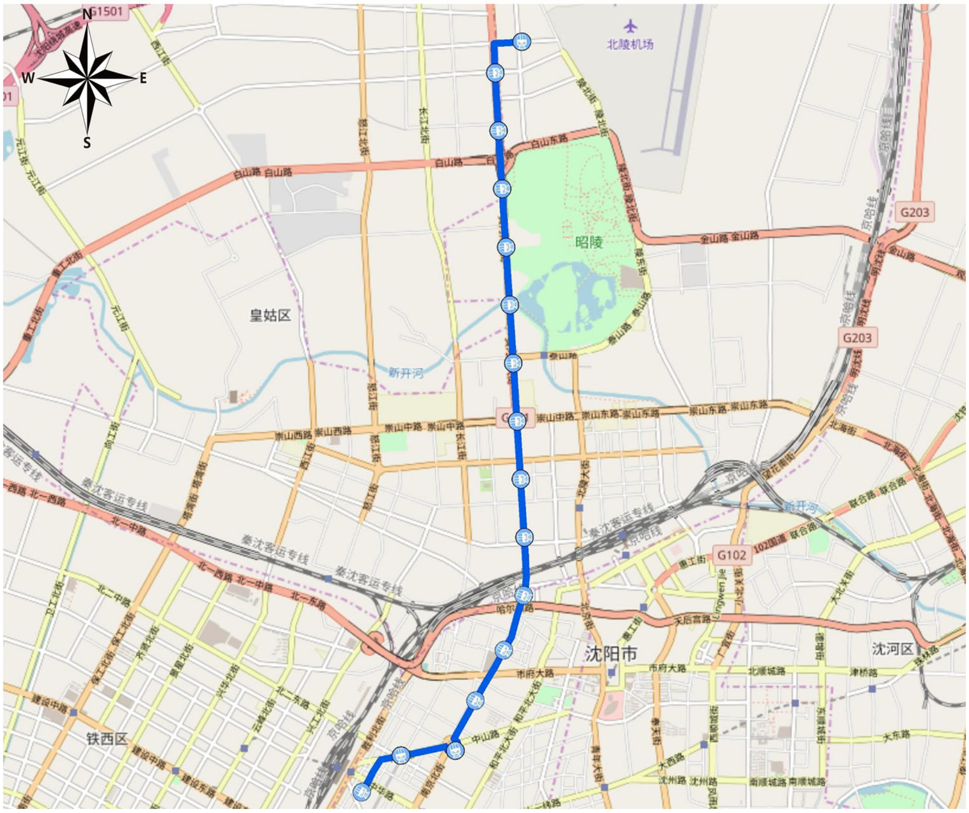


Fig. 5 The location of bus route No. 232

Table 1 Detailed information of the selected bus route

Stop no.	Stop name	Distance between bus stops (m)	Cumulative distance (m)
1	Santaizi	–	–
2	Songling Cultural Palace	861	861
3	Songshan Road	769	1630
4	Armed Police Corps	858	2488
5	Lingxi Road	851	3339
6	Xinle Dorm	874	4213
7	Xinle Site	787	5000
8	Liaoning Building	894	5894
9	Experimental Middle School	797	6691
10	Ningshan Road	794	7485
11	Qishan road	866	8351
12	Fourth Hospital	816	9167
13	Huangshi Square	858	10,025
14	North Market	862	10,887
15	Shenyi Second Hospital	824	11,711
16	Beiliuma Road	870	12,581
17	Taiyuan Road	794	13,375
18	Shenyang Railway Station	807	14,182
19	Wanda Plaza	820	15,002

There are also two hyperparameters, while using RFs: m_{try} and n_{tree}. By using the same parameter selection method as SVM model, the two parameters in RFs are set as 4 and 1000, respectively. Finally, the results are summarized in Fig. 6 and the details are given in Table 4.

