Alexandria Engineering Journal (2020) xxx, xxx–xxx

HOSTED BY

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

An improved deep belief network for traffic prediction considering weather factors

Xuexin Bao *, **Dan Jiang, Xuefeng Yang, Hongmei Wang**

School of Shipping and Naval Architecture, Chongqing Jiaotong University, Chongqing 400074, China

Received 1 July 2020; revised 28 August 2020; accepted 4 September 2020

KEYWORDS

Traffic prediction;
Deep learning;
Support vector regression (SVR);
Deep belief network

Abstract The timely access to accurate traffic data is essential to the development of intelligent traffic systems. However, the existing traffic prediction methods cannot achieve satisfactory results, mainly because of three factors: the structure is too simple to extract deep features; many external factors are overlooks, such as weather and traffic incidents; the nonlinearity of traffic flow is not well handled. To solve the problem, this paper improves the deep belief network (DBN), a deep learning method, for accurate traffic prediction under poor weather. Firstly, the data of poor weather and traffic data were collected from IoV, rather than induction coils in traditional methods. Next, the support vector regression (SVR) was introduced to improve the classic DBN. In the improved DBN, the underlying structure is a traditional DBN that learns the key features of traffic data in an unsupervised manner, and the top layer is an SVR that performs supervised traffic prediction. To verify its effectiveness, the improved DBN was applied to predict the traffic data based on the traffic data from the control center of an expressway and the weather data from local monitoring stations, in comparison with the autoregressive integrated moving average (ARIMA) model and the traditional neural network. The experimental results show that the improved DBN controlled the traffic prediction error within 9%, and maintained good robustness despite the extension of the time interval. To sum up, this paper provides an effective way to predict traffic flow under poor weather, shedding new light on the application of deep learning in traffic prediction.

© 2020 Production and hosting by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The timely access to accurate traffic data offers great benefits to tourists, as well as enterprises and government agencies engaged in tourism, such as rationalizing travel decisions, facilitating route guidance, alleviating traffic congestion and reducing carbon emissions [1–4]. With the advent of intelligent transportation system (ITS), more and more attention has been paid to traffic prediction, which is the basis for traffic control and guidance. Traffic prediction is widely regarded as the key to developing the ITS and similar systems like advanced traffic management system (ATMS) and advanced traffic information system (ATIS) [5,6]. Traffic prediction mainly focuses on the short term, aiming to estimate the number of vehicles in a specific area within a time interval. The

* Corresponding author.
E-mail address: xuexin_bao@163.com (X. Bao).
Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2020.09.003
1110-0168 © 2020 Production and hosting by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

traffic is mainly forecasted based on the historical data and the real-time data collected by various sensors, namely, induction coil, radar, camera and mobile GPS sensors. Recent years has seen the proliferation of traditional traffic sensors, marking the dawn of the era of big data transportation [7]. Traffic data have become increasingly important in traffic management and control.

Currently, there are many shallow models of traffic prediction, which are designed for normal conditions. Without considering external factors (e.g. weather), these models often output predictions that deviate greatly from theoretical results, failing to forecast traffic flows in a satisfactory manner. To solve the problem, a viable option is to apply deep learning algorithms to traffic prediction, in the light of the abundance of traffic data collected by various sensors. Some scholars [8] pointed out that traffic flows are essentially chaotic, especially in traffic jams. However, some other scholars argued that traffic data are stochastic rather than chaotic, and can thus be modeled and predicted by stochastic prediction methods.

Traffic prediction has been a research hotspot in recent decades. The popular approaches for traffic prediction are driven either by model(s) or data.

The model-driven methods, a.k.a. parametric methods, mainly include historical average model [9], time series model [10], and Kalman filter [11]. For example, the auto-regressive moving average (ARMA) model [12] assumes the traffic flow as a nonstationary stochastic sequence with three or six parameters, and makes traffic predictions based on this sequence. The effectiveness of model-driven methods hinges on the practicality of the theoretical assumptions, which are often not fully established due to the complexity of actual traffic. Therefore, it is difficult for these methods to perform well in actual practices.

The data-driven methods attempt to interpret and learn the big data of transportation, and implement them in predicting short-term traffic. The typical examples are artificial neural network (ANN) [13], support vector regression (SVR) [14] and fuzzy neural network (FNN) [15,16]. The most important data-driven method is the ANN. Traditional ANNs, such as the feedforward neural network with backpropagation algorithm, has been extensively used for short-term traffic prediction [17–19]. There are several defects with the data-driven methods. For instance, these methods are not good at processing high-dimensional data, and have difficulty in depicting the complex and nonlinear changes in actual traffic data. The parameter weights of these methods are mainly selected empirically by experts. Fortunately, the emerging theory of deep learning has offered a good way to mine complex features out of high-dimensional data.

To date, deep learning has been successfully applied in many industrial fields, ranging from task classification, natural language recognition, environmental perception, motion modeling to human behavior recognition. Deep learning algorithms enjoy a unique advantage in traffic prediction, thanks to their ability to extract essential features in massive data from bottom up, using their multi-layer structure or deep structure. Despite the complexity of traffic data, these algorithms can acquire the traffic features without requiring prior knowledge. Much research attention has also been paid to the weather influence on traffic prediction. Maze et al. [20] demonstrated that rain, snow, fog, and wind might suppress traffic demand, reducing driving safety and traffic flow. Asamer and Reinthaler [21] put forward a method to estimate urban road conditions under poor weather based on traffic data measured by sensors. The *Highway Capacity Manual* (HCM) [22] provides a fixed attenuation ratio of prediction capacity to estimate the impact of bad weather on traffic capacity of urban roads. However, the attenuation ratio actually varies from region to region, due to the regional difference in weather. Kwon et al. [23] proved that, in many cases, the HCM method underestimates or overestimates the actual weather effects in many cases. Similar to the HCM, Colyar et al. [24] proposed a fixed weather-impact factor after simulating different weather conditions. Alhassan and Ben-Edigbe [25] empirically analyzed the rain-induced loss of highway capacity. Hou et al. [26] introduced poor weather factors into mesoscopic network simulation, and thus revised the traffic prediction model. Lam et al. [27] simulated the impact of rainfall intensity on traffic flow, and calibrated the traffic prediction model based on the hourly rainfall of Hong Kong. In most of the above methods, the traffic prediction parameters are only modified for normal weather conditions. The modified parameters cannot reflect the complex impacts of weather factors on traffic.

