A Novel DBN Model for Time Series Forecasting

Yongpan Ren, Jingli Mao, Yong Liu, Yingzhe Li

***Abstract*—Deep Belief Network (DBN) via stacking Restricted Boltzmann Machines (RBMs) has been successfully applied to time series forecasting. In this paper, a novel DBN model composed of two RBMs is proposed for time series forecasting, in which Gaussian-Bernoulli RBM (GBRBM) is adopted for continuous input and an Artificial Neural Networks (ANN) is exploited to supervised learning respectively. Specifically, a novel error correction algorithm is introduced to further improve the forecasting accuracy. Experiments results verify the effectiveness of our proposed DBN model and show better forecasting performance.**

***Index Terms*—restricted boltzmann machine, gaussian-bernoulli restricted boltzmann machine, deep brief network.**

I. Introduction

TIME series forecasting is used for forecasting the future based on historical observations in various domains, such as Egypt Wheat Imports [1], stock forecasting [2], etc. In Statistics, the majority of the reported time series forecasting models (such as, Autoregressive Moving Average Models (ARMA), Autoregressive Integrated Moving Average Models (ARIMA) [3], etc.) focus on the analysis of linear problems as regression methods rather than nonlinear problems. However, in practice, many nonlinear problems exist in the real world where traditional linear estimators are not applicable any more. Recent works on time series forecasting models have also shown that the performance of forecasting results can be significantly improved by capturing the nonlinear relationship associated to the structure of a given input datasets.

The majority of existing machine learning methods, e.g. Support Vector Machines (SVMs) [4] and ANN [5], demonstrated to be powerful non-linear estimators, not only have been used on the field of pattern recognition or dimensionality reduction [6], but also have been used in time series forecasting. Although these models achieved fairly good results, they are still not the most effective models due to their defect of shallow learning. In 2006, Hinton et al. found that the performance of a deep neural network could be significantly improved when a non-supervised learning algorithm is used, pretraining one layer after another, starting from the first layer [7]. The deep learning algorithm can be regarded as a learning process, which can learn more abstract features of data represented by higher levels [8]. It can be more useful in extracting information for classification or prediction for more abstract representations [9]. Besides these advantages, the learned intermediate features can be shared among different tasks. Therefore, to learn the kind of complicated features that can represent high-level abstractions, deep learning architectures are needed [10]. DBN [11] consisting of a stack of RBMs [12], as a deep architecture, has been successfully applied to many complex and high-dimensional sequences [13].

In [14], a greedy layer-wise unsupervised training strategy is adopted based on DBN to bring better generalization. In [11], the authors proposed an approach depending on DBN in clustering and classification of continuous input data, which has a better performance than the traditional ANN due to the initialization of the connecting weights rather than just using random weights in ANN. In [13], a 3-layer DBN network of RBMs is proposed to capture the feature of input space of time series data, in which particle swarm optimization (PSO) is adopted during the training processes. In [15], an ensemble of multiple DBNs is proposed for time series forcasting, in which the outputs from various DBNs were aggregated by a Support Regression Vector (SVR) model. Similarly, an ensemble of classifiers is proposed by integrating RBMs with bagging to generate diverse and accurate individual classifiers [16]. Authors of [17] proposed a DBN architecture composed of RBM and multi-layer perceptron (MLP) to predict chaotic time series data. In [18], the conventional RBMs are extended to model high-dimensional motion time series data by introducing an extra term in the energy function to explicitly model the local spatial interactions in the input data. Furthermore, the authors proposed a method using not only a kind of DBN with RBM and MLP but also ARIMA to improve the forecasting accuracy in [19].

In the conventional RBM, each visible neuron is represented by a binary variable, and real-valued data is normalized to [0, 1] and treated as a probability, that is RBM learns distributions over binary vectors. However, this representation is restricted to bounded variables. Therefore, in [6], the binary visible neurons are replaced with Gaussian ones to address this problem. The corresponding model is called Gaussian-Bernoulli Restricted Boltzmann Machine (GBRBM). In [20], a few remedies to the conventional training methods for G-BRBM are proposed. In [21], GBRBM and Deep Boltzmann Machine (DBM) are combined together in a single model, allowing their joint optimization. The corresponding model is called Gaussian-Bernoulli Deep Boltzmann Machine (GDB-M).