Figure 6 shows the performance of all five prediction models for both directions of the selected bus route in terms of MAE, RMSE and MAPE. With respect to MAE, the DBN model outperforms other four prediction models by 46.01%, 67.47%, 24.17% and 17.91% in the south direction and 49.64%, 50.79%, 0% and 4.12% in the north direction, respectively. In terms of MAPE, the DBN model outperforms other four prediction models by 43.01%, 71.15%, 30.67% and 23.25% in the south direction and 50.77%, 43.73%, 8.53% and 10.65% in the north direction, respectively. Moreover, regarding the RMSE, the DBN model outperforms other four prediction models by 41.72%, 56.03%, 14.73% and 11.39% in the south direction. However, in the north direction, the performance of SVM model is the best. In summary, the DBN model has the best performance except the north direction in RMSE. It demonstrates that the DBN model is more effective than shallow learning architectures in extracting inherent features from massive data. In terms of MAE and MAPE, though the performances of SVM and RFs models are worse than that of the DBN model, the results of the two models are still better than *k*-NN and BP network model. It is mainly because the structure risk minimization principle of SVM and feature reduction principle of RFs can effectively avoid overfitting problem. It also illustrates why the two types of prediction model are widely used in the forecast field. Moreover, though the performance of *k*-NN model is worse than that of SVM, RFs and DBN models, it performs better than BP network model and the performance of BP network model is the worst. Thus, considering the simple structure and easy implement, *k*-NN is still a satisfying model for bus travel time prediction.

As to computation time, all models were coded in MATLAB of version R2012a and executed on a PC equipped with 4 GB of RAM and a dual-core 3.2 GHz processor. The results are listed in Table 5. In Table 5, it is obvious that the computation time of RFs is the least among all bus travel time prediction models while the *k*-NN model has the maximum running time. Specifically, the computation time of *k*-NN is almost 20 times that of RFs. It could be contributed to the process of similarity measurement in the *k*-NN model. Compared with the *k*-NN model, the shallow learning methods, BP network and SVM models, have relatively shorter running time. Moreover, due to the property of multilayer, it takes more computation time to establish a trained DBN model. However, with the help of cloud and parallel computing, the DBN model can be easily applied to the real-time bus travel time prediction. Considering the trade-off of accuracy and computation time, DBN model is the best approach among the five bus travel time prediction models and RFs is also a competitive method.

Since bus travel time is affected by time and space variables, there are significant differences in the bus travel time between peak hours and off-peak hours and between different

Table 2 Descriptive statistic for the valid observation

Segment no.	Route direction	Sample size (vehicle)	Descriptive statistics			
			Min (s)	Max (s)	Ave (s)	SD
1	South	1561	65	87	80	5.56
	North	1470	101	139	126	11.32
2	South	1481	120	148	136	8.99
	North	1190	121	154	140	11.04
3	South	1561	30	73	62	8.86
	North	1381	183	218	205	10.93
4	South	1545	135	191	167	20.67
	North	1323	117	151	136	11.97
5	South	1497	120	157	144	9.56
	North	1176	116	155	138	13.65
6	South	1481	30	75	57	11.11
	North	1367	147	186	171	12.86
7	South	1384	105	147	130	12.51
	North	1396	83	121	104	12.95
8	South	1497	74	115	101	10.7
	North	1381	305	343	329	11.93
9	South	1432	215	313	274	32.74
	North	1249	71	123	103	18.57
10	South	1400	210	340	297	44.51
	North	1572	58	97	80	13.46
11	South	1484	160	250	216	31.81
	North	1293	98	164	142	18.79
12	South	1416	190	248	226	19.26
	North	1381	217	259	242	13.98
13	South	1545	90	130	117	10.01
	North	1264	61	97	80	13.29
14	South	1560	165	197	181	11.24
	North	1367	171	207	191	12.98
15	South	1465	85	115	104	8.63
	North	1220	68	110	93	13.87
16	South	1610	80	127	116	10.91
	North	1278	188	232	213	14.55
17	South	1352	90	128	115	9.5
	North	1352	195	239	222	14.18
18	South	1593	220	392	316	55.41
	North	1352	120	162	143	14.79

driving directions. The reliability of predicted bus travel time would decrease in peak hours because at this time, traffic conditions are more complex and the number of passengers waiting to board is larger. Thus, in this paper, to further validate the proposed DBN model, the bus travel time predictions for both south and north directions in peak hours (i.e., 7 am–9 am and 5 pm–7 pm) on July 21, 2017, are also conducted. The prediction results of the five models in peak hours in terms of MAPE are shown in Fig. 7.