The existing traffic prediction approaches are not suitable for poor weather. Firstly, most of them adopt a shallow structure. For example, there is only one hidden layer in the ANN. The neural networks with multiple hidden layers are very likely to fail in training. Secondly, some methods require artificially designed features, which are lengthy and error-prone. These methods cannot extract or select features without the prior knowledge in relevant fields. Thirdly, the traffic data are separated from the external factors like weather, traffic environment and road conditions. The road conditions are estimated solely based on previously observed traffic data. In fact, the elements in the traffic system are closely correlated. The shared knowledge in the system should be adopted to enhance the accuracy of traffic prediction.

Through the above analysis, this paper develops an improved deep learning structure for traffic prediction under poor weather, based on deep belief network (DBN) and the SVR. The parameters and prediction effect of the proposed structure were respectively calibrated and evaluated based on the data were provided by the control center of an expressway. The main contributions of this research are summarized as follows:

(1) To the best of our knowledge, this is the first attempt to apply deep learning in traffic prediction under poor weather. Our approach applies to complex traffic systems with massive data and high-dimensional features.
(2) Our approach can learn multiple related tasks at the same time, achieving a high accuracy in traffic prediction. Fully integrating an SVR layer, the improved DBN outperforms the classic traffic prediction methods that process one task at a time.
(3) The weather factors, together with their correlations, are fully considered in traffic prediction. In this way, the traffic flow under poor weather can be forecasted accurately.

The rest of this paper is organized as follows: Section 2 presents a new mechanism to process traffic data; Section 3 designs an improved deep learning method based on the

Please cite this article in press as: X. Bao et al., An improved deep belief network for traffic prediction considering weather factors, Alexandria Eng. J. (2020), https://doi.org/10.1016/j.aej.2020.09.003

DBN and the SVR; Section 4 verifies the proposed method through comparative experiments; Section 5 puts forward the conclusions and highlights the directions of future research.

2. Traffic data processing mechanism

Induction coils can only collect a limited amount of data on traffic flow and surrounding environment. What is worse, there are not many autonomous learning methods for these data. Therefore, it is difficult to predict the traffic flow accurately through intelligent learning of these data. To overcome the difficulty, this paper develops a traffic data processing mechanism, which relies on big data and the Internet of vehicles (IoV). Both the external impact data and the IoV data were adopted for learning and prediction [28,29]. The external impact data were set reasonably, while the number of vehicles was obtained from the GPS data in the IoV data files [30].

2.1. Data preprocessing mechanism

Let t be the time interval (min); n be daily acquisition frequency of the GPS data of the IoV ($n \in N$ is a positive integer). The acquired GPS data contain various types of data, including the longitude and latitude of each vehicle. Based on the GPS data, it is possible to predict the number of vehicles *Cn p* passing through an area j in the nth prediction cycle:

$$C_j^n = C_{j-0}^n + C_{j-i}^n + C_{j-o}^n \tag{1}$$

where, *Cn j-0* is the number of vehicles in area j in the nth and n-1th prediction cycles; *Cn j-I* is the number of vehicles in area j in the nth prediction cycle but not in that area in the n-1th prediction cycle; C_{j-o}^n is number of vehicles in area j not in area j in the nth prediction cycle but in that area in the n-1th prediction cycle.

The expressions of *Cn j-0*, *Cn j-i*, and *Cn j-o* are as follows:

$$\begin{cases} C_{j-0}^n = \sum\limits_{gps \in G} \left| \left(gps_n \in T_j\right) \cap \left(gps_n \in T_j\right) \right| \\ C_{j-i}^n = \sum\limits_{gps \in G} \left| \left(gps_n \in T_j\right) \cap \left(gps_{n-1} \notin T_j\right) \right| \\ C_{j-o}^n = \sum\limits_{gps \in G} \left| \left(gps_n \notin T_j\right) \cap \left(gps_{n-1} \in T_j\right) \right| \end{cases} \tag{2}$$

where, *gps* is the GPS data of a vehicle; G is the GPS data of all vehicles; T_j is area j; gps_n is the GPS data of vehicles in the nth prediction cycle.

2.2. Processing of weather data files

The external impact data were divided into two parts: the weather conditions (1 for poor weather; 0 for good weather) and whether the current day is a holiday (1 for holiday; 0 for non-holiday). Note that poor weather refers to hail, fog, snowstorm, etc., while holidays include both weekends and legal holidays. The rainfall affects the traffic capacity of the road network to a certain extent. High rainfall reduces the visibility of the road, thus affecting the traffic capacity. In addition to the impact on the speed of traffic, snowfall can also affect the traffic capacity. Heavy snowfall can make roads surface ice, and reduce visibility, thus reducing the traffic capacity of roads. The most noticeable effect of heavy fog on traffic is the reduction in traffic speed. The decrease of traffic speed also leads to the increase of travel time and travel delay, which directly affects the traffic flow. According to HCM 2000, the poor weather is defined as: rainfall > 0.1 in./h, snowfall > 0.05 in./h, or Fog Visibility < 1 miles.