In this paper, we introduce a novel DBN model for time series forecasting that can further improve the forecasting accuracy. The proposed DBN model is composed of two RBMs, in which GDBM using Gaussian units in the visible layer of DBM is adopted. An ANN after the unsupervised learning is exploited to forecast and error back-propagation

Manuscript received March 11, 2016; revised December 19, 2016. This work is supported by National Natural Science Foundation of China (No.61501041), Open Foundation of State Key Laboratory (No. ISN16-08) and Huawei Innovation Research Program (HIRPO20140512).

Yongpan Ren and Yong Liu are with Beijing University of Posts and Telecommunications, Beijing 100876, China and The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail:*ren708818305@126.com; liuyongamtf@aliyun.com*)

Jingli Mao is with Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail:*mjl018@126.com*).

Yingzhe Li is with Huawei Technologies Co., Ltd., Shenzhen 201206, China (e-mail:*yingzhe.li@huawei.com*).

(BP) [22] is adopted to fine-tuning. Specifically, a novel error correction algorithm is introduced to further improve the forecasting accuracy.

Extensive evaluations, carried out on the three well-known datasets, show that the proposed novel DBN model leads to improvement in forecasting accuracy compared to the two state-of-the-art models.

The rest of this paper is organized as follows. Section 2 provides a brief review of the theories of RBM. In Section 3, a novel DBN model together with error correction algorithm is proposed. Experiments are presented in Section 4, and some concluding remarks are given in Section 5.

II. Review to Restricted Boltzmann Machine

Boltzmann Machines (BMs) can be regarded as undirected graphical models also known as Markov Random Fields (MRF). An RBM is an MRF associated with a bipartite undirected graph, which consists of m visible units $\boldsymbol{v} = (v_1, \cdots, v_m)$ to represent observable data and n hidden units $\boldsymbol{h} = (h_1, \cdots, h_n)$ to capture dependencies between observed variables. In binary RBMs, the random variables $(\boldsymbol{v}, \boldsymbol{h})$ take values $(\boldsymbol{v}, \boldsymbol{h}) \in \{0, 1\}^{m+n}$. An RBM has only connections between the layer of hidden and visible variables but not between two variables of the same layer.

Being energy-based model, RBM has an energy with a joint configuration $(\boldsymbol{v}, \boldsymbol{h})$ of the visible and hidden units as follows

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_j v_i - \sum_{i=1}^{n} c_i v_i - \sum_{j=1}^{m} b_j h_j$$

where v_i, h_j are the binary states of visible unit i and hidden unit j respectively, c_i, b_j are their biases and w_{ij} is a real valued weight between them.

Considering that there is the link between the adjacent layers and no connection within the RBM layer, the posterior distributions given another neighbouring layer for the hidden and visible units take the form:

$$P(h_j = 1|\boldsymbol{v}) = sigm(\sum_{i=1}^{n} w_{ij} v_i + b_j)$$

$$P(v_i = 1|\boldsymbol{h}) = sigm(\sum_{j=1}^{n} w_{ij} h_j + c_i)$$

where $sigm(.)$ is the nonlinear activiation function. The logistic fuction, $sigm(x) = \frac{1}{1+e^{-x}}$, is a common choice.

The learning process of RBM is to find the model parameters $\boldsymbol{b} = (b_1, \cdots, b_m), \boldsymbol{c} = (c_1, \cdots, c_n), \boldsymbol{W} = (w_{ij}), i \in (1, \cdots, n), j \in (1, \cdots, m)$ to make the energy of the system minimum. Therefore, updating rules of parameters are given as follows

$$\Delta w_{ij} = \epsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$$

$$\Delta c_i = \epsilon(\langle v_i \rangle_{data} - \langle v_i \rangle_{model})$$

$$\Delta b_j = \epsilon(\langle h_j \rangle_{data} - \langle h_j \rangle_{model})$$

where ϵ is a learning rate, $\langle . \rangle_{data}$ and $\langle . \rangle_{model}$ are used to represent the expected values of the data and the model respectively.