For all prediction models, it can be seen from Fig. 7 that the results of the north direction are better than those of the south direction. Traffic conditions and the number of passengers waiting to board in the south direction are always more complex than those in the north direction since the south direction is from the suburb to city center, while the north direction is from the city center to suburb. This indicates that prediction accuracy decreases as the traffic conditions become more complex. Nonetheless, compared with other four prediction models, the performance of DBN model in peak hours remains to be the best. In summary, the prediction results obtained by DBN model are shown to be more reliable than those obtained by other prediction models.

Table 3 Prediction models with different input variables

Model	Input variables						Average MAE(s)	
	Upstream traffic conditions	Current traffic conditions	Downstream traffic conditions	Bus dwell time at bus stop				
	$t_{i-1,k}$	$t_{i,k-1}$	$t_{i+1,k-1}$	$dt_{i,k}$	$\overline{dt_i}$	$\sigma(dt)_i^2$	South	North
1		√	√	√	√		3.49	3.82
2		√	√	√		√	3.07	2.98
3	√	√		√	√		3.34	3.39
4	√	√		√		√	3.15	2.95
5	√	√	√		√	√	3.54	2.91
6	√	√	√	√	√	√	**2.98**	**2.79**

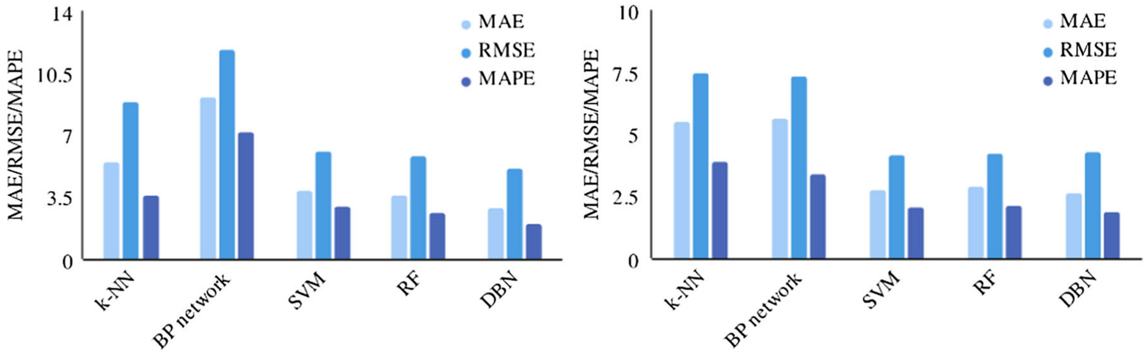
Finally, the detailed prediction results for both south and north directions in peak hours on July 21, 2017, are illustrated in Fig. 8. It can be seen that, in both directions, the maximum error of all segments is less than 12.19. Figure 8 also shows that the prediction errors of segments 9, 10 and 11 in the south direction are the largest, while in the north direction, the segments are 5 and 7. In Table 2, the standard deviations of the segments 9, 10 and 11 in the south direction and segments 5 and 7 in the north direction are 32.74, 44.51, 31.81, 13.65 and 12.95, respectively, which are obviously larger than those of other segments in each direction. The results give a further verification of that the prediction accuracy decreases as the increase in the uncertainty of traffic conditions.

4 Conclusion

This paper proposed a bus travel time prediction model based on the concept of deep belief network. Since the input variables of the proposed model are continuous data, the basic units of DBN were improved by introducing

Table 4 The average results of the five different prediction models

Direction		South	North
MAE	*k*-NN	5.52	5.54
	BP network	9.16	5.67
	SVM	3.93	2.79
	RFs	3.63	2.91
	DBN	2.98	2.79
MAPE	*k*-NN	3.65	3.92
	BP network	7.21	3.43
	SVM	3.00	2.11
	RFs	2.71	2.16
	DBN	2.08	1.93
RMSE	*k*-NN	8.94	7.49
	BP network	11.85	7.36
	SVM	6.11	4.17
	RFs	5.88	4.26
	DBN	5.21	4.33



(a) The performance for south direction

(b) The performance for north direction

Fig. 6 The performance of the five models

Table 5 Computation time of all prediction models (unit: s)

Direction	*k*-NN	BP network	SVM	RFs	DBN
South direction	41,778	4215	9754	2012	18,301
North direction	50,847	5103	10,369	2394	19,267