Thus, the output of the data module is in the following format: (y_T, y_J, y_N, y_R, y_S, M). The value of y_N depends on the benchmark year. The quantized value of the benchmark year was set to 1. Then, the quantized value of a year equals 1 plus the difference between that year and the benchmark year.

The value of y_R equals the serial number of the week in the year. The quantized value of the first week in the year is 1, that of the second week, 2... and that of the nth week, n.

The value of y_S was set according to the prediction cycle. Each day was divided into $1{,}440/T$ prediction cycles. The quantized value of the first prediction cycle is 1, that of the second prediction cycle, 2... the rest can be deduced by analogy.

3. Improved DBN for traffic prediction

3.1. Traffic prediction

Traffic prediction aims to estimate the number of vehicles on the road in the next few time intervals. The interval is usually set to 5 to 30 min. Let $f_{i,t}$ be the traffic flow at the ith observation point in the tth time interval. The task at time T is to predict the traffic flow $f_{i,T+1}$ at time $T + 1$ based on the historical traffic flow sequence $F = \{f_{i,t} | \mathrm{i} \in O, \mathrm{t} = 1,2,\ldots,T\}$, where O is the set of observation points. In some studies, the main prediction target is the traffic flow in the next time intervals from $T + 1$ to $T + n$.

Generally, the traffic flow can be predicted in two steps, namely, feature learning and model learning. Feature learning extracts and selects the most representative features from the historical traffic flow sequence F by learning a feature representation model g. Through feature learning, the traffic flow sequence is mapped into the feature space $g(F) \to X$. The prediction task $f_{i,T+1}$ can be expressed as Y.

In some strategies, feature learning is often designed artificially. Some important traffic factors, such as speed, vehicle volume and vehicle density, are calculated from the original data as prediction features. In addition, most traffic prediction methods solely rely on time series features. For example, the autoregressive integrated moving average (ARIMA) model only uses the traffic data before a specific observation point j:

$$\begin{aligned} \forall y_j &= f_{j,T+1}, \\ x_j &= \left\{ f_{j,t} | t = T, T-1, \ldots, T-m+1 \right\} \end{aligned} \tag{3}$$

where, m is the time step in the ARIMA model. Without considering any additional data, the ARIMA model cannot describe the impacts of bad weather. Feature learning generates appropriate features, serving as the inputs to the prediction model.

Model learning is a process of supervised learning. According to the pair of feature X and task Y obtained from the historical traffic flow, the optimal parameters of the prediction model $Y = h(X)$ are learned to minimize the loss function:

$$L\left(Y, \hat{Y}\right) = \frac{1}{2}\left(Y - \hat{Y}\right)^2 \tag{4}$$

The traffic prediction model may take different forms, but the objective functions are always similar.

Please cite this article in press as: X. Bao et al., An improved deep belief network for traffic prediction considering weather factors, Alexandria Eng. J. (2020), https://doi.org/10.1016/j.aej.2020.09.003

3.2. Improved DBN traffic prediction model under poor weather

To disclose the traffic impacts of different weather factors, this paper proposes a correlation coefficient to evaluate how much the weather affects the road traffic. The correlation coefficient $r(x_w,y_t)$ between weather and traffic flow can be described by a 2D relationship matrix, which consists of pairs of processed traffic flow y_t and weather x_w:

$$r(x_w,y_t)=\left|\frac{n\sum x_w y_t-\sum x_w\sum y_t}{\sqrt{n\sum x_w^2-(\sum x_w)^2}\sqrt{n\sum y_t^2-(\sum y_t)^2}}\right| \tag{5}$$

where, n is the number of collected data entries; $r(x_w, y_t)$ is a dimensionless term independent from the unit of y_t and x_w. The value of $r(x_w,y_t)$ falls within [0, 1].

(1) The DBN

With multiple hidden layers, deep learning can utilize a large number of parameters before inducing overfitting. In deep learning, the multilayer neural network learns the target features through unsupervised pre-training with lots of untagged data. Then, the learned features are adjusted through supervised fine-tuning based on tagged data, laying a good basis for prediction.

The DBN is one of the most popular and effective deep learning models. It is a stacking of various restricted Boltzmann machines (RBMs). As shown in Fig. 1, each RBM has only one hidden layer. The hidden layer output of a trained RBM serves as the input of the RBM on the next layer. In this way, the RBMs in the entire DBN can be learned layer by layer.

The RBM, as a special Markov random field, can be regarded as an energy-based model. The visible layer v of the RBM is linked up with the random hidden layer h via an undirected weighted connection. There is no connection within the visible or hidden layer. Therefore, the RBM structure is a typical undirected graph. The high-order correlation of the visible layer is outputted by the hidden layer. For a binary RBM, the probability distribution model $p(v,h;\theta)$ and the energy function $E(v,h;\theta)$ can be respectively expressed as:

$$p(v,h;\theta)=\frac{1}{z(\theta)}\exp(-E(v,h;\theta)) \tag{6}$$

$$\begin{aligned}E(v,h;\theta) &= -\sum_{i=1}^{D}a_i v_i-\sum_{j=1}^{F}b_j h_j-\sum_{i=1}^{D}\sum_{j=1}^{F}w_{ij}v_i h_i\\ &= a^T v-b^T h-v^T W h\end{aligned} \tag{7}$$