To fast the learning procdure, a simple and efficient method called Contrastive Divergence (CD) is proposed in [12], which starts by setting the states of the visible units as a training vector. Moreover, conventional RBM defines the state of each neuron to be binary, which seriously limits their application area. To solve this problem, GBRBM is introduced, in which the binary visible units are replaced by linear units with independent Gaussian noise [6]. The energy function is then extended as

$$E(\boldsymbol{v}, \boldsymbol{h}) = \sum_{i=1}^{n} \frac{(v_i - c_i)^2}{2\sigma_i^2} - \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{j=1}^{m} b_j h_j$$

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

With the GBRBM energy function, the conditional probabilities could be achieved as follows.

$$P(h_j = 1|\boldsymbol{v}) = sigm(\sum_{i=1}^{n} \frac{w_{ij} v_i}{\sigma_i^2} + b_j)$$

$$P(v_i = 1|\boldsymbol{h}) = N(v_i | \sum_{j=1}^{n} w_{ij} h_j + c_i, \sigma_i^2)$$

where $N(.|\mu, \sigma_i^2)$ is probability density of Normal distribution with a mean μ and a standard deviation σ_i.

Then the corresponding updating rules of RBM parameters are as follows

$$\Delta w_{ij} = \epsilon(\langle \frac{v_i h_j}{\sigma_i} \rangle_{data} - \langle \frac{v_i h_j}{\sigma_i} \rangle_{model})$$

$$\Delta c_i = \epsilon(\langle \frac{v_i}{\sigma_i^2} \rangle_{data} - \langle \frac{v_i}{\sigma_i^2} \rangle_{model})$$

$$\Delta b_j = \epsilon((h_j)_{data} - (h_j)_{model})$$

where ϵ is the learning rate.

III. Proposed DBN Model

In this section, an novel DBN model is proposed for time series forecasting. The architecture of the proposed DBN model is given in Section 3.1. An error correction algorithm contained in the model will be stated in Section 3.2.

A. Architecture of proposed DBN model

The proposed DBN model for time series forecasting is a type of deep learning architecture, which is composed of three modules as shown in Fig.1.

Assume the time series data as $x(t), t = 1, \cdots, T$, where T is the number of samples of the time series. The input data for forecasting is denoted as $x(t - \tau), \cdots, x(t - M\tau)$, where M is the window size of input data and τ is a positive integer representing the interval between delays. The forecasting output of our proposed DBN model is $y(t)$.

From Fig.1 we can see that there are three modules from bottom to top. The bottom module contains two RBMs used for unsupervised learning. The hidden units extract relevant features from the input data of previous M times $\{x(t-\tau), \cdots, x(t-M\tau)\}$. These features can serve as input to another RBM. By stacking RBMs in this way, features can be learned in the expectation of arriving at a high-level representation.

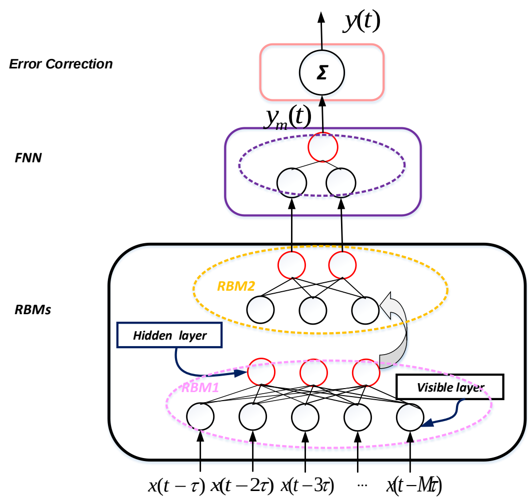


Fig. 1. Architecture of proposed DBN model

The middle module includes an ANN used as a supervised learning system. A common ANN is the feedforward neural network (FNN). In the proposed DBN model, the features extracted by the bottom module will serve as input. In fact, the hidden layer of the second RBM is the input of the FNN. In this module, BP algorithm is used for fine-tuning, which repeatedly adjusts the weights of the connections in the FNN network so as to minimize the measurement of difference between the actual output vector of the network and the desired output vector [22]. By this module, the forecasting output of current time could be achieved, which is denoted as $y_m(t)$ in Fig.1.

