Applied Soft Computing 58 (2017) 669–680

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A multi-pattern deep fusion model for short-term bus passenger flow forecasting

Yun Bai, Zhenzhong Sun, Bo Zeng, Jun Deng, Chuan Li*

School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, China

ARTICLE INFO

Article history:
Received 24 October 2016
Received in revised form 5 May 2017
Accepted 5 May 2017
Available online 17 May 2017

Keywords:
Multi-pattern deep fusion
Forecast
Short-term bus passenger flow
Affinity propagation
Deep belief network

ABSTRACT

Short-term passenger flow forecasting is one of the crucial components in transportation systems with data support for transportation planning and management. For forecasting bus passenger flow, this paper proposes a multi-pattern deep fusion (MPDF) approach that is constructed by fusing deep belief networks (DBNs) corresponding to multiple patterns. The dataset of the short-term bus passenger flow is first segmented into different clusters by an affinity propagation algorithm. The passenger flow distribution of these clusters is subsequently analyzed for identifying different patterns. In each pattern, a DBN is developed as a deep representation for the passenger flow. The outputs of the DBNs are finally fused by chronological order rearrangement. Taking a bus line in Guangzhou city of China as an example, the present MPDF approach is modeled. Five approaches, non-parametric and parametric models, are applied to the same case for comparison. The results show that, the proposed model overwhelms all the peer methods in terms of mean absolute percentage error, root-mean-square error, and determination coefficient criteria. In addition, there exists significant difference between the addressed model and the comparison models. It is recommended from the present study that the deep learning technique incorporating the pattern analysis is promising in forecasting the short-term passenger flow.

1. Introduction

Passenger flow forecast is of essential importance to the organization of the transportation system and is one of the most significant basics for decision-making on transportation pattern and operation planning. Therefore, many forecast models and techniques have been proposed and applied to address this issue.

Generally, the short-term transportation forecasting approaches can be classified into two categories: parametric and non-parametric methods [1]. In the parametric model, Box-Jenkins model [2], e.g., autoregressive model (AR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA), is a traditional and effective approach for passenger flow forecasting [3–5]. However, the applications of these models are limited because of a linear assumption among time lagged variables. To track the nonlinear characteristic of the real passenger flow, various non-parametric models have been introduced and improved by researchers [6–11]. Unlike the parametric model, the mainly process of these non-parametric models is to construct a nonlinear relationship between input and output variables without a priori knowledge. Therefore, they are much more flexible and widely-used in feature regression. In view of this point, some hybrid models integrating the parametric and non-parametric methods are designed for improving forecast performances [12–14].

Among these forecasting models, artificial neural networks (ANN) have been regarded as a promising model and proven to be effective in dealing with sophisticated time series [15–18]. However, affected by multiple sources in practice (disturbances increasing), the performances of the ANN are trapped in the feature learning difficulties and the networks calculation complexities. To address this issue, researchers put their efforts from two aspects, i.e., data pre-processing for pattern recognition, and intensive networks for feature learning.

For the data pre-processing, there have two perspectives, i.e., time series decomposition and cluster analysis. The idea of the former is that time series can be viewed as a mixed signal, including various structural and noise components at different scales [19]. Hence various decomposition techniques are put forward to highlight the time-frequency features [1,9,11,17], which are then identified as different transport patterns. Nevertheless, the starting point of the latter is the similarity analysis, i.e., grouping a set of data according to their features (the same or much more similar in distance or logic) [20,21]. The previous studies have proved that extracting remarkable patterns embedded implicitly in dataset through the data pre-processing can enhance the capability of fore-

* Corresponding author.
E-mail address: chuanli@21cn.com (C. Li).

http://dx.doi.org/10.1016/j.asoc.2017.05.011
1568-4946/

casting models, and thus the research of developing and applying data pre-processing techniques is still a hot topic in the passenger flow forecast. Affinity propagation (AP) [22], one of cluster analysis representations, is a novel self-adaptive clustering method, which aims to identify data clusters and each cluster represented by a data point (namely, exemplar). Differing from other clustering methods such as K-means and fuzzy C-means, the AP considering all data points as potential exemplars, can cluster big data into several exemplars in a short time without a given cluster number. Based on this point, the AP algorithm has been successfully applied in gene-expression [23], band selection [24], and pattern recognition [25]. To date, however, there is little literature for its application in transport pattern recognition.

For the intensive networks, Hinton et al. [26] proposed a deep learning (DL) framework. Unlike shallow learning methods (traditional neural network which contains one or zero hidden layer), the essential of the DL is its hierarchical levels (stack networks), that is, the higher levels are determined by the lower levels, where the representation of the low levels may specify several different features of the high levels, this makes the data representation more abstract and nonlinear for the higher levels [26,27]. Realized by "layer-wise" representations [28], the "deeper" feature of the real passenger flow can be captured by the forecasting models sufficiently [29]. As one of the typical structures in the DL, deep belief network (DBN) is adopted successfully in feature extraction [30], classification [31,32], and regression [33]. Recently, the DL technique also has been caught attention in the transport system [34,35].

To fit the sophisticated characteristics of the passenger flow affected by the multi-factor, e.g., transportation managements, holidays, and behavior habits, a multi-pattern deep fusion approach (MPDF), combining the AP algorithm for identifying the passenger flow pattern and the DBN framework for learning sophisticated features of the passenger flow pattern, is proposed in this paper. The MPDF model has three parts: (1) the AP algorithm is operated for cluster analysis, segmenting the passenger flow dataset into several exemplars according, which are then grouped into some special patterns in terms of passenger flow distribution analysis; (2) the DBN framework is applied for digging the features of each special pattern, generating hybrid DBNs for the deep representations of the different patterns; and (3) the outputs of the multi-pattern DBNs are fused as the final forecasts via rearranging in the chronological order. To investigate the forecasting capacity of the proposed model, a real dataset of bus line (Line 10) in Guangzhou city, China, is utilized for modeling and validation. In addition, comparisons with other classical parametric and non-parametric methods are studied.

The remainder of the paper is organized as follows. Section 2 describes the methodologies in detail. The case information, model developments and its evaluation criteria are introduced in Section 3. Section 4 gives the results with relevant discussion. Conclusions are drawn in Section 5.

2. Methodologies

In this section, the systematic methodology of the MPDF approach is described in detail. Following the aforementioned three-step procedure, the constituent techniques are outlined step-by-step in the subsections. The modeling processes of the present model are overviewed in the last subsection.

2.1. Affinity propagation

As introduced in Section 1, the AP algorithm is a promising method for clustering, which has been shown its superiority over the previous algorithms in the literatures [22,36]. Compared to other traditional cluster approaches, the AP is a deterministic clustering method with a stable cluster result. Because the AP algorithm regards each data point as a representative candidate, avoiding the clustering results limited by the choice of the initial represents points [25]. In addition, there is no requirement for the similarity matrix symmetry generated by the data set. Therefore, in this paper, the AP is selected as the pattern identification tool for the passenger flow. According to the literature [22], the AP is described briefly as follows:

For a passenger flow series $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ (N is the length of time series), a similarity of each sample (x_i, x_j) $(i, j \in [1, N])$ is set as

$$s(i,j) = \begin{cases} -\|x_i - x_j\|^2 & i \neq j \\ p & i = j \end{cases}, \tag{1}$$

where p denotes pReferences

Then define responsibility (Eq. (2)) and availability (Eq. (3)) function as follows

$$r(i,j) = s(i,j) - \max_{j'/j}\{a(i,j') + s(i,j')\}, \tag{2}$$

$$a(i,j) = \begin{cases} \min\{0, r(j,j) + \sum_{i' \notin i,j} \max\{0, r(i',j)\}\} & i \neq j \\ \sum_{i'/j} \max\{0, r(i',j)\} & i = j \end{cases}. \tag{3}$$

After a few iterations (m), $r_m(i,j)$ and $a_m(i,j)$ can be updated by

$$\begin{cases} r_m(i,j) = (1-\lambda) r_m(i,j) + \lambda r_{m-1}(i,j) \\ a_m(i,j) = (1-\lambda) a_m(i,j) + \lambda a_{m-1}(i,j) \end{cases}, \tag{4}$$

where $\lambda \in [0,1]$ is a damping factor. This updating step is continued until $(r_m(i,j) + a_m(i,j) > 0)$, and one can get c clusters.

In the AP algorithm, the p is a key parameter, affecting the number of identified exemplars [36]. Due to this point that difficulty in specifying p, the AP algorithm may lead to a suboptimal clustering solution in some cases. Therefore, in this paper, a qualitative analysis, considering travel habits of the passengers, is used to improve the AP cluster results generating some special patterns.

In general, the passenger flow can be categorized as three patterns, i.e., slack hour, normal hour, and rush hour [37]. However, there is no set definition for the time intervals corresponding to three patterns for the different lines. Based on this point, in this paper, the AP algorithm is firstly utilized to segment the flow into several clusters containing different time intervals. According to the distributions in different time intervals, the clusters are then aggregated into three patterns. By this way, the drawback of the AP is conquered, and the time intervals of the three patterns are defined explicitly.

2.2. Deep belief network

After achieving the patterns, the DL technique is introduced to dig the instinct features of each pattern. As mentioned in Introduction section, among the deep learning architecture, the DBN is an effective and typical framework. Therefore, the DBN is constructed in this paper, and the brief description is as follows.