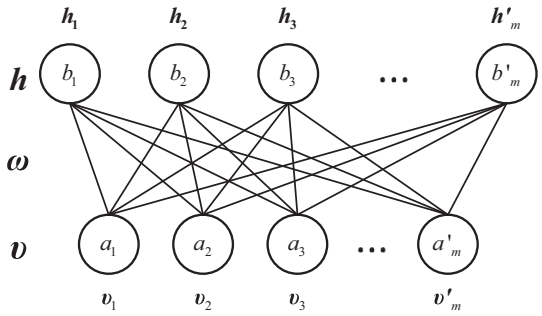

Fig. 1 The structure of the RBM.

where, $z(\theta)$ is the regularization factor; v is the input vector of the visible layer; h is the output vector of the hidden layer; w is the weight vector; $\theta=(w,a,b)$ is the set of internal parameters; a and b are the biases of the visible layer and the hidden layer, respectively. The parameters of each unit are defined as follows:

$$\begin{cases} v = (v_1, v_2, \ldots, v_{i'})^T \; i' \in N* \\ h = (h_1, h_2, \ldots, h_{j'})^T \; j' \in N* \\ \omega = (\omega_{i'j'})^T \; i' \in [1,m'], j' \in [1,n'] \\ a = (a_1, a_2, \ldots, a_{m'})^T \\ b = (b_1, b_2, \ldots, b_{n'})^T \end{cases} \tag{8}$$

Let $|V|$ and $|H|$ be the number of nodes in the visible layer and the hidden layer, respectively. The conditional probability distributions of the visible layer and the hidden layer can be respectively calculated by:

$$p(h_j|v;\theta)=\mathrm{sigm}\left(a_j+\sum_{i=1}^{|V|}v_i w_{ij}\right) \tag{9}$$

$$p(v_j|h;\theta)=sigm\left(b_i+\sum_{j=1}^{|H|}h_j w_{ij}\right) \tag{10}$$

where, the sigmoid function $\mathrm{sigm}(x)=(1/(1+e^{-x}))$ is the activation function; $\theta=(w,a,b)$ is the internal parameter vector obtained by training the k-step contrastive divergence (CD-k) algorithm (Table 1).

Next, several RBMs can be stacked into a DBN. The joint probability distribution of the visible layer v can be expressed as:

$$\begin{aligned}p(v;\theta) &= \sum_h \frac{e^{-E(v,h;\theta)}}{\sum_{v,h}e^{-(v,h;\theta)}}\\ &= \frac{1}{z(\theta)}\sum_h \exp\left(v^T w h+b^T v+a^T h\right)\\ &= \frac{1}{z(\theta)}e^{(b^T v)}\prod_{j=1}^{F}\left(1+\exp\left(a_j+\sum_{i=1}^{D}w_{ij}v_i\right)\right)\end{aligned} \tag{11}$$

Table 1 The steps of CD-k algorithm.

Algorithm 1: CD-k algorithm for the RBM

Input: Training data: $S=\{x_1, x_2, \cdots, x_i\}$, the learning rate: γ; number of steps k; $v_i(i = 1, 2, \ldots, n)$ and $h_j(j = 1, 2, \ldots, m)$
Output: Internal parameter vector $\theta=(w,a,b)$
1: Randomly initialize internal parameter vector $\theta=(w,a,b)$
2: For every training sample x_1 in S, do
$v^0 \leftarrow v$
for $t = 0, 1, \ldots, k$-1 do
for $i = 1, 2, \ldots, n$, calculate $h_i^t p\left(h_i|\tilde{v}^{(t)}\right)$
for $j = 1, 2, \ldots, m$, calculate $v_i^{t+1}p\left(v_j\middle|h^{(t)}\right)$
for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$
$\Delta w_{ij}=\gamma\times\left(p\left(h_j=1\middle|v^{(0)}\right)\cdot v_i^{(0)}-p\left(h_j=1\middle|v^{(k)}\right)\cdot v_i^{(k)}\right)$,
$w_{ij}=w_{ij}+\Delta w_{ij}$
$\Delta b_j=\gamma\times\left(v_i^{(0)}-v_i^{(k)}\right), b_j=b_j+\Delta b_j$
$\Delta a_i=\gamma\times\left(p\left(h_j=1\middle|v^{(0)}\right)-p\left(h_j=1\middle|v^{(k)}\right)\right)$, $a_i=a_i+\Delta a_i$
End

If no tag is provided, the DBN can be used as a tool for unsupervised feature learning.

(2) The SVR

The SVR relies on a nonlinear kernel function to map input sample to a high-dimensional feature space, and then fits the mapping results through regression. Fig. 2 illustrates the steps of the SVR.

Taking the output of the last RBM layer as the input, the SVR prediction model can be defined as:

$$f(x) = \sum_{n=1}^{N}(a_n - a_n^*)K(x_n, \bar{x}) + b \tag{12}$$

where, x_n is the nth ($n = 1, 2, .., N$) input of the SVM; a_n and $a^* n$ are Lagrange multipliers; $K(x_n, \bar{x})$ is the kernel function; b is the bias.

The selection of kernel function is the key step in the SVR of nonlinear problems. Currently, the most popular kernel functions include linear kernel function, polynomial kernel function, sigmoid kernel function and radial basis function (RBF).

The RBF is selected for this research:

$$K(x_n, \bar{x}) = \exp\left(-\| x_n - \bar{x} \|^2 / 2\delta^2\right) \tag{13}$$

where, δ is the width parameter; $\bar{x}$ is the center.