The top module is involved in an error correction algorithm used for improving the forecasting accuracy further. Generally, in time series, the values of the neighboring times show good correlation. Hence, the known actual value of previous time can be exploited to forecast the value of current time. In this case, an error correction algorithm is proposed to forecast the output $y(t)$ of current time, which will be described in detail in next subsection.

B. Error correction algorithm

The proposed error correction algorithm could be implemented as shown in Fig.2.

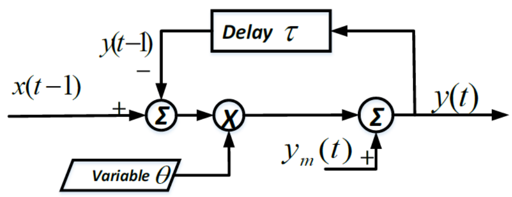


Fig. 2. Schematic diagram of error correction algorithm

In the Fig.2, $x(t-1)$ represents the actual value of previous time, $y_m(t)$ represents the output of FNN network, and variable θ is a parameter used to adjust the final output $y(t)$. Specifically, a formula is given according to the Fig.2 as follows

$$y(t) = y_m(t) + \theta\left[x(t-1) - y(t-1)\right], \theta \in [0,1] \qquad (1)$$

The parameter θ in Eq.(1) will be calculated in the training phase according to the following rule

$$arg \min_{\theta} \sqrt{\frac{\sum_i^{N_1} \left(y(t)_i - x(t)_i\right)^2}{N_1}}$$

where N_1 represents the number of forecasting values able to be obtained in the training phase, $y(t)_i$ is the forecasting output corresponding to its actual value $x(t)_i$ for the i-th forecasting value, which is represented by Eq.(1).

The pseudo-code for the procedure of searching the optimal θ is presented in Table I.

TABLE I
ALGORITHM OF SEARCHING OPTIMAL θ

Input : output value $y_m(k)$ of FNN, the known actual value $x(k)$, $for\ k = 1, \cdots, n$.
Output : the optimal θ_{opt} according to Eq.(1).
Step : updating step size of searching optimal θ.

$\boldsymbol{for}$ $all\ index \in \left[0, \frac{(1-0)}{step}\right]$ $\boldsymbol{do}$

  $\boldsymbol{for}$ $every\ k = 1, \cdots, n$ $\boldsymbol{do}$

    $y(k) \leftarrow y_m(k) + \theta(index)\left[x(k-1) - y(k-1)\right]$

  $\boldsymbol{end}$

  $RMSE\left[index\right] \leftarrow sqrt\{\sum_j (x(j) - y(j))^2/n\}$

$\boldsymbol{end}$

$[RMSE(min), \theta_{opt}] \leftarrow min\{RMSE\left[1\right], \cdots, RMSE\left[k\right],$
$\cdots, RMSE\left[n\right]\}$