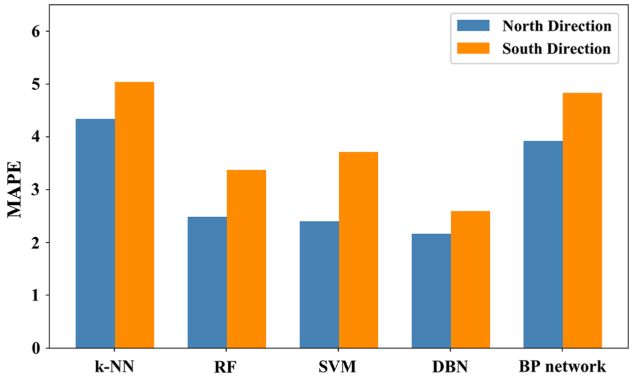


Fig. 7 Prediction results of the five models in peak hours

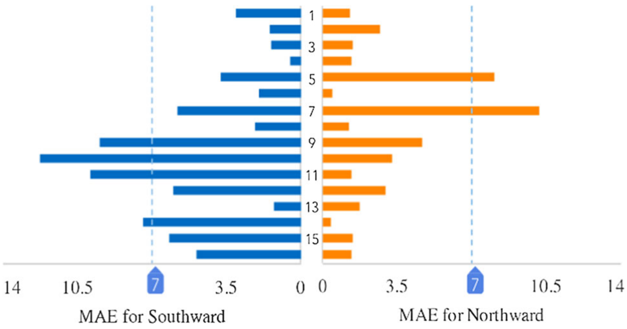


Fig. 8 Prediction errors of the DBN model of different segments

Gaussian–Bernoulli RBMs. In addition, a BP neutral network algorithm was also used to predict bus travel time in a supervised fashion. To validate the proposed model, real-world data from bus route No. 232 in Shenyang, China, were collected. Four other models, including *k*-NN, ANN, SVM and RFs, were also introduced. Results showed that the performance of DBN model was the best among all five travel time prediction models. The performance of the proposed model was also quite good in peak hours. The maximum errors in such a case, in terms of MAE, were only 12.19 in the south direction and 10.22 in the north direction, respectively.

In this paper, only the bus data from a single route are considered as the input variables. Further study should consider more factors that might affect prediction accuracy, such as weather conditions, the running time of other bus routes or vehicles on the same segment, and the environment of signalized intersections to enhance the performance of the proposed bus travel time prediction model. In addition, with the development of computer technology, multiple types of deep learning models have been proposed. In the future, the comparison among different deep learning models will be conducted to further prove the validity of the proposed method.

Acknowledgements This work was supported in National Natural Science Foundation of China (U1811463 and 51578112), The State Key Laboratory of Structural Analysis for Industrial Equipment (S18307). Finally, the authors gratefully acknowledge financial support from China Scholarship Council.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Classical RBM

RBM is a special kind of generative energy-based model that can learn a probability distribution over a set of inputs. A classical RBM has binary valued hidden and visible units. And the energy of a joint configuration (v, h) of the visible and hidden units can be obtained by:

$$E(v,h) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{k} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{k} v_i h_j w_{ij} \tag{16}$$

where v_i and h_j are the binary states of visible unit i and hidden unit j, a_i and b_j are their biases and w_{ij} is the weight. Then, the probability that is assigned to every possible pair of a visible and a hidden vector is calculated via the energy function:

$$p(v,h) = \frac{\mathrm{e}^{-E(v,h)}}{\sum_{v,h} \mathrm{e}^{-E(v,h)}} \tag{17}$$

Then, the probability of a particular visible state configuration v is derived by summing over all possible hidden vectors:

$$p(v) = \sum_{h} p(v,h) = \frac{\sum_{h} \mathrm{e}^{-E(v,h)}}{\sum_{v,h} \mathrm{e}^{-E(v,h)}} \tag{18}$$

Similarly, the formula of $p(h)$ is entirely analogous to that of $p(v)$:

$$p(h) = \sum_{v} p(v,h) = \frac{\sum_{v} \mathrm{e}^{-E(v,h)}}{\sum_{v,h} \mathrm{e}^{-E(v,h)}} \tag{19}$$

Some other conditional expressions can also be derived as follows:

$$p(v|h) = \frac{p(v,h)}{p(h)} = \frac{\mathrm{e}^{-E(v,h)}}{\sum_v \mathrm{e}^{-E(v,h)}} \tag{20}$$

$$p(h|v) = \frac{p(v,h)}{p(v)} = \frac{\mathrm{e}^{-E(v,h)}}{\sum_h \mathrm{e}^{-E(v,h)}} \tag{21}$$