By introducing relaxation variables $\zeta_i \geq 0$ and $\zeta^* i \geq 0$, the objective function can be expressed as :

$$\min_{b,\zeta,\zeta^*}\left\{\frac{1}{2}\| (a_n - a_n^*) \|^2 + c\sum_{i=1}^{n}(\zeta_i + \zeta_i^*)\right\} \tag{14}$$

The boundary conditions are defined as:

$$\begin{cases} (a_n - a_n^*)^{\mathrm{T}} K(x_n, \bar{x}) + b_i - y_i \leqslant \varepsilon + \zeta_i \ \zeta_i \ \geqslant 0 \\ y_i - (a_n - a_n^*)^{\mathrm{T}} K(x_n, \bar{x}) - b_i \leqslant \varepsilon + \zeta_i^* \ \zeta_i^* \ \geqslant 0 \end{cases} \tag{15}$$

By introducing Lagrange multipliers, the problem can be converted into an optimization problem [31]:

$$\begin{aligned} \max_{\alpha,\alpha^*} & \sum_{i=1}^{n} y_i(\alpha_i^* - \alpha_i) - \varepsilon\sum_{i=1}^{n}(\alpha_i^* + \alpha_i) - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n} \\ & (\alpha_i^* - \alpha_i)\left(\alpha_j^* - \alpha_j\right)\left[K(x_i, \bar{x}) \cdot K(x_j, \bar{x})\right] \end{aligned} \tag{16}$$

$$s.t. \ \sum_{i=1}^{n}(\alpha_i^* - \alpha_i) = 0, \alpha_i, \alpha_i^* \in [0, c]$$

3.3. DBN traffic prediction algorithm with SVR

As shown in Fig. 2, the proposed improved DBN structure for traffic prediction under poor weather mainly includes two parts: unsupervised pre-training and learning of the object features by the DBN, and the supervised fine-tuning by the SVR on the top layer. The structure of the improved DBN is explained in Fig. 2. The weather factors were integrated into the in-depth prediction framework.

As shown in Fig. 3, the input information is usually the original collected data. To predict traffic flow based on neural network, all observation points and weather data were put into a large input space $O|\times k$, where $|O|$ is the number of observation points and k is the number of time intervals.

The improved DBN structure only requires the prior knowledge on the number of previous time intervals in k. Unlike the other neural network-based methods, our approach does not artificially extract or select any feature from the original data. The only step of preprocessing is to normalize the flow rate to [0, 1]. The input vector X can be expressed as $X = \{x_i^t | t \in T, i \in O\}$, where x_i^t is the normalized number of vehicles at observation point i in the time interval t.

In our structure, the DBN is placed at the bottom to learn the features. The nonlinear features are transformed in each layer of the DBN. On the top level of the DBN, the most representative features for data modelling are learned: $H^p = \{h_1^p, h_2^p, \ldots, h_m^p\}$, where p is the top layer; m is the number of features of the top layer. The H^p features are taken as the input vectors for the prediction by the SVR layer on the top of our structure.

To sum up, our structure adopts the DBN to learn the representative and robust features from the input, through multilayer nonlinear feature transform. The DBN-based learning

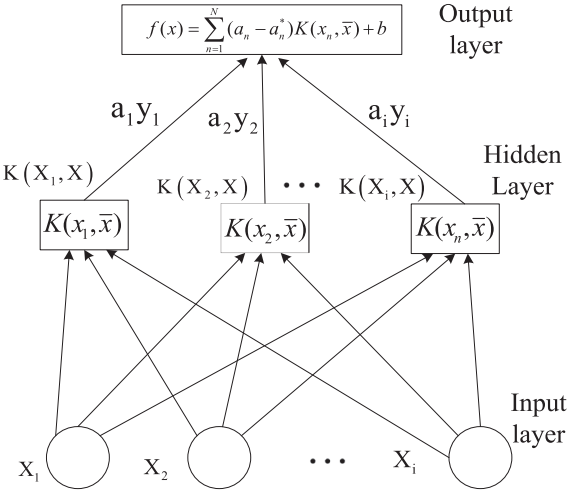

Fig. 2 The process of the SVR.

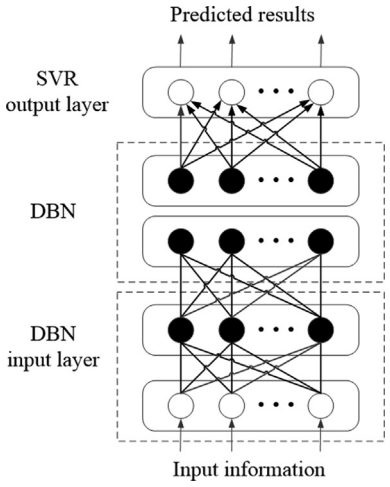

Fig. 3 Structure of improved DBN.

helps to clarify the complex relationship between the features of input data. Next, the SVR was performed to predict the traffic flow based on DBN features.

3.4. Steps of improved algorithm

Step 1: Obtain weather data and traffic data in several consecutive time intervals by the traffic data processing mechanism in Section 2.
Step 2: Randomly initialize the network parameters (ω^i, b^i) ($1 \leq i \leq r + 1$) and kernel functions in the SVR layer.
Step 3: Train the first RBM by the CD-k algorithm, and denote the visible layer and hidden layer of the RBM as v and h_1, respectively.
Step 4: If $1 \leq i \leq r$-1, set up the RBM with h_{i-1} as the ith visible layer and h_i as the ith hidden layer, before training the RBM layer-by-layer with the CD-k algorithm.
Step 5: If $i = r$, set up the RBM with the whole of h_{r-1} and y as the visible layer and h_r as the hidden layer, before training the RBM layer-by-layer with the CD-k algorithm.
Step 6: Take the weights and biases between the trained RBMs as the initial weights and biases of the DBN, and the data tags as supervisory signals to compute network errors. Then, perform the SVR to predict the traffic flow under poor weather through feature learning.