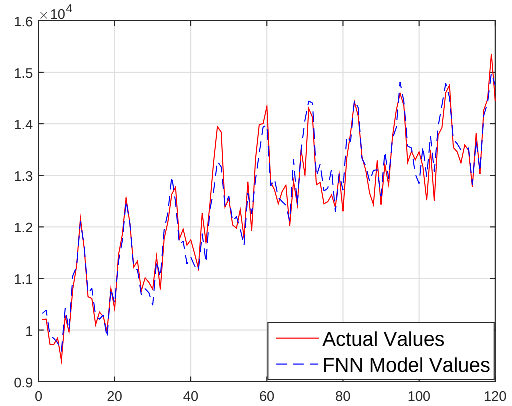
$\boldsymbol{return}$ θ_{opt}

IV. EXPERIMENTS

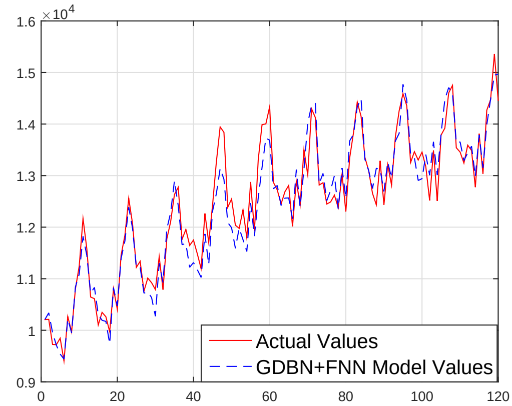
To evaluate the effectiveness of the proposed DBN model, three time series, Australia Energy Production *[Energy]* and Dollar to Libra Conversion *[Dollar]* obtained in [23] and Taiwan Stock Exchange Capitalization Weighted Stock Index *[TAIEX]* available in [24] are used to make predictions. *[Energy]* shows the electric energy monthly production from Australia from January 1956 up to August 1995 with 476 samples, *[Dollar]* shows monetary information monthly from US$ dollars conversion to Libra from January 1981 up to July 2005 with 295 values, and *[TAIEX]* from January 2004 to December 2004 is composed of 260 samples.

Moreover, based on the aforementioned three datasets, comparison experiments of the proposed model and the two state-of-the-art models, i.e., FNN model and GDBM+FNN model in [23], are conducted.

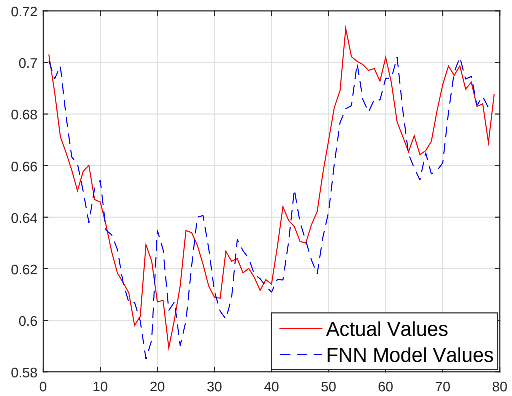
In the experiments, 70% of the samples for each time series are used to train and 30% used to test. The parameter of window size M is set to 8, 5 and 7 for *[Energy]*, *[Dollar]* and *[TAIEX]* respectively, and the interval τ between delays is set to 3, 2 and 1 accordingly.



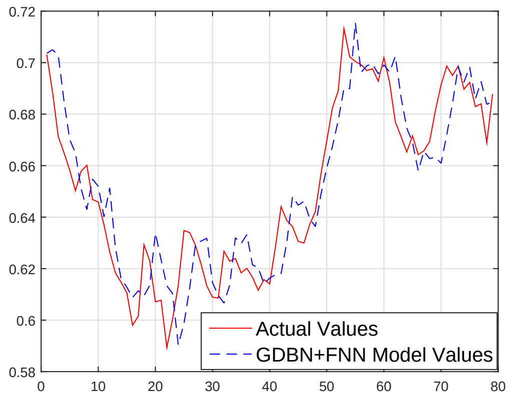
(a) FNN Model with [Energy]



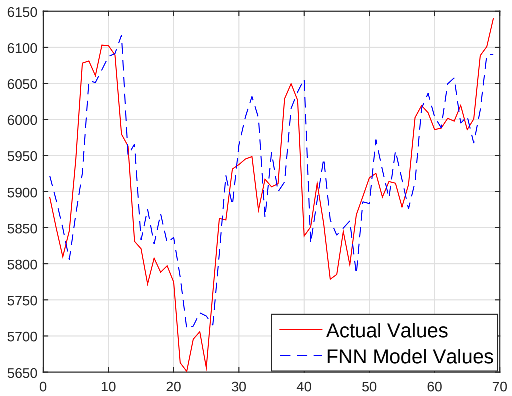
(d) GDBM+FNN Model with [Energy]



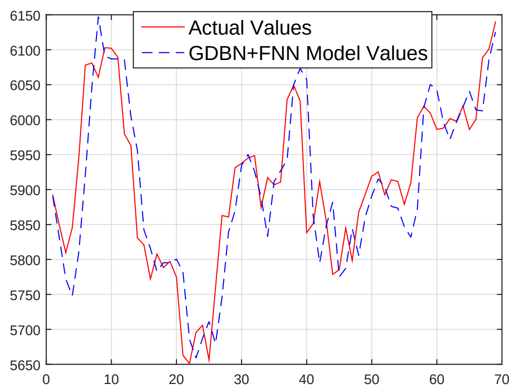
(b) FNN Model with [Dollar]