Thus, the probability of a particular visible unit being on given a hidden configuration, i.e., the state of a visible node, given a hidden vector, is derived by:

$$p(v_i = 1|h) = \frac{p(v_i = 1, h)}{p(h)} = \frac{1}{1 + \mathrm{e}^{-\left(a_i + \sum_{j=1}^{k} h_j w_{ij}\right)}} \tag{22}$$

Similarly, for randomly selected training input v, the binary state h_j of each hidden unit j is set to 1 with probability:

$$p\left(h_j = 1|v\right) = \frac{p\left(h_j = 1, v\right)}{p(h)} = \frac{1}{1 + \mathrm{e}^{-\left(b_j + \sum_{i=1}^{m} v_i w_{ij}\right)}} \tag{23}$$

Given $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$, formulas (22) and (23) can be rewritten as follows:

$$p(v_i = 1|h) = \sigma\left(a_i + \sum_{j=1}^{k} h_j w_{ij}\right) \tag{24}$$

$$p\left(h_j = 1|v\right) = \sigma\left(b_j + \sum_{i=1}^{m} v_i w_{ij}\right) \tag{25}$$

Given a set of C training cases $\{v^c | c \in \{1, \ldots, C\}\}$, the goal is to maximize the average log probability of the set under the model's distribution:

$$\sum_{c=1}^{C} \log p(v^c) = \sum_{c=1}^{C} \log \frac{\sum_h \mathrm{e}^{-E(v^c,h)}}{\sum_{v,h} \mathrm{e}^{-E(v,h)}} \tag{26}$$

Then, the gradient or the derivative of the log probability of the training vector with respect to a weight w_{ij} has the following form:

$$\frac{\partial}{\partial w_{ij}} \sum_{c=1}^{C} \log p(v^c) = \frac{\partial}{\partial w_{ij}} \left(\sum_{c=1}^{C} \log \sum_h \mathrm{e}^{-E(v^c,h)} - \log \sum_{v,h} \mathrm{e}^{-E(v,h)} \right) \tag{27}$$

The first term of formula (26) can be written as:

$$\frac{\partial}{\partial w_{ij}} \sum_{c=1}^{C} \log \sum_h \mathrm{e}^{-E(v^c,h)} = -\sum_{c=1}^{C} \frac{\sum_h \mathrm{e}^{-E(v^c,h)} v_i^c h_j}{\sum_h \mathrm{e}^{-E(v^c,h)}} \tag{28}$$

Notice that the term $\frac{\sum_h \mathrm{e}^{-E(v^c,h)} v_i^c h_j}{\sum_h \mathrm{e}^{-E(v^c,h)}}$ is just the expected value of $v_i^c h_j$ given that v is clamped to the data vector v^c. This is easy to compute since we know v_i^c and we can compute the expected value of h_j using formula (25).

The second term of formula (27) can also be written as:

$$\frac{\partial}{\partial w_{ij}} \sum_{c=1}^{C} \log \sum_{v,h} \mathrm{e}^{-E(v,h)} = -\sum_{c=1}^{C} \frac{\sum_{v,h} \mathrm{e}^{-E(v,h)} v_i h_j}{\sum_{v,h} \mathrm{e}^{-E(v,h)}} \tag{29}$$

Here, the term $\frac{\sum_{v,h} \mathrm{e}^{-E(v,h)} v_i h_j}{\sum_{v,h} \mathrm{e}^{-E(v,h)}}$ is the expected value of $v_i h_j$ under the model's distribution. This expectation can be approximated well in finite time by the contrastive divergence (CD) algorithm.

By using $\langle . \rangle_d$ and $\langle . \rangle_m$ to represent the expected values of the training data and model, respectively, formula (27) can be rewritten.

$$\frac{\partial}{\partial w_{ij}} \log p(v) = \left\langle v_i h_j \right\rangle_d - \left\langle v_i h_j \right\rangle_m \tag{30}$$

Thus, the update rule for weight w_{ij} is shown as follows:

$$\Delta w_{ij} = \varepsilon\left(\left\langle v_i h_j \right\rangle_d - \left\langle v_i h_j \right\rangle_m \right) \tag{31}$$

where ε is the learning rate.