4. Experimental results and analysis

To verify its effectiveness, the improved DBN was applied to predict the traffic data based on the traffic data from the control center of an expressway and the weather data from local monitoring stations, in comparison with the ARIMA and the traditional neural network. The traffic prediction experiments were conducted in Matlab2017b on a computer (Intel Core i7-7820X; RAM 32 G).

The improved DBN and the two contrastive methods were both trained by the traffic data collected in the 122 days between June and September 2018 at intervals of 15 min, and tested by the traffic data recorded on a stormy day in the rainy season of 2019.

The parameters of the improved DBN were configured through empirical calibrations and trial-and-error. Specifically, the number of hidden layers in each RBM was set to 3, the number of nodes on each hidden layer was set to 20, the learning rate was set to 0.2, the number of iterations was set to 350, the sigmoid function was adopted as the activation function, and the penalty factor c of the SVR was set to 0.02.

The three prediction models were compared in terms of prediction error, accuracy and computing time. The prediction error was measured by the mean absolute percentage error (MAPE):

$$MAPE(y, y') = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - y_i'|}{y_i} \tag{17}$$

where, y_i is the real value; y_i is the predicted value.

Figs. 4–6 display the real traffic flow and the predicted traffic flow under the stormy weather of the three methods. Fig. 7 shows the prediction error of each method.

As shown in Figs. 4–7, the traffic flows predicted by all three methods had basically the same trend with the real traffic flow. However, the three methods differed greatly in prediction accuracy. The ARIMA prediction witnessed the largest deviation (30%) from the real traffic flow, indicating that this method is not suitable for traffic prediction under the stormy weather. By contrast, the improved DBN controlled the deviation within 10%, which is much better than that of the contrastive methods.

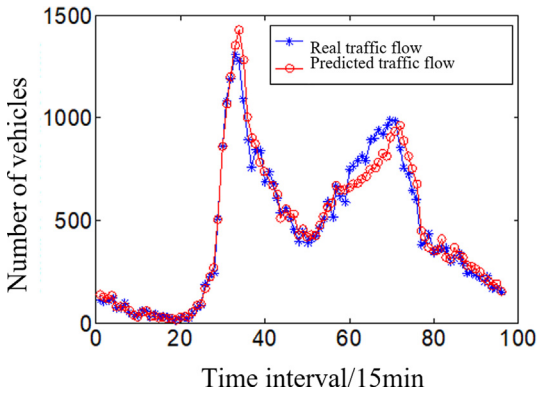

Fig. 4 The prediction effect of the ARIMA.

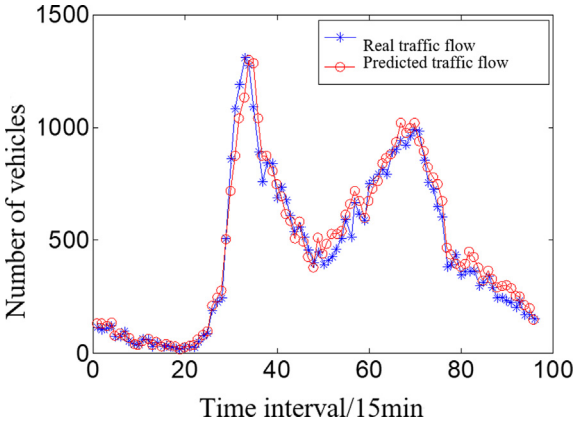

Fig. 5 The prediction effect of the numeral network.

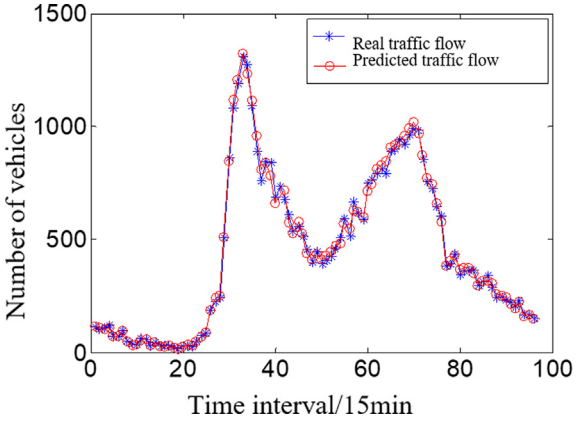

Fig. 6 The prediction effect of the improved DBN.

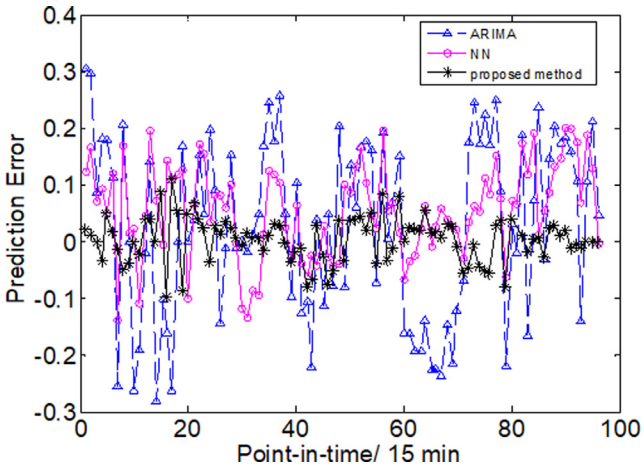

Fig. 7 The prediction errors of the three methods.