The DBN (Fig. 1(b)) is a stack of simple and unsupervised networks [38]. In this paper, an autoencoder (AE) network (Fig. 1(a)) is taken as the simple component of the DBN framework, named AE-based DBN. From Fig. 1(c), one can see that the AE-based DBN for the passenger flow forecasting is modeled via one AE captures the specific information, information learned by "layer-wise" in many hidden layers, and a final regression layer to accomplish this.

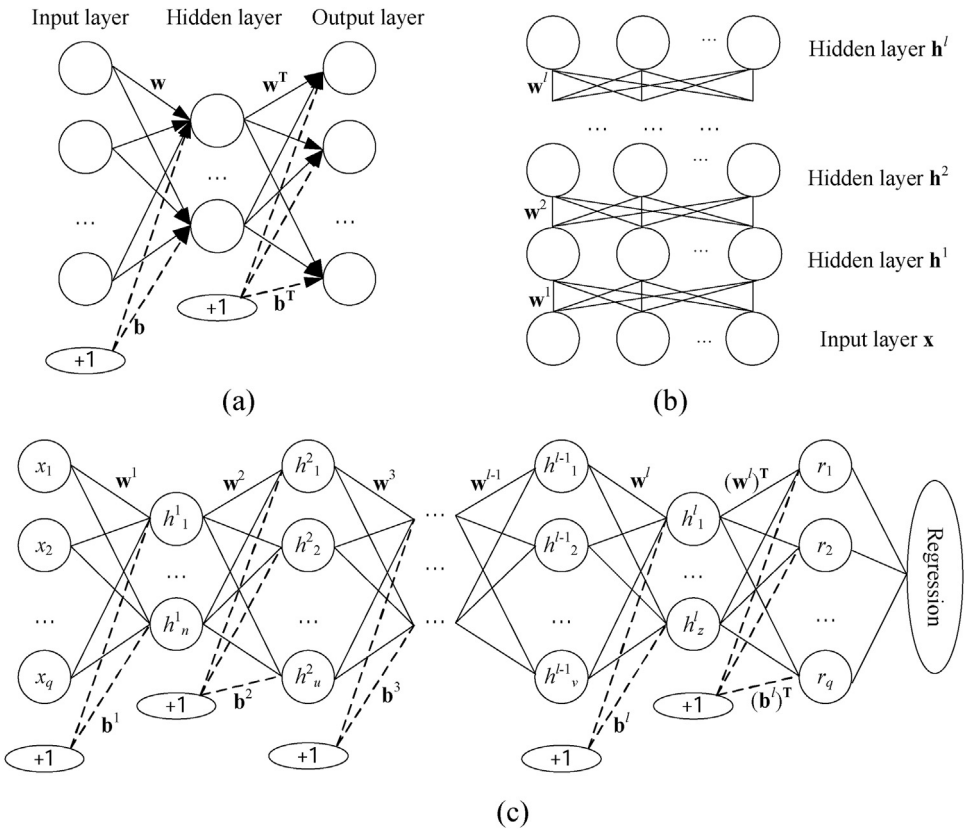

Fig. 1. Structures of the AE-based DBN for regression. $\mathbf{w}^l$ denotes the connection weight matrix between two layers (l-1) and l. (a) AE network, (b) DBN framework, and (c) AE-based DBN model.

As shown in Fig. 1(c), the AE [39] has the same output vector dimensions (reconstruction results $\mathbf{r}$) as the input ones ($\mathbf{h}^{l-1}$ ($\mathbf{h}^0 = \mathbf{x}$), $\mathbf{P}_k$ as inputs for pattern learning) aiming to reconstruct input with a minimum reconstruction error (Eq. (5)).

$$RE(\mathbf{h}^{l-1}, \mathbf{r}) = -\sum_{q=1}^{Q}[h_q^{l-1}\log(r_q) + (1-h_q^{l-1})\log(1-r_q)]. \quad (5)$$

To solve this problem, the AE-based DBN is operated step by step for encoder $f_e(.)$ (Eq. (6)) and decoder $f_d(.)$ (Eq. (7)), until achieving optimal parameter sets ($\mathbf{w}$, $\mathbf{b}$) based on a minimal loss function (Eq. (8)), where $\mathbf{w}$ and $\mathbf{b}$ represent the weights and bias values between neighboring layers, respectively.

$$\mathbf{h}^l = f_e(\mathbf{h}^{l-1}) = Sigm(\mathbf{w}\mathbf{h}^{l-1} + \mathbf{b}), \quad (6)$$

$$\mathbf{r} = f_d(\mathbf{h}^l) = Sigm(\mathbf{w}^T\mathbf{h}^l + \mathbf{b}^T), \quad (7)$$

where Sigm(.) means the sigmoid activation function.

$$L(\mathbf{w}, \mathbf{b}) = \sum RE(\mathbf{h}^{l-1}, \mathbf{r}). \quad (8)$$

According to the literature [40] and the graphical representation shown in Fig. 1, modeling of the AE-based DBN has three steps: (1) From the lower to top layers (layer 1 to layer l), operate generative unsupervised learning layer-wise on the AE. (2) From the top to lower layers (layer l to layer 1), fine-tuning by a supervised learning method, back propagation algorithm, on the whole AE-based DBN to tweak the parameter sets ($\mathbf{w}$, $\mathbf{b}$). (3) From the hidden (top) to output layer, perform regression network using the pre-training parameter sets ($\mathbf{w}$, $\mathbf{b}$). Through the three steps, different DBNs modeled by the different patterns are applied to forecast the passenger flow.

2.3. Overview of the proposed MPDF approach

There are three stages including pattern identification (stage 1), hybrid DBNs (stage 2), and pattern combination (stage 3). Having addressed the constituents separately, the present approach for the passenger flow forecasting can be summarized as follows and illustrated by Fig. 2.

Step 1. Collect passenger flow data $\mathbf{x}$.

Step 2. Cluster $\mathbf{x}$ into c groups using the AP algorithm, i.e., $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_c]$.

Step 3. Identify c clusters' features based on the analysis for the travel habits of the passengers, and construct k patterns, i.e., $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_k]$.

Step 4. Construct deep framework for the feature learning of different patterns, respectively, i.e., DBN_1, DBN_2, …, DBN_k.

Step 5. Perform the hybrid DBNs trained by Step 4, and get $[\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_k]$ using the current intervals. When the data update in a time interval, the DBNs forecasting is realized real-time updates.

Step 6. Fuse the deep representations $[\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_k]$ by the rearrangement with the order of $\mathbf{x}$, and output the final results in the next interval. End.

3. Case study

In this section, a bus line (Line 10) in Guangzhou city of China is investigated as an example for testing the performances of the proposed model. The model developments and their evaluation criteria are also introduced in this section.

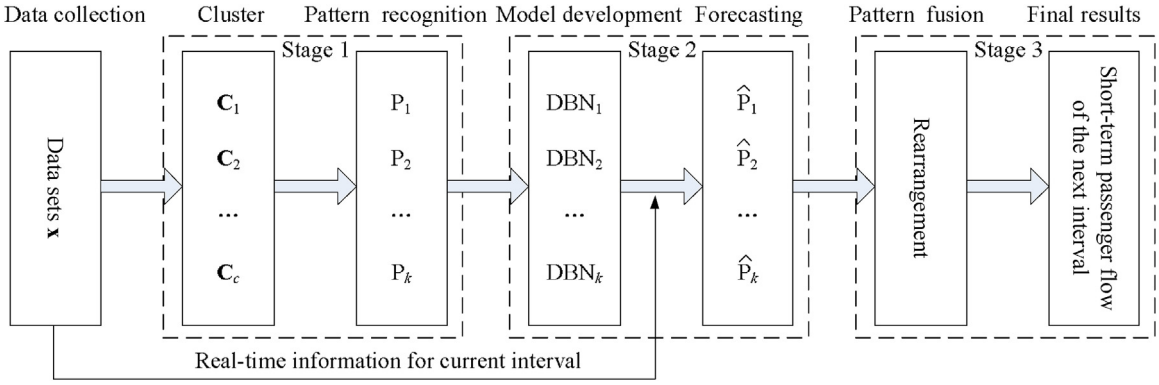

Fig. 2. Framework of the proposed MPDF approach for short-term bus passenger flow forecasting.

3.1. Dataset

The historical bus passenger flow data of Line 10 in Guangzhou city, China, were collected to evaluate the present modeling approach, during the period from Aug. 1 to Dec. 31, 2014 (totally 153 days). Taking the operation time of the Line 10 into consideration, the time period of data collection for each day is from 5:00 to 23:59. There are 2907 observations (153 days × 19 intervals) with a 60-min interval (totally 19 time intervals in each day, i.e., 5–23), e.g., $[5:00-5:59] \in 5$, which is remarked as $\mathbf{x} = [x_\omega(t) \mid t = 5, 6, \ldots, 23, \omega = 1, 2, \ldots, 7]$, where t denotes time interval, and ω denotes Monday ~ Sunday. The details are shown in Fig. 3. For convenience, the time intervals in the horizontal axis are replaced by the number of the time series.