The update rules for the biases are similarly derived to be:

$$\Delta v_i = \varepsilon\left(\langle v_i \rangle_d - \langle v_i \rangle_m \right) \tag{32}$$

$$\Delta h_j = \varepsilon\left(\left\langle h_j \right\rangle_d - \left\langle h_j \right\rangle_m \right) \tag{33}$$

Gaussian–Bernoulli RBM

The classical RBM was developed only using binary logistic units for visible and hidden units; in this paper for the traffic data that are continuous, a conversion to continuous-valued inputs is used as described in Refs. [42, 47]. To model continuous data, the binary visible units of RBM are replaced by linear units with Gaussian noise, and then the energy function of GBRBM becomes:

$$E(v,h) = -\sum_{i=1}^{m} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{k} b_j h_j - \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{v_i}{\sigma_i} W_{ij} h_j \tag{34}$$

where σ_i is the standard deviation of the Gaussian noise for visible unit i.

Given the energy function (34), the distribution $p(v|h)$ can be derived as follows:

$$\begin{aligned} p(v|h) &= \frac{\mathrm{e}^{-E(v,h)}}{\int_v \mathrm{e}^{-E(v,h)} \mathrm{d}v} = \frac{\mathrm{e}^{-\sum_{i=1}^{m} \frac{(v_i - a_i)^2}{2\sigma_i^2} + \sum_{j=1}^{k} b_j h_j + \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{v_i}{\sigma_i} W_{ij} h_j}}{\int_v \mathrm{e}^{-\sum_{i=1}^{m} \frac{(v_i - a_i)^2}{2\sigma_i^2} + \sum_{j=1}^{k} b_j h_j + \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{v_i}{\sigma_i} W_{ij} h_j} \mathrm{d}v} \\ &= \prod_{i=1}^{m} \frac{1}{\sigma_i \sqrt{2\pi}} \cdot \mathrm{e}^{-\frac{1}{2\sigma_i^2}\left(v_i - a_i - \sigma_i \left(\sum_{j=1}^{k} W_{ij} h_j \right) \right)^2} \end{aligned} \tag{35}$$

Thus, $p(h_k = 1|v)$ is computed as follows.

$$
\begin{aligned}
p(h_k = 1|v) &= \frac{\sum_{h_j \neq k} p(v, h_k = 1, h_{j\neq k})}{p(v)} \\
&= \frac{\sum_{h_j \neq k} \mathrm{e}^{\left(\sum_{i=1}^{m} \frac{v_i}{\sigma_i} w_{ik} + b_j\right) + \left(\sum_{i=1}^{m} \sum_{j\neq k}^{k} \frac{v_i}{\sigma_i} W_{ij} h_j + \sum_{i=1}^{m} \frac{(v_i - a_i)^2}{2\sigma_i^2} + \sum_{j\neq k}^{k} h_j b_j\right)}}{\sum_h \mathrm{e}^{-E(v,h)}} \\
&= \frac{1}{1 + \mathrm{e}^{-\left(\sum_{i=1}^{m} \frac{v_i}{\sigma_i} w_{ik} + b_j\right)}}
\end{aligned}
\tag{36}
$$

Note that Eq. (36) is the same as in the classical RBM except the v_i scaled by the reciprocal of its standard deviation σ_i.

The training procedure for a GBRBM is identical to that of an RBM. As in that case, we take the derivative shown in formula (27). We find that

$$
\begin{aligned}
\frac{\partial}{\partial w_{ij}} \sum_{c=1}^{C} \log \sum_h \mathrm{e}^{-E(v^c,h)} &= -\sum_{c=1}^{C} \frac{\sum_h \mathrm{e}^{-E^{(v^c,h)}} \frac{\partial E(v^c,h)}{\partial w_{ij}}}{\sum_h \mathrm{e}^{-E(v^c,h)}} \\
&= -\frac{1}{\sigma_i} \sum_{c=1}^{C} \frac{\sum_h \mathrm{e}^{-E(v^c,h)} v_i^c h_j^c}{\sum_h \mathrm{e}^{-E(v^c,h)}}
\end{aligned}
\tag{37}
$$

Similarly,

$$
\frac{\partial}{\partial w_{ij}} \sum_{c=1}^{C} \log \sum_v \sum_h \mathrm{e}^{-E(v,h)} = -\frac{1}{\sigma_i} \sum_{c=1}^{C} \frac{\sum_v \sum_h \mathrm{e}^{-E(v,h)} v_i h_j}{\sum_v \sum_h \mathrm{e}^{-E(v,h)}}
\tag{38}
$$

which we estimate, as before, using CD algorithm.