To further validate the superiority of the improved DBN, the three methods were applied to predict the traffic flow in a whole month of the rainy season. The predicted results of the three methods are compared in Fig. 9, where the peak time refers to 8:00–19:00 and 16:00–19:00 on each day, and all time refers to the whole day. As shown in Fig. 8, the improved DBN achieved basically the same accuracy in peak time and all time, while the peak time accuracy was 7% and 6% lower than the all-time accuracy of the ARIMA and the neural network, respectively.

Furthermore, the time interval was adjusted to 30, 45 and 60 min in turn, aiming to verify the robustness of the improved DBN. The predicted accuracies of the three methods at different time intervals are presented in Fig. 8. The prediction accuracy and computing time of each method are listed in Table 2. Obviously, the prediction accuracy of every method decreased with the growth in the time interval. However, the improved DBN remained robust in the prediction of long-term traffic flow.

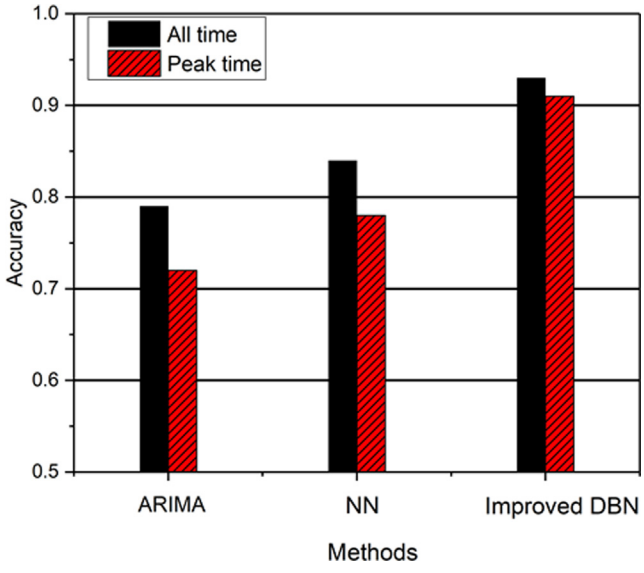

Fig. 8 All time and peak time prediction accuracies of the three methods.

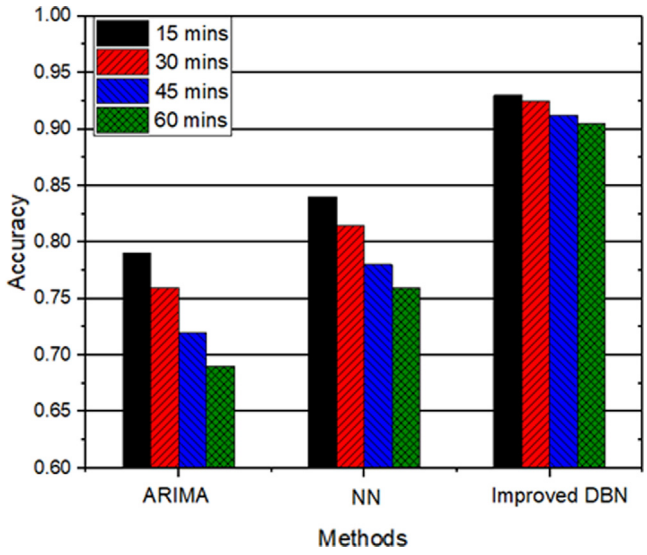

Fig. 9 The prediction accuracies of the three methods at different time intervals.

Table 2 Comparison of prediction accuracy and computing time.

Methods	Prediction accuracy			
	30 min	45 min	60 min	Mean computing time/s
ARIMA	76%	72%	69%	10
Neural network	81.5%	78%	76%	27
Improved DBN	92.5%	91.2%	90.5%	11

5. Conclusion and future work

This paper improves the DBN under a deep framework for traffic prediction under poor weather. In the improved network, the underlying structure is the classic DBN, i.e. a stacking of RBMs, and the top layer is the SVR. The DBN can effectively learn the object features in an unsupervised manner. Meanwhile, the SVR was integrated easily with the DBN and weather factors, and used to map the complex relationship between the elements in the traffic system. The improved DBN was verified through comparative experiments on real traffic data in stormy weather. The experimental results show that the improved DBN outperformed the ARIMA and the neural network in traffic prediction accuracy, and could learn features with very limited prior knowledge. In addition, the improved DBN led the two contrastive methods by 9% in all-time prediction accuracy and 15% in peak time prediction accuracy. The computing time of the improved DBN was slighter longer than the ARIMA, but much shorter than the neural network. These advantages manifest the great application potential of the improved DBN. The future research will further optimize the improved DBN in computing time and application scope. In the future, we will focus on the improvement of traffic prediction under more complexity conditions, such as the combination of weather with other influential factors (road accidents, conditions of the alternative transport, etc.).

Please cite this article in press as: X. Bao et al., An improved deep belief network for traffic prediction considering weather factors, Alexandria Eng. J. (2020), https://doi.org/10.1016/j.aej.2020.09.003

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors gratefully acknowledge supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN201800725), Chongqing primary and secondary school Innovation Talent Project (Grant No. CY200707), General program of Chongqing Natural Science Foundation (Grant No. cstc2019jcyj-msxmX0729), National Science Foundation of China (Grant No. 51909017) and China Scholarship Council (Grant No. 201808505096).