From Fig. 3, one can find two abnormal time intervals during all the collection periods, i.e., (a) system management and (b) public holiday, which may affect the forecasting performances. Since the route adjustment interval has no data, it is not considered in the cluster analysis, but still participated in other processes. Meanwhile, one can also extract a tendency from Fig. 3 without the periods (a) and (b), which is plotted in Fig. 4(a). It is noted that the changes of one week including working days and weekends in Fig. 4(a) are chosen randomly from the dataset shown in Fig. 3 except two abnormal periods. To reveal the public holiday features, the passenger flow on the National Day also plotted in Fig. 4(b).

As shown in Fig. 4(a), some significant regularities can be obtained: (1) on each working day, there have two rush hours in the morning and evening, respectively; (2) on the weekends, the whole passenger flow keeps impartial in some certain, without obvious rush hours; (3) the passenger flow in the weekends is less than that on the working days. In addition, these changes will reappear in the next week (see Fig. 3), thus it can be regarded as periodicity. For the holiday, from Fig. 4(b), one can find that the fluctuations on the National Day is similar to that on the weekends (Fig. 4(a)), and there is little differences between the holiday and the weekends in the amounts of passenger flow. These phenomena follow the travel habits completely. Therefore, in the following section, the National Day interval can be viewed as weekends in a normal week, reducing the influence on the modeling.

3.2. Model development

In this subsection, the ingredients of the proposed model, i.e., the AP and the DBN, are developed for the real passenger flow in detail.

For the AP, the similarity matrix $\mathbf{S} = [s(i, j) | i, j \in [1, N]]$ must be computed at first, which is determined by the pairwise (x_i, x_j). Therefore, considering the analysis in subsection 3.1, the two cluster pair samples are chosen from 1/9/2014 to 30/10/2014, including two normal periods with one holiday. Then the key parameter p is set as the median of the similarity matrix $\mathbf{S}$ (default value), and the λ is designed as 0.5 according to the experimental results in literature [36].

Table 1
Model developments of each stage.

Model stage	Experimental design
AP for similarity clustering	λ=0.5; p = median($\mathbf{S}$); max iteration = 500
DBN for deep feature learning	Input nodes = [one week-ahead, two weeks-ahead, three weeks-ahead, four weeks-ahead]; Hidden layers = [1–3]; Hidden nodes = [10,20,30,40,50]; Output nodes = [next week]. Other parameters: dropout 0.5, learning rate 1, size of batch training 100, and training iteration 30.

For the DBN, the numbers of the input nodes, hidden nodes and layers, and output nodes are main variables. In this paper, an experimental method is employed to determine these parameters. According to the analysis in subsection 3.1, the passenger flow has a natural period (entire one week). To reveal its features in one week, a multi-input and multi-output structure is developed in this paper, i.e., the records of one or more week(s) in advance as inputs, and the forecasts of next week as outputs. Therefore, the experimental designing is as follows: four levels of the input nodes ranging from 1 to 4 week(s) will be validated, three levels of the hidden layers sizes from 1 to 3 will be considered (for the comparison, only one hidden layer is designed), the number of nodes in each hidden layer is set to be the same ranging from 10 to 50 (interval 10), and the output nodes is set to one week.

In the following study, the original time series shown in Fig. 3 is normalized as [0,1], which will be used for the training (90%) and testing (10%). The details of the model developments are listed in Table 1.

3.3. Performance criteria

Three criteria, mean absolute percentage error (MAPE), root-mean-square error (RMSE), and determination coefficient (R^2), are employed to assess the forecasting performances. The definitions of three criteria are listed as follows:

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{x_i - \hat{x}_i}{x_i}\right|, \quad \mathrm{RMSE} = \sqrt{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2/N},$$

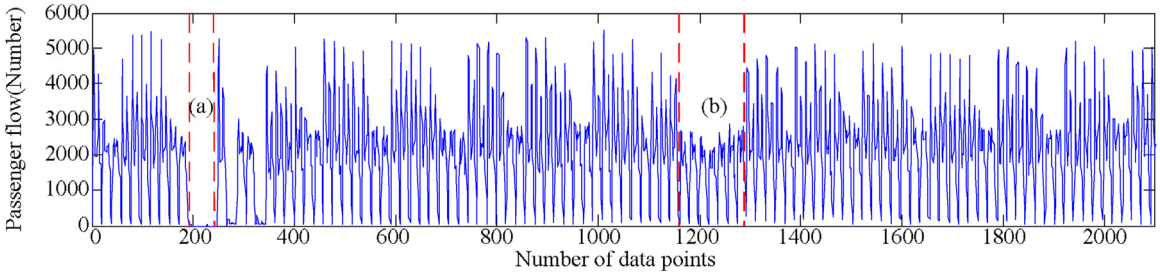

Fig. 3. Bus passenger flow of the Line 10 from 1/8/2014 to 31/12/2014 in Guangzhou of China, where (a) represents 11/8/2014–13/8/2014 (route adjustment) and (b) stands for 1/10/2014–7/10/2014 (National Day).

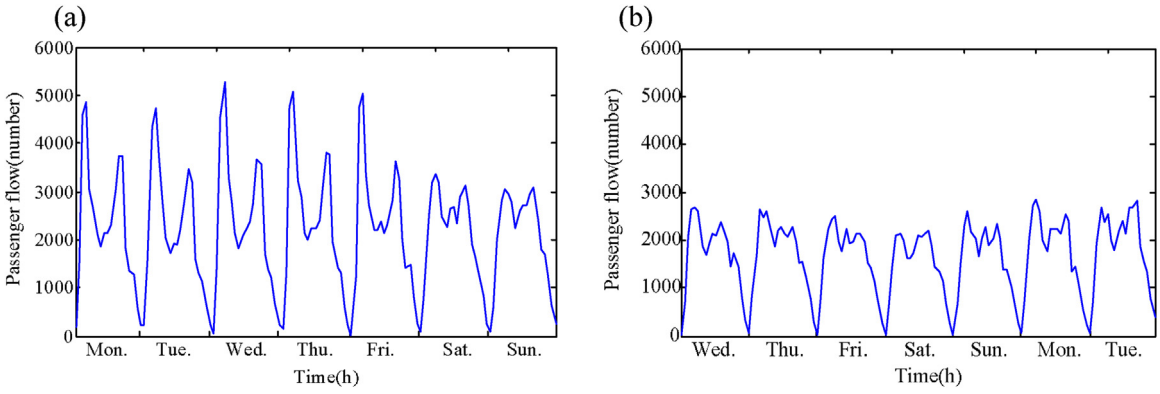

Fig. 4. Passenger flow distribution of the Line 10: (a) one week, and (b) National Day.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{N}(x_i - \bar{x})^2}, \tag{9}$$

where x_i and $\hat{x}_i$ represent the i-th recoded and forecasted values, respectively, and $\bar{x}$ refer to the mean values of the records.

4. Results and discussion

In this section, the addressed model is tested using the real data of the bus Line 10 in Guangzhou city, China. To show the superiority of the proposed model, the forecasting performances are also compared with other methods.

4.1. Pattern identification results

At the first stage, the time series data of passenger flow during 1/9/2014 – 30/10/2014 are clustered into nine components ($\mathbf{C}_1$, $\mathbf{C}_2$, …, $\mathbf{C}_9$) by the AP method, as shown in Fig. 5. It is noted that the clusters are named in the order they are clustered. The performances of the AP are as follows: number of iterations = 150, elapsed time = 4.893 s, and fitness (net similarity) = −3.73535.

According to the clustering results shown in Fig. 5, Table 2 gives a summary for the clusters' distribution.

From Table 2, one can find that the AP algorithm can cluster the pair samples successfully both for the week ω and the time interval t. However, the clustering results may be a suboptimal solution, that is, some clusters have the same time interval, e.g., $\mathbf{C}_1$, $\mathbf{C}_3$, $\mathbf{C}_4$. This maybe resulted from the similarity computed by Eq. (1). On the one hand, $\mathbf{S}$ is evaluated by the Euler distance that may cover the logical features. On the other hand, it is hard to determine the p value [22], which is just an empirical value without any theoretical basis. Therefore, to improve the suboptimal solution, based on the cluster results and the analysis in subsection 3.1, the clusters $\mathbf{C}_1$, $\mathbf{C}_2$, …, $\mathbf{C}_9$ can be aggregated into three patterns for each week, i.e.,

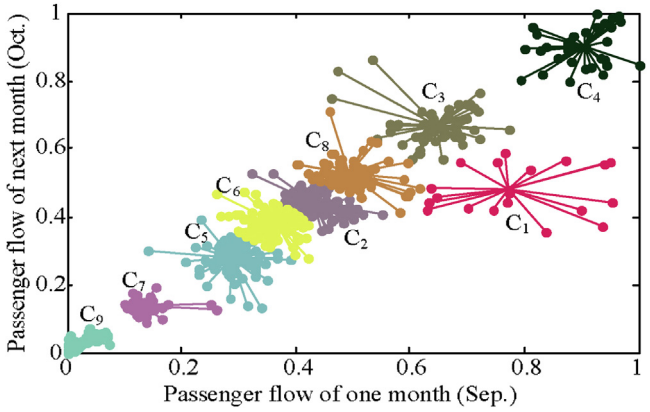

Fig. 5. Clustering results using the AP algorithm.

slack hours $\mathbf{P}_1 = [\mathbf{C}_7, \mathbf{C}_9]$, normal hours $\mathbf{P}_2 = [\mathbf{C}_2, \mathbf{C}_5, \mathbf{C}_6, \mathbf{C}_8]$, and rush hours $\mathbf{P}_3 = [\mathbf{C}_1, \mathbf{C}_3, \mathbf{C}_4]$. Fig. 6 shows the three patterns respectively.