References

1. Petersen NC, Rodrigues F, Pereira FC (2019) Multi-output bus travel time prediction with convolutional LSTM neural network. Expert Syst Appl 120:426–435
2. Yu B, Lam WH, Tam ML (2011) Bus arrival time prediction at bus stop with multiple routes. Transp Res C-Emerg 19(6):1157–1170
3. Yu B, Wang HZ, Shan WX, Yao BZ (2018) Prediction of bus travel time using random forests based on near neighbors. Comput-Aided Civ Inf 33(4):333–350
4. Williams BM, Hoel LA (2003) Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. J Transp Eng 129(6):664–672
5. Jeong RH (2005) The prediction of bus arrival time using automatic vehicle location systems data. Ph.D. dissertation, Department of Civil Engineering, Texas A&M University, Texas, USA
6. Chung EH, Shalaby A (2007) Expected time of arrival model for school bus transit using real-time global positioning system-based automatic vehicle location data. Int J Intell Transp Syst Res 11(4):157–167
7. Yang M, Liu Y, You Z (2010) The reliability of travel time forecasting. IEEE Trans Intell Transp Syst 11(1):162–171
8. Thomas T, Weijermars W, Berkum EV (2010) Predictions of urban volumes in single time series. IEEE Trans Intell Transp Syst 11(1):71–80
9. Rice J, Van Zwet E (2004) A simple and effective method for predicting travel times on freeways. IEEE Trans Intell Transp Syst 5(3):200–207
10. Kwon J, Coifman B, Bickel P (2000) Day-to-day travel-time trends and travel-time prediction from loop-detector data. Transp Res Rec 1717:120–129
11. Kwon J, Petty K (2005) Travel time prediction algorithm scalable to freeway networks with many nodes with arbitrary travel routes. Transp Res Rec 1935:147–153
12. You J, Kim TJ (2000) Development and evaluation of a hybrid travel time forecasting model. Transp Res C-Emerg 8(1):231–256
13. Smith BL, Williams BM, Oswald RK (2002) Comparison of parametric and nonparametric models for traffic flow forecasting. Transp Res C-Emerg 10(4):303–321
14. Chang H, Park D, Lee S, Lee H, Baek S (2010) Dynamic multi-interval bus travel time prediction using bus transit data. Transportmetr A 6(1):19–38
15. Adeli H (2001) Neural networks in civil engineering: 1989–2000. Comput-Aided Civ Inf 16(2):126–142
16. Chien SIJ, Ding Y, Wei C (2002) Dynamic bus arrival time prediction with artificial neural networks. J Transp Eng 128(5):429–438
17. Mazloumi E, Currie G, Rose G (2010) Using traffic flow data to predict bus travel time variability through an enhanced artificial neural network. In: Presented at the 12th world Congress on transport research. Lisbon, Portugal
18. Yao BZ, Chen C, Zhang L, Yu B, Wang YP (2019) Allocation method for transit lines considering the User Equilibrium for operators. Transp Res C-Emerg 105:666–682
19. Yu B, Ye T, Tian XM, Ning GB, Zhong SQ (2012) Bus travel-time prediction with a forgetting factor. J Comput Civ Eng 28(3):06014002
20. Gal A, Mandelbaum A, Schnitzler F, Senderovich A, Weidlich M (2017) Traveling time prediction in scheduled transportation with journey segments. Inf Syst 64:266–280
21. Yu B, Song XL, Guan F, Yang ZM, Yao BZ (2016) *k*-Nearest neighbor model for multiple-time-step prediction of short-term traffic condition. J Transp Eng-ASCE 142(6):04016018
22. Yao BZ, Chen C, Cao QD, Jin L, Zhang MH, Zhu HB, Yu B (2017) Short-term traffic speed prediction for an urban corridor. Comput-Aided Civ Inf 32(2):154–169
23. Reddy KK, Kumar BA, Vanajakshi L (2016) Bus travel time prediction under high variability conditions. Curr Sci 111(4):700
24. Wang WS, Liu JM, Yao BZ, Jiang YL, Wang YP, Yu B (2019) A data-driven hybrid control framework to improve transit performance. Transp Res C-Emerg 107:387–410
25. Chen M, Liu X, Xia J, Chien SI (2004) A dynamic bus arrival time prediction model based on APC data. Comput-Aided Civ Inf 19(5):364–376
26. Yu B, Yang ZZ, Chen K, Yu B (2010) Hybrid model for prediction of bus arrival times at next station. J Adv Transp 44(3):193–204
27. Shalaby A, Farhan A (2003) Bus travel time prediction model for dynamic operations control and passenger information systems. In: Presented at the 82nd TRB annual meeting, Washington D.C., USA
28. Vanajakshi L, Rilett LR (2007) Support vector machine technique for the short term prediction of travel time. In: Proceedings of intelligent vehicles symposium. Istanbul, Turkey, pp 600–605
29. Billings D, Yang JS (2006) Travel time prediction using a seasonal autoregressive integrated moving average time series model. In: International conference on systems, man, and cybernetics. Taipei, Taiwan, pp 2529–2534
30. Guin A (2006) Application of the ARIMA models to urban roadway travel time prediction-a case study. In: Intelligent