References

[1] Y. Lv, Y. Duan, W. Kang, Z. Li, F.Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2) (2014) 865–873.
[2] N.G. Polson, V.O. Sokolov, Deep learning for short-term traffic flow prediction, Transport. Res. Part C: Emerg. Technol. 79 (2017) 1–17.
[3] W. Liu, Traffic flow prediction based on local mean decomposition and big data analysis, Ingénierie des Systèmes d'Information 24 (5) (2019) 547–552.
[4] W. Xu, L. Liu, Q. Zhang, P. Liu, Location decision-making of equipment manufacturing enterprise under dual channel purchase and sale mode, Complexity (2018) 3797131.
[5] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201.
[6] T. Tuncer, O. Yar, Fuzzy logic-based smart parking system, Ingénierie des Systèmes d'Information 24 (5) (2019) 455–461.
[7] S. Guo, Y. Lin, S. Li, Z. Chen, H. Wan, Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting, IEEE Trans. Intell. Transp. Syst. 20 (10) (2019) 3913–3926.
[8] B.L. Smith, B.M. Williams, R.K. Oswald, Comparison of parametric and nonparametric models for traffic flow forecasting, Transport. Res. Part C: Emerg. Technol. 10 (4) (2002) 303–321.
[9] J. Xie, Y.K. Choi, Hybrid traffic prediction scheme for intelligent transportation systems based on historical and real-time data, Int. J. Distrib. Sens. Netw. 13 (11) (2017).
[10] Y. Sun, M. Zhang, S. Chen, X. Shi, A financial embedded vector model and its applications to time series forecasting, Int. J. Comput., Commun. Control 13 (5) (2018) 881–894.
[11] S. Wang, Y.Z. Hu, Binocular visual positioning under inhomogeneous, transforming and fluctuating media, Traitement du Signal 35 (3–4) (2018) 253–276.
[12] C.K. Moorthy, B.G. Ratcliffe, Short term traffic forecasting using time series methods, Transport. Plann. Technol. 12 (1) (1988) 45–56.
[13] T. Mostefa, B. Tarak, G. Hachemi, An automatic diagnosis method for an open switch fault in unified power quality conditioner based on artificial neural network, Traitement du Signal 35 (1) (2018) 7–21.
[14] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Adv. Comput. Math. 13 (2000) 1.
[15] J. Xu, C. Chen, D. Gao, S. Luo, Q. Qian, Nonlinear dynamic analysis on maglev train system with flexible guideway and double time-delay feedback control, J. Vibroeng. 19 (8) (2017) 6346–6362.
[16] F. Fei, T. Wang, Adaptive fuzzy-neural-network based on RBFNN control for active power filter, Int. J. Mach. Learn. Cybern. 10 (5) (2019) 1139–1150.
[17] M.S. Dougherty, M.R. Cobbett, Short-term inter-urban traffic forecasts using neural networks, Int. J. Forecast. 13 (1) (1997) 21–31.
[18] G.M.A. Soliman, T.H.M. Abou-El-Enien, Terrorism prediction using artificial neural network, Revue d'Intelligence Artificielle 33 (2) (2019) 81–87.
[19] T. Pamuła, Impact of data loss for prediction of traffic flow on an urban road using neural networks, IEEE Trans. Intell. Transp. Syst. 20 (3) (2018) 1000–1009.
[20] T.H. Maze, M. Agarwal, G. Burchett, Whether weather matters to traffic demand, traffic safety, and traffic operations and flow, Transp. Res. Rec. 2006 (1948) 170–176.
[21] J. Asamer, M. Reinthaler, Estimation of road capacity and free flow speed for urban roads under adverse weather conditions, in: Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, 2010, pp. 812–818.
[22] H.C. Manual, HCM2010, Transportation Research Board, National Research Council, Washington, DC, 2010, p. 1207.
[23] T.J. Kwon, L.P. Fu, C.Z. Jiang, Effect of winter weather and road surface conditions on macroscopic traffic parameters, Transport. Res. Rec.: J. Transport. Res. Board 2329 (1) (2013) 54–62.
[24] J. Colyar, Z. Zhang, J. Halkias, Identifying and assessing key weather-related parameters and their impact on traffic operations using simulation, Institute of Transportation Engineers 2003 Annual Meeting and Exhibit (held in conjunction with ITE District 6 Annual Meeting), Institute of Transportation Engineers (ITE), 2003.
[25] H. Alhassan, J. Ben-Edigbe, Extent of highway capacity loss due to rainfall, Int. J. Civil, Environ. Struct. Construct. Arch. Eng. 6 (12) (2012) 1154–1161.
[26] T. Hou, H. Mahmassani, R. Alfelor, J. Kim, M. Saberi, Calibration of traffic flow models under adverse weather and application in mesoscopic network simulation procedures, Transport. Res. Rec.: J. Transpor. Res. Board 2 (2391) (2013) 92–104.
[27] W.H.K. Lam, M. Asce, M.L. Tam, X. Cao, X. Li, Modeling the effects of rainfall intensity on traffic speed, flow, and density relationships for urban roads, J. Transp. Eng. 139 (7) (2013) 758–770.
[28] A. Theofilatos, G. Yannis, Investigation of powered 2-wheeler accident involvement in urban arterials by considering real-time traffic and weather data, Traffic Inj. Prev. 18 (3) (2017) 293–298.
[29] G.W. Hua, Y.D. Xu, Optimal deployment of charging stations and movable charging vehicles for electric vehicles, J. Syst. Manage. Sci. 9 (1) (2018, 2018,) 105–116.
[30] M. Wazid, P. Bagga, A.K. Das, S. Shetty, J.J. Rodrigues, Y.H. Park, AKM-IoV: authenticated key management protocol in fog computing-based internet of vehicles deployment, IEEE Internet Things J. 6 (5) (2019) 8804–8817.
[31] W. Xu, Y. Yin, Functional objectives decision-making of discrete manufacturing system based on integrated ant colony optimization and particle swarm optimization approach, Adv. Prod. Eng. Manage. 13 (4) (2018) 389–404.