From Fig. 6, one can see that the pattern identification is appropriate, except for two significant outliers in $\mathbf{P}_2$ (the route adjustment time interval is out of consideration). From the statistical point of view, the mean values of three patterns are 540, 2091, and 3765, respectively. This shows that the distinction degree is acceptable. In addition, the results shown in Fig. 7 prove the validity of the pattern identification strategy. Compared with Fig. 6(c), an obvious outlier appears plotted in shown Fig. 7(b) which is corresponding to the public holiday (National Day). Moreover, the period of the National Day has been transferred into Fig. 7(a), generating a new pattern shown in Fig. 6(b). There has no notable difference in this process. Therefore, it is reasonable for regarding the National Day period as a normal weekend.

To sum up, based on the qualitative analysis for the travel habits of the bus passengers (see subsection 3.1), the drawback of the AP

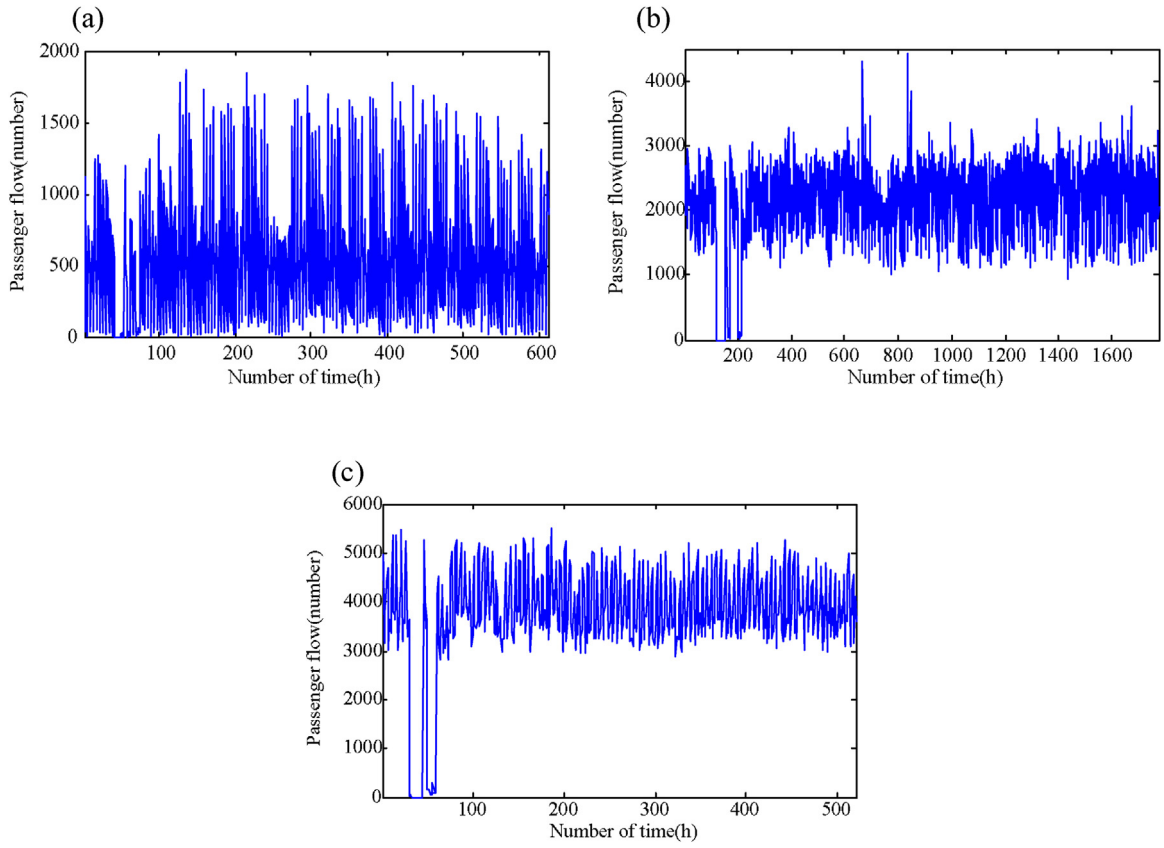

Fig. 6. Three patterns distributions. (a) slack hours $\mathbf{P}_1$, (b) normal hours $\mathbf{P}_2$, and (c) rush hours $\mathbf{P}_3$.

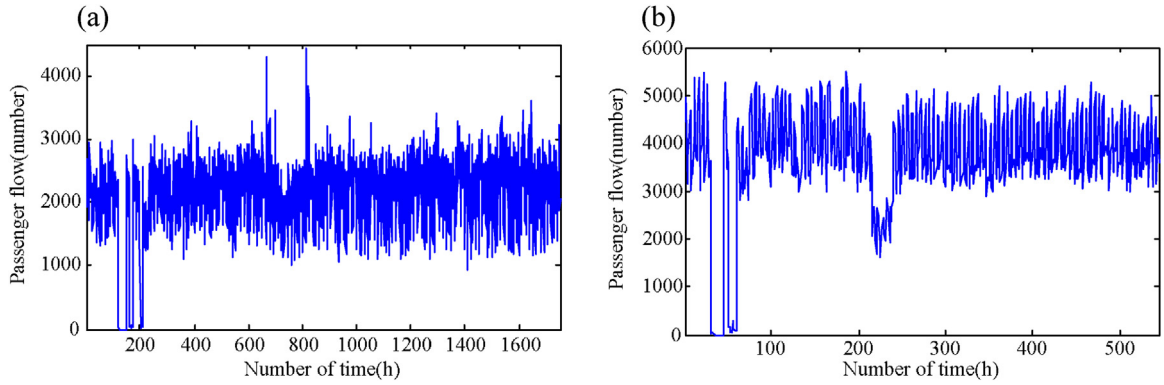

Fig. 7. Comparison results obtained by the single application with AP algorithm. (a) normal hours $\mathbf{P}_2$ and (b) rush hours $\mathbf{P}_3$.

algorithm has been eliminated. The results of the pattern identification are much more suitable as the inputs for the DBN regression in the next step.

4.2. Deep feature learning results

After identifying patterns, the hybrid DBNs for feature learning is designed according to Table 1. Specifically, the input $\mathbf{x}$ and the outputs of three patterns should be set as shown in Table 3, and the modeling performances (MAPE and RMSE) of the DBNs are summarized in Table 4. It is noted that there have four categories of the input-output structures modeled in Table 1. For the three patterns, the corresponding input-output structures are set as $\mathbf{P}_1$ [one week-ahead (28 nodes)| next week (28 nodes)], [two weeks-ahead (56 nodes)| next week (28 nodes)], [three weeks-ahead (84 nodes)| next week (28 nodes)], [four weeks-ahead (112 nodes)| next week (28 nodes)], $\mathbf{P}_2$ [one week-ahead (80 nodes)| next week (80 nodes)], [two weeks-ahead (160 nodes)| next week (80 nodes)], [three weeks-ahead (240 nodes)| next week (80 nodes)], [four weeks-ahead (320 nodes)| next week (80 nodes)], and $\mathbf{P}_3$ [one week-ahead (25 nodes)| next week (25 nodes)], [two weeks-ahead (50 nodes)| next week (25 nodes)], [three weeks-ahead (75 nodes)| next week (25 nodes)], [four weeks-ahead (100 nodes)| next week (25 nodes)].

As shown in Table 4, one can summarize the following points: (1) the results of the deep structure are better than that of the "shallow" framework (one hidden layer) in terms of the MAPE, the RMSE, and the R, (2) generally, the deeper the hidden layer is, the better the performance will be for feature learning, and (3) the deep structures of the three patterns exhibit the diversity in the deep representa-

Table 2
Distribution of different clusters.

Cluster	Distribution
$\mathbf{C}_1$	Mon. t = 7, 8, Tue. t = 7, 8, 17, 18; Wed. t = 7, 8, 9, 17, 18; Thu. t = 7, 8, 17, 18; Fri. t = 7, 8, 17, 18
$\mathbf{C}_2$	Mon. t = 10, 11, 13–17; Tue. t = 10, 11, 13–15, 19; Wed. t = 10, 14, 15; Thu. t = 11, 14–16; Fri. t = 10, 11, 14, 15, 19; Sat. t = 7-18; Sun. t = 8, 10–18
$\mathbf{C}_3$	Mon. t = 9, 17, 18; Tue. t = 9, 17, 18; Wed. t = 9, 17, 18; Thu. t = 9, 17, 18; Fri. t = 9, 17, 18
$\mathbf{C}_4$	Mon. t = 7, 8; Tue. t = 7, 8, 17; Wed. t = 7, 8; Thu. t = 7, 8; Fri. t = 7, 8
$\mathbf{C}_5$	Mon. t = 6, 19–21; Tue. t = 6, 12, 20, 21; Wed. t = 6, 20, 21; Thu. t = 6, 19–21; Fri. t = 6, 20, 21; Sat. t = 7, 19–21; Sun. t = 7, 19–21
$\mathbf{C}_6$	Mon. t = 11–14, 18, 19; Tue. t = 11–14, 19; Wed. t = 11–14, 19, 20; Thu. t = 11–14, 19; Fri. t = 11–14, 19; Sat. t = 11-15, 19; Sun. t = 7, 11–13, 18
$\mathbf{C}_7$	Mon. t = 6, 22; Tue. t = 6, 22; Wed. t = 6, 22; Thu. t = 6, 22; Fri. t = 6, 22; Sat. t = 6, 22; Sun. t = 6, 22
$\mathbf{C}_8$	Mon. t = 9, 10, 16; Tue. t = 9, 10, 15, 16; Wed. t = 10, 16; Thu. t = 9, 10, 16; Fri. t = 9, 10, 15, 16; Sat. t = 8–10, 13–18; Sun. t = 8–10, 14, 16, 17
$\mathbf{C}_9$	Mon. t = 5, 23; Tue. t = 5, 23; Wed. t = 5, 23; Thu. t = 5, 23; Fri. t = 5, 23; Sat. t = 5, 23; Sun. t = 5, 23

tions, and the hybrid DBNs with the smallest MAPE, RMSE, and R are trained as follows: $\mathbf{P}_1$ l = 2, 50 nodes in each hidden layer, $\mathbf{P}_2$ l = 3, 40 nodes in each hidden layer, and $\mathbf{P}_3$ l = 3, 30 nodes in each hidden layer.