transportation systems conference. Toronto, Ontario, Canada, pp 494–498

31. Kumar P, Sehgal V, Chauhan DS (2011) Performance evaluation of decision tree versus artificial neural network based classifiers in diversity of datasets. In: World congress on information and communication technologies (WICT). Mumbai, India, pp 798–803
32. Kumar P, Sehgal VK, Chauhan DS (2012) A benchmark to select data mining based classification algorithms for business intelligence and decision support systems. Int J Data Min Knowl Manage. Process (IJDKP) 2(5):25–42
33. Weigend A (1993) On overfitting and the effective number of hidden units. Department of Computer Science, University of Colorado, Boulder, Colorado, USA, CU-CS-674-93
34. Li DQ, Fu BW, Wang YP, Lu GQ, Berezin Y, Stanley HE, Havlin S (2015) Percolation transition in dynamical traffic network with evolving critical bottlenecks. Proc Natl Acad Sci USA 112(3):669–672
35. Tang TQ, Shi YF, Wang YP, Yu GZ (2012) A bus-following model with an on-line bus station. Nonlinear Dyn 70(1):209–215
36. Sun D, Ni X, Zhang L (2016) A discriminated release strategy for parking variable message sign display problem using agent-based simulation. IEEE Trans Intell Transp Syst 17(1):38–47
37. Lv Y, Duan Y, Kang W, Li Z, Wang FY (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16(2):865–873
38. Siripanpornchana C, Panichpapiboon S, Chaovalit P (2016) Travel-time prediction with deep learning. In: Region 10 conference (TENCON), Singapore. Singapore, pp 1859–1862
39. Li LC, Qu X, Zhang J, Wang YG, Ran B (2019) Traffic speed prediction for intelligent transportation system based on a deep feature fusion model. J Intell Transp Syst. https://doi.org/10.1080/15472450.2019.1583965
40. Ran X, Shan Z, Fang Y, Lin C (2019) An LSTM-based method with attention mechanism for travel time prediction. Sensors 19(4):861
41. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
42. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
43. Huang W, Song G, Hong H, Xie K (2014) Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans Intell Transp Syst 15(5):2191–2201
44. Koesdwiady A, Soua R, Karray F (2016) Improving traffic flow prediction with weather information in connected cars: a deep learning approach. IEEE Trans Veh Technol 65(12):9508–9517
45. Soua R, Koesdwiady A, Karray F (2016) Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. In: International joint conference on neural networks. Vancouver, BC, Canada, pp 3195–3202
46. Hrasko R, Pacheco AG, Krohling RA (2015) Time series prediction using restricted Boltzmann machines and backpropagation. Proc Comput Sci 55:990–999
47. Hinton GE (2012) A practical guide to training restricted Boltzmann machines. In: Neural networks: tricks of the trade. Springer, Berlin, Heidelberg, pp 599–619
48. Kumar A, Johari S, Proch D, Kumar P, Chauhan DS (2018) A tree based approach for data pre-processing and pattern matching for accident mapping on road networks. Proc Natl Acad Sci India Sect A Phys Sci 89(3):453–466

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.