Additionally, according to the literature [41,42], the Wilcoxon signed ranks test is applied for statistical comparisons between the control model (with the smallest MAPE, RMSE, and R) and the others. Through carrying out the Wilcoxon test, one can choose the simplest model's structure that is not significantly different from the control model, so as to obtain better generalization ability. Table 5 gives the results of the Wilcoxon test containing the control models of different patterns and the corresponding models without significant differences at the confidence level of 5%. Synthesizes each component of the deep structures (input nodes, hidden layers and hidden nodes in each hidden layer), the control models of $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ are selected in this paper.

Using the selected models, the deep learning results for the testing data, translated into the real data by the inverse normalization, are plotted in Fig. 8.

As shown in Fig. 8(a), the DBN_1 achieves successful feature learning results for the $\mathbf{P}_1$, illustrating that the fluctuation of the slack hours are tracked in a deep fashion. In addition, the criteria are computed as MAPE = 19.628%, RMSE = 82.079, R^2 = 0.934, which state that the higher MAPE occurs in the slack hours, i.e., late evening (after 22:00) and early morning (before 7:00). That is due to the fact that the passenger flow in this period tends to be rather small but fluctuates greatly (the changes in large orders of magnitude). This is hard to capture all the features in these intervals, especially for the transition periods.

From Fig. 8(b), the DBN_2 capture the fluctuations of the $\mathbf{P}_2$ in general, except for few peak values, and the criteria are MAPE = 10.208%, RMSE = 161.071, and R^2 = 0.711. The lower R reveals that the deep structure is affected by the complex and frequent changes in a certain extent. However, during these periods, i.e., normal hours (10:00-16:00, 19:00-21:00) on the workday and (7:00-21:00) on the weekend, the MAPE is not influenced obviously. It is maybe due to the relatively stable travel habit, because passengers have relatively less departure choice to work in the morning, to rest in the evening, and to enjoy their weekend (travel mode diversification).

The DBN_3 plotted in Fig. 8(c), also follows the trends of the $\mathbf{P}_3$ in a certain extent, and the criteria are MAPE = 4.605%, RMSE = 134.577, and R^2 = 0.834. However, one can find a significant phenomenon, i.e., the fitting capacity of the a.m. rush hour interval (t = 7-9) is better than that of the p.m. rush hour interval (t = 17-18). Compared to the time before work (fixed working time), the time is much more flexible, which is maybe lead to the poor performances in the p.m. rush hour.

4.3. Pattern fusion results

After getting the multi-pattern results of the passenger flow using the three DBNs respectively, the final results are fused via the rearrangement. The pattern combination is plotted in Fig. 9(a), and the scatter plot is generated in Fig. 9(b). Note that the dash line represents the records, and the full line means the forecasts.

From Fig. 9(a), one can see that the forecasts follow the trends and fluctuations of the testing records successfully, except for the rush hour in the weekends (3 points). As explained before, unlike the working day, the passenger flow, affected by random factors severely, exhibits flexibility and variability. From Fig. 9(b), one can find that the correlations between the recorded and the forecasted data are concentrated nearby the ideal fit. Furthermore, the quantitative evaluation results, MAPE = 10.743%, RMSE = 157.312, and R^2 = 0.958, respectively, also exhibit a high accuracy. Compared with the criteria of three patterns, the error accumulation effect is not significant.

According to the quantitative and qualitative analysis above, the proposed model MPDF precisely forecasts the passenger flow in the entire testing time with acceptable errors. The performance can be attributed to the multi-pattern feature observation and fitting, i.e., multi-pattern extraction and deep feature representation.

4.4. Comparisons

To evaluate forecasting performance of the proposed model, four models are applied for the comparisons using the same dataset, i.e., the non-parametric model (the DBN, the AE, the AP-based AE, and feed forward neural network (FFNN)) and the parametric model (the ARIMA). These model structures are listed in Table 6. Note that the structures are determined by the experiments (except for the ARIMA which is determined by [4]), the same as the training processes of the proposed model, and other parameters are the same as the proposed model. The forecasting results with the three models are displayed in Fig. 10.

Table 3
Input and output nodes of different patterns.

Pattern	Input nodes (data on the previous day)	Output nodes (data on the next day)	Number of nodes
$\mathbf{P}_1$	$x_\omega(t)$, t = 5, 6, 22, 23 $\omega \in [1, 7]$	$x_\omega(t)$, t = 5, 6, 22, 23 $\omega \in [1, 7]$	28
$\mathbf{P}_2$	$x_\omega(t)$, t = 10-16, 19–21 $\omega \in [1, 5]x_\omega(t)$, t = 7-21 $\omega \in [6, 7]$	$x_\omega(t)$, t = 10-16, 19–21 $\omega \in [1, 5]x_\omega(t)$, t = 7-21 $\omega \in [6, 7]$	80
$\mathbf{P}_3$	$x_\omega(t)$, t = 7-9, 17, 18 $\omega \in [1, 5]$	$x_\omega(t)$, t = 7-9, 17, 18 $\omega \in [1, 5]$	25

Table 4
Results of different patterns with different DBNs.

Pattern	Input-output	Hidden	MAPE	RMSE	R^2	Hidden	MAPE	RMSE	R^2	Hidden	MAPE	RMSE	R^2
$\mathbf{P}_1$	28–28	10	51.356	135.331	0.807	20	26.549	95.779	0.918	30	21.046	94.713	0.910
	56–28		36.451	139.987	0.834		28.081	121.441	0.873		18.365	137.491	0.851
	84–28		16.051	155.089	0.825		17.092	145.643	0.831		15.500	111.868	0.893
	112–28		15.470	141.868	0.817		15.294	111.203	0.897		18.115	150.736	0.815
	28–28	40	25.687	99.372	0.913	50	20.325	105.125	0.904	10×10	26.804	110.437	0.889
	56–28		15.824	160.145	0.795		20.740	159.149	0.789		23.628	130.302	0.860
	84–28		14.613	116.298	0.885		15.124	124.213	0.867		15.904	141.329	0.839
	112–28		13.609	111.934	0.879		14.650	155.313	0.820		15.736	149.774	0.826
	28–28	20×20	19.488	104.935	0.899	30×30	17.686	112.646	0.884	40×40	16.618	70.290	0.946
	56–28		20.109	116.792	0.869		19.219	111.627	0.880		13.751	57.462	0.954
	84–28		26.732	100.087	0.884		11.442	95.718	0.921		12.169	66.528	0.955
	112–28		23.512	124.143	0.851		17.406	123.863	0.855		12.043	90.434	0.920
	28–28	50×50	11.897	71.820	0.949	$10 \times 10 \times 10$	25.763	81.767	0.937	$20 \times 20 \times 20$	13.191	75.289	0.945
	56–28		**10.875**	**66.136**	**0.965**		19.716	107.044	0.906		16.931	70.732	0.947
	84–28		11.987	81.582	0.941		17.880	137.997	0.844		16.652	74.739	0.938
	112–28		11.452	90.926	0.925		16.857	137.544	0.835		17.777	66.478	0.949
	28–28	$30 \times 30 \times 30$	15.496	81.808	0.935	$40 \times 40 \times 40$	14.886	62.685	0.953	$50 \times 50 \times 50$	17.546	100.683	0.901
	56–28		13.719	76.603	0.943		16.211	75.968	0.937		13.542	61.807	0.956
	84–28		13.015	71.444	0.952		14.229	63.170	0.951		13.427	64.119	0.954
	112–28		13.637	54.328	0.957		13.921	76.353	0.938		15.870	76.220	0.933
$\mathbf{P}_2$	80–80	10	15.112	224.927	0.399	20	15.380	213.987	0.423	30	13.653	197.094	0.546
	160–80		17.064	207.307	0.363		15.687	230.885	0.380		15.286	229.606	0.414
	240–80		16.359	212.692	0.374		16.143	220.709	0.365		15.088	224.345	0.416
	320–80		16.746	259.642	0.263		15.883	218.453	0.385		15.523	226.999	0.403
	80–80	40	12.752	193.561	0.606	50	11.800	198.388	0.598	10×10	16.589	217.470	0.329
	160–80		12.046	206.225	0.601		12.174	214.217	0.573		16.097	200.964	0.418
	240–80		13.587	209.461	0.520		12.649	211.480	0.573		16.042	207.418	0.412
	320–80		16.085	224.494	0.363		12.777	204.123	0.575		16.607	254.093	0.281
	80–80	20×20	16.159	222.697	0.374	30×30	15.130	206.665	0.466	40×40	12.301	217.269	0.560
	160–80		15.207	220.067	0.435		12.748	194.185	0.611		10.487	177.349	0.717
	240–80		13.014	199.362	0.579		12.409	187.177	0.625		10.768	204.801	0.655
	320–80		14.784	224.246	0.440		11.864	215.512	0.593		12.376	210.813	0.587
	80–80	50×50	11.321	182.536	0.679	$10 \times 10 \times 10$	16.076	209.677	0.395	$20 \times 20 \times 20$	14.053	182.029	0.537
	160–80		11.275	182.813	0.670		15.974	215.387	0.404		12.984	202.765	0.570
	240–80		11.799	188.659	0.636		16.544	225.892	0.319		13.442	225.049	0.513
	320–80		12.415	180.652	0.634		16.707	254.610	0.276		13.874	203.938	0.521
	80–80	$30 \times 30 \times 30$	12.015	193.884	0.620	$40 \times 40 \times 40$	11.354	174.537	0.686	$50 \times 50 \times 50$	10.953	179.234	0.678
	160–80		12.722	214.998	0.571		**9.906**	**166.413**	**0.772**		11.801	189.211	0.645
	240–80		11.927	202.397	0.621		11.856	169.337	0.678		10.866	172.787	0.716
	320–80		11.503	181.397	0.666		11.322	198.027	0.666		10.245	193.560	0.709
$\mathbf{P}_3$	25–25	10	9.098	236.072	0.342	20	7.070	185.972	0.606	30	4.760	156.490	0.787
	50–25		5.847	173.223	0.703		8.062	255.121	0.384		6.785	192.284	0.624
	75–25		9.501	187.419	0.383		8.313	199.430	0.500		7.268	147.163	0.613
	100–25		7.605	233.905	0.491		4.268	107.811	0.853		4.880	168.855	0.756
	25–25	40	4.113	117.914	0.857	50	6.674	188.651	0.646	10×10	9.371	250.096	0.381
	50–25		5.524	164.008	0.743		5.733	200.869	0.673		9.958	192.343	0.322
	75–25		4.819	134.876	0.800		5.353	163.664	0.743		6.625	196.167	0.626
	100–25		4.203	130.026	0.836		5.013	114.064	0.816		4.753	132.703	0.807
	25–25	20×20	7.617	200.911	0.546	30×30	4.186	152.409	0.814	40×40	6.952	172.210	0.621
	50–25		4.741	130.205	0.812		4.434	133.925	0.817		4.478	111.727	0.847
	75–25		4.951	93.038	0.834		5.833	149.989	0.726		4.143	111.375	0.862
	100–25		4.332	115.340	0.844		4.877	143.108	0.787		4.675	92.776	0.848
	25–25	50×50	4.979	132.753	0.798	$10 \times 10 \times 10$	8.794	247.469	0.359	$20 \times 20 \times 20$	8.460	246.778	0.359
	50–25		5.226	177.210	0.731		4.732	156.418	0.782		4.821	140.53	0.800
	75–25		5.135	140.478	0.777		5.221	135.594	0.776		5.004	113.597	0.812
	100–25		8.662	216.995	0.532		5.178	175.864	0.735		6.384	194.659	0.639
	25–25	$30 \times 30 \times 30$	**4.031**	**119.568**	**0.868**	$40 \times 40 \times 40$	4.764	135.976	0.819	$50 \times 50 \times 50$	5.034	142.713	0.791
	50–25		4.736	145.191	0.794		4.888	165.700	0.762		4.901	120.185	0.812
	75–25		5.859	193.308	0.668		6.495	90.422	0.741		4.435	122.551	0.831
	100–25		4.471	100.597	0.850		9.542	248.370	0.384		4.714	130.805	0.813

For the denotation A-B, A and B represent the number of the input and output nodes respectively.

Table 5
Wilcoxon test results of different patterns.

Pattern	Control model	Model not significant difference
$\mathbf{P}_1$	56–50–50-28	28–10–10–10-28, 28-20-20-20-28, 28-30-30-30-28, 28-40-40-40-28, 28-50-50-50-28, 56–10–10–10-28, 56-20-20-20-28, 56-30-30-30-28, 56-40-40-40-28, 84–10–10-28, 84–10–10–10-28, 84-20-20-20-28, 84-30-30-28, 84-30-30-30-28, 84-40-40-28, 84-40-40-40-28, 84-50-50-28, 84-50-50-50-28, 112–10–10-28, 112–10–10–10-28, 112-20-20-28, 112-20-20-20-28, 112-30-30-28, 112-30-30-30-28, 112-40-40-40-28, 112-50-50-28, 112-50-50-50-28
$\mathbf{P}_2$	160–40–40–40-80	80–50–50–50-80, 160–50–50–50-80, 240-10-10-10-80, 240-20-20-20-80, 240-40-40-40-80, 240–50–50-80, 240–50–50–50-80, 320-10-10-10-80, 320-20-20-80, 320-20-20-20-80, 320-30-30-30-80, 320-40-40-80, 320-40-40-40-80, 320–50–50-80, 320–50–50–50-80
$\mathbf{P}_3$	25-30-30-30-25	25-50-50-50-25, 50–10–10–10-25, 50-30-30-30-25, 50-40-40-40-25, 50-50-50-50-25, 75–10–10–10-25, 75-20-20-20-25, 75-30-30-30-25, 75-40-40-25, 75-40-40-40-25, 75-50-50-25, 75-50-50-50-25, 100–10–10–10-25, 100-20-20-25, 100-20-20-20-25, 100-30-30-30-25, 100-40-40-25, 100-40-40-40-25, 100-50-50-25, 100-50-50-50-25

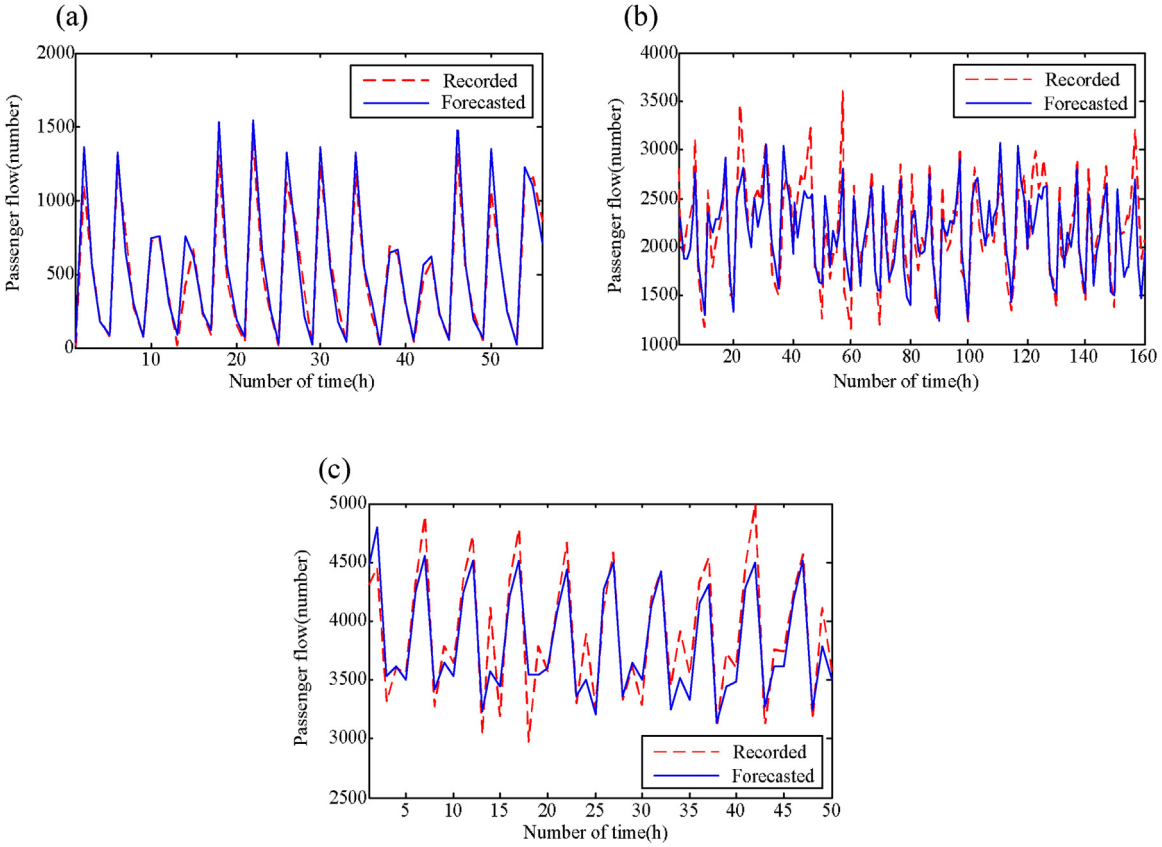

Fig. 8. Deep learning results of three patterns with different DBNs. (a) DBN_1 results of $\mathbf{P}_1$ with 56–50–50–28; (b) DBN_2 results of $\mathbf{P}_2$ with 160–40–40–40–80; and (c) DBN_3 results of $\mathbf{P}_3$ with 25–30–30–30–25.

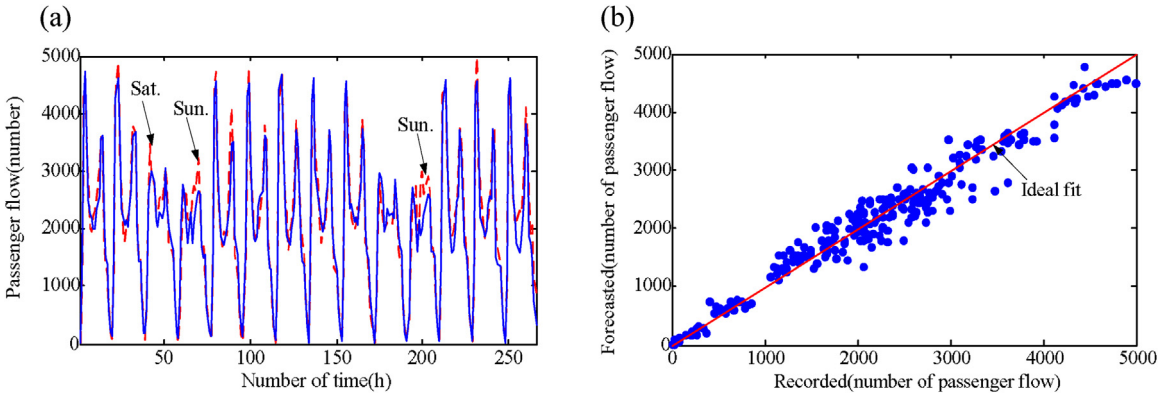

Fig. 9. Pattern fusion results: (a) plots the comparison with the records and the forecasts, and (b) shows the scatter plots.

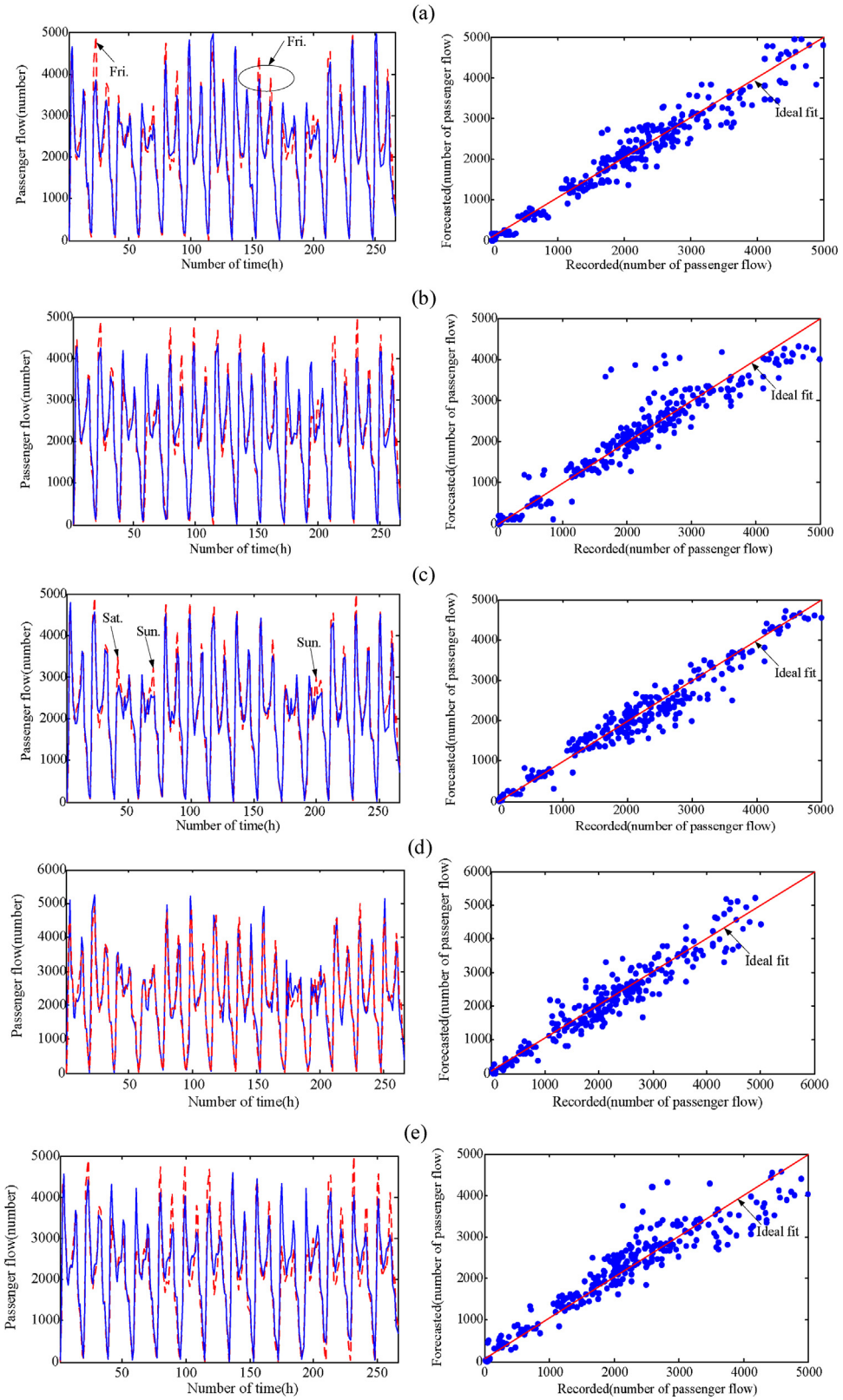

Fig. 10. Forecasting results with different models. (a) DBN with 266-50-50-133, (b) AE with 133–30–133, (c) AP-based AE with 112–40–28, 80–50–80, 25–40–25 respectively, (d) FFNN with 133–25–133, and (e) ARIMA $(2,1,2)(2,1,2)_{19}$.

As shown in Fig. 10, the results of the three models are all follow the whole time-varying changes of the testing dataset. However, there have some unsatisfactory forecasts in these models, which are resulted from the different capacities for feature learning. The DBN with single application has poor fitting abilities for the rush hour on Friday, the AE for the rush hour on working day (lower than records) and weekends (higher than records), the AP-based AE for the rush hour on the weekends, the FFNN for the rush hour on work-

Table 6
Structure of different models for comparison.

Comparison model	Model structure
DBN	Input nodes = 266, hidden layer = 2, hidden nodes in each layer = 50, output nodes = 133
AE	Input nodes = 133, hidden layer = 1, hidden nodes = 30, output nodes = 133
AP-based AE	Pattern is the same as the proposed model, $\mathbf{P}_1$: Input nodes = 112, hidden layer = 1, hidden nodes = 40, output nodes = 28, $\mathbf{P}_2$: Input nodes = 80, hidden layer = 1, hidden nodes = 50, output nodes = 80, $\mathbf{P}_3$: Input nodes = 25, hidden layer = 1, hidden nodes = 40, output nodes = 25
FFNN	Input nodes = 133, hidden layer = 1, hidden nodes = 25, output nodes = 133
ARIMA	Autoregressive degree (AR) = 2, differencing degree = 1, moving average degree (MA) = 2, seasonality = 19, seasonality autoregressive degree (SAR) = 2, seasonality differencing degree = 1, and seasonality moving average degree (SMA) = 2

ing day and weekends (both higher than records), and the ARIMA for the rush and slack hours (especially, the errors increase with the time). The scatter plots shown in Fig. 10 reveal that the correlations between the records and the forecasts of all the models are divergent in a slight certain (the R index has little difference). Comprehensively, the AP-based AE has a better performance among these comparison models, attributing to the pattern identification which can improve the forecasting capacity.

Compared with Figs. 9(a)–(b), the proposed model MPDF has the best performances among these models, especially from the perspective of the scatter plots. Therefore, according to the qualitative analysis, the forecasting performance rank is the MPDF (best), the AP-based AE, the DBN, the AE, the FFNN, and the ARIMA (worst).

Besides, quantitative evaluations are also investigated, and the results are summarized in Table 7. The computation time of each model is also recorded in Table 7 with the computation environment Intel Core i5–2450 M CPU @2.50 GHz 2.50 GHz, and Memory 4.00GB.

As shown in Table 7, the ARIMA model has the highest MAPE and RMSE, illustrating that the parametric model has difficulty for tracking the trends of the nonlinear time series sufficiently. In addition, the models considering the pattern feature (the MPDF and the AP-based AE) has lower MAPE, RMSE, and higher R than that of the model without considerations (the DBN, the AE, and the FFNN), demonstrating the pattern identification's superiority. In addition, the models under the deep structures (the MPDF and the DBN) have deeper feature understanding abilities than that with the "shallow" networks (the AE and the FFNN). Therefore, the proposed model, integrating the pattern recognition and deep feature learning, is beneficial to explore sophisticated features, and displays better regression capacity for the short-term passenger flow. According to its better performance, the loss of the computation time with the complexity structure is acceptable.

Moreover, the results of the Wilcoxon signed ranks test at 5% level are listed in Table 8. From Table 8, one can find the values of the significance (asymptotic) between the proposed model and the comparison models are all less than 0.05, demonstrating that the proposed model has significantly different from the comparison models, and thus the MPDF model has a better generalization ability.

5. Conclusions

In this paper, a multi-pattern deep fusion approach (MPDF), combining the AP algorithm for identifying the passenger flow pattern and the DBN framework for learning sophisticated flow features, has been reported to forecast the short-term bus passenger flow. Three steps of the MPDF are overviewed as follows: Stage 1 (pattern identification), the AP is operated for clustering the passenger flow into several exemplars according to their similarities in distances, which are then grouped into special patterns in terms of passenger flow distribution analysis; Stage 2 (deep representation of each pattern), the DBN is modeled for learning the features of each pattern, generating hybrid DBNs for the deep representations of the passenger flow; Stage 3 (pattern fusion), the outputs of the multi-pattern DBNs are then fused as the final forecasts via the chronological order rearrangement. A case study of the bus Line 10 in Guangzhou city, China, is utilized to investigate the forecasting capacity of the proposed model. The experiments reveal its better pattern identification and feature learning abilities. Comparisons with other parametric and non-parametric methods are also studied. The results show that the addressed method exhibits the best forecasting performance among all the peer methods, and the proposed model has significantly different from the comparison models.

The innovation of the proposed approach is that the sophisticated features of the short-term bus passenger flow are characterized by different patterns under the deep frameworks, leading to better feature learning and forecasting performance. In a future study, the authors will focus on the popularization and application of the proposed approach.

Acknowledgements

This work is supported in part by the National Key Research & Development Program of China (2016YFE0132200),

Table 7
Comparison of the forecasting performances using different models.

Model	Performance criteria			Elapsed time (second)
	MAPE (%)	RMSE	R^2	
MPDF (proposed model)	10.743	157.312	0.958	18.003
DBN	20.114	203.713	0.939	8.741
AE	24.497	294.478	0.893	6.561
AP-AE	15.772	180.623	0.947	16.493
FFNN	27.095	238.612	0.914	23.302
ARIMA	29.072	293.388	0.873	7.845

Table 8
Wilcoxon signed ranks test between the proposed model and the comparison models.

Model-Model	MPDF-DBN	MPDF-AE	MPDF-(AP-AE)	MPDF-FFNN	MPDF-ARIMA
Significance	0.010	0.021	0.018	0.013	0.000

the Postdoctoral Science Foundation of China (2016M602459), and the Research Program of Higher Education of Guangdong (2016KZDXM054, 2016KQNCX165). We also extend special thanks to the editor/reviewers for their valuable comments in improving the quality of this paper.

References

[1] Y. Wei, M.C. Chen, Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks, Transp. Res. C 21 (2012) 148–162.
[2] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis: Forecasting and Control, Wiley, Hoboken, NJ, 2013.
[3] S. Anvari, S. Tuna, M. Canci, M. Turky, Automated Box-Jenkins forecasting tool with an application for passenger demand in urban rail systems, J. Adv. Transp. 50 (1) (2015) 25–49.
[4] M. Milenković, L. Švadlenka, V. Melichar, N. Bojović, Z. Avramović, SARIMA modelling appoach for rail way passenger flow forecasting, Transport (2016), http://dx.doi.org/10.3846/16484142.2016.1139623.
[5] M. Gan, Y. Cheng, K. Liu, G. Zhang, Seasonal and trend time series forecasting based on a quasi-linear autoregressive model, Appl. Soft Comput. 24 (2014) 13–18.
[6] A. Hofleitner, R. Herring, A. Bayen, Arterial travel time forecast with streaming data: a hybrid approach of flow modeling and machine learning, Transp. Res. B 46 (2012) 1097–1122.
[7] F. Dou, J. Xu, L. Wang, L. Jia, A train dispatching model based on fuzzy passenger demand forecasting during holidays, J. Ind. Eng. Manage. 6 (1) (2013) 320–335.
[8] J. Guo, W. Huang, B.M. Williams, Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification, Transp. Res. C 43 (2014) 50–64.
[9] X. Jiang, L. Zhang, X. Chen, Short-term forecasting of high-speed rail demand: a hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China, Transp. Res. C 44 (2014) 110–127.
[10] R. Chen, C.Y. Liang, W.C. Hong, D.X. Gu, Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm, Appl. Soft Comput. 26 (2015) 435–443.
[11] Y. Sun, B. Leng, W. Guan, A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system, Neurocomputing 166 (2015) 109–121.
[12] C. Babu, B. Reddy, A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data, Appl. Soft Comput. 23 (2014) 27–38.
[13] G. Xie, S. Wang, K.K. Lai, Short-term forecasting of air passenger by using hybrid seasonal decomposition and least squares support vector regression approaches, J. Air Transp. Manage. 37 (2014) 20–26.
[14] N. Glišović, M. Milenković, N. Bojović, L. Švadlenka, Z. Avramović, A hybrid model for forecasting the volume of passenger flows on Serbian railways, Oper. Res. 16 (2) (2016) 271–285.
[15] S. Lee, Y.I. Lee, B. Cho, Short-term travel speed prediction models in car navigation systems, J. Adv. Transp. 40 (2) (2006) 123–139.
[16] T.H. Tsai, C.K. Lee, C.H. Wei, Neural network based temporal feature models for short-term railway passenger demand forecasting, Expert Syst. Appl. 36 (2) (2009) 3728–3736.
[17] X. Zhang, B. Mao, Y. Wang, J. Feng, M. Li, Wavelet neural network-based short-term passenger flow forecasting on urban rail transit, Telkomnika Indonesian J. Electr. Eng. 11 (2013) 7379–7385.
[18] Y. Chen, B. Yang, Q. Meng, Small-time scale network traffic prediction based on flexible neural tree, Appl. Soft Comput. 12 (1) (2012) 274–279.
[19] Y. Bai, P. Wang, J.J. Xie, J. Li, C. Li, An additive model for monthly reservoir inflow forecast, J. Hydrol. Eng. 20 (7) (2015) 04014079.
[20] Z.L. Ma, J.P. Xing, M. Mesbah, L. Ferreira, Predicting short-term bus passenger demand using a pattern hybrid approach, Transp. Res. C 39 (2014) 148–163.
[21] Z. Nadezda, R. Andrejs, M. Yuri, Transport simulation model calibration with two-step cluster analysis procedure, Inf. Technol. Manage. Sci. 18 (1) (2015) 49–56.
[22] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 305 (5814) (2007) 972–976.
[23] M. Leone, S.M. Weigt, Clustering by soft-constraint affinity propagation: applications to gene-expression data, Bioinformatics 23 (20) (2007) 2708–2715.
[24] H. Su, Y. Sheng, P. Du, K. Liu, Adaptive affinity propagation with spectral angle mapper for semi-supervised hyperspectral band selection, Appl. Opt. 51 (14) (2012) 2656–2663.
[25] F. Shang, L.C. Jiao, J. Shi, F. Wang, M.G. Gong, Fast affinity propagation clustering: a multilevel approach, Pattern Recognit. 45 (2012) 474–486.
[26] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[27] Y. Bengio, Learning deep architectures for AI. Found, Trends Mach. Learn. 2 (2009) 1–127.
[28] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. Syst. 19 (2007) 153–160.
[29] C. Li, Y. Bai, B. Zeng, Deep feature learning architectures for daily reservoir inflow forecasting, Water Resour. Manage. 30 (14) (2016) 5145–5161.
[30] C. Li, D. Cabrera, J. Oliveira, R.V. Sanchez, M. Cerrada, G. Zurita, Extracting repetitive transients for rotating machinery diagnosis using multiscale clustered grey infogram, Mech. Syst. Sig. Process. 76–77 (2016) 157–173.
[31] M.M.A. Rahhal, Y. Bazi, H. Alhichri, N. Alajlan, F. Melgani, R.R. Yager, Deep learning approach for active classification of electrocardiogram signals, Inf. Sci. 345 (2016) 340–354.
[32] C. Li, R.V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, R. Vásquez, Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals, Mech. Syst. Sig. Process. 76–77 (2016) 283–293.
[33] Y. Bai, Z.Q. Chen, J.J. Xie, C. Li, Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models, J. Hydrol. 532 (2016) 193–206.
[34] W. Huang, H. Hong, M. Li, W. Hu, G. Song, K. Xie, Deep architecture for traffic flow prediction, Int. Conf. Adv. Data Min. Appl. Springer-Verlag (2013) 165–176.
[35] 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.
[36] R. Refianti, A.B. Mutiara, A.A. Syamsudduha, Performance evaluation of affinity propagation approaches on data clustering, Int. J. Adv. Comput. Sci. Appl. 7 (3) (2016) 420–429.
[37] X.Y. Xu, J. Liu, H.Y. Li, J.Q. Hu, Analysis of subway station capacity with the use of queueing theory, Transp. Res. C 38 (2014) 28–43.
[38] Y. Bengio, A. Courville, P. Vincent, Unsupervised feature learning and deep learning: A review and new perspectives Cornell University Library 2014 arXiv:1206.5538v3.
[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[40] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognit. Lett. 42 (6) (2014) 11–24.
[41] D. Janez, 2006: Statistical comparisons of classifiers over multiple datasets, J. Mach. Learn. Res. 7 (2016) 1–30.
[42] F.X. Diebold, R.S. Mariano, Comparing predictive accuracy, J. Bus. Econ. Stat. 13 (3) (1995) 253–263.