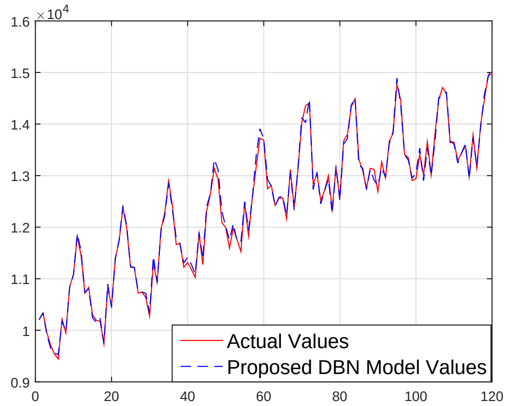
(e) GDBM+FNN Model with [Dollar]



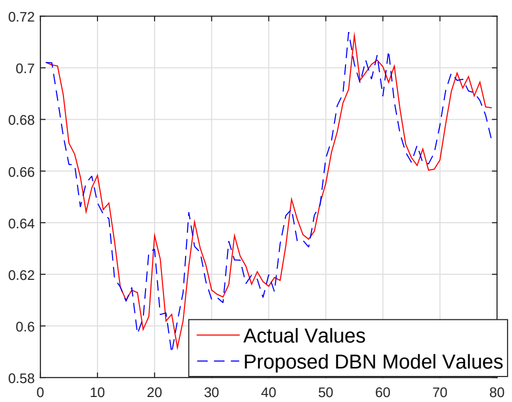
(c) FNN Model with [TAIEX]



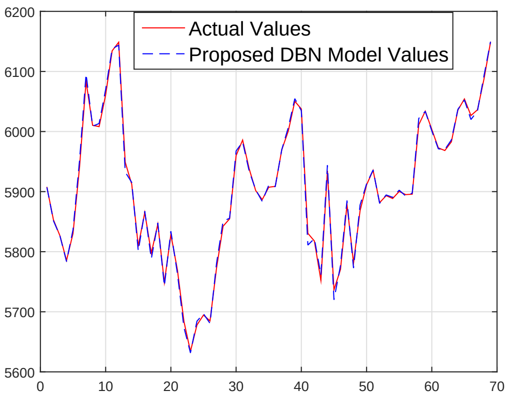
(f) GDBM+FNN Model with [TAIEX]



(g) Proposed DBN Model with [Energy]



(h) Proposed DBN Model with [Dollar]



(i) Proposed DBN Model with [TAIEX]

Fig. 3. Forecasting results of different models with [Energy], [Dollar] and [TAIEX]

The forecasting results of the FNN model, the GDBM+FNN model and proposed DBN model using three time series *[Energy]*, *[Dollar]* and *[TAIEX]* respectively are shown in Fig.3.

In the Fig.3, (a), (d) and (g) show the results of three models using time series *[Energy]*, and (b), (e) and (h) show the results of three models using time series *[Dollar]*, and (c), (f) and (i) show the results of three models using time series *[TAIEX]*. From Fig.3, we can see that the forecasting results of our proposed model fit more close to the actual values of each time series compared to FNN model and GDBM+FNN model.

To obtain comprehensive evaluation for the performance of the proposed DBN models in the testing phase, three criteria, i.e., root mean square error (RMSE), mean absolute error (MAE) and percentage of mean absolute error (MAPE), are proposed, which are defined in Eq.(2), Eq.(3) and Eq.(4) respectively.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{P}\sum_{j=1}^{N_2}(x(t)_{ij} - y(t)_{ij})^2}{P * N_2}} \quad (2)$$

$$MAE = \frac{1}{P * N_2}\sum_{i=1}^{P}\sum_{j=1}^{N_2}|(x(t)_{ij} - y(t)_{ij})| \quad (3)$$

$$MAPE = \frac{1}{P * N_2}\sum_{i=1}^{P}\sum_{j=1}^{N_2}\frac{|(x(t)_{ij} - y(t)_{ij})|}{|x(t)_{ij}|} \quad (4)$$

where P is the number of patterns in the data set, N_2 represents the number of output units of the model, $x(t)_{ij}$ and $y(t)_{ij}(i = 1, \cdots, P,\ j = 1, \cdots, N_2)$ are the actual value and forecasting value in the training phase respectively. In the experiment, N_2 is set to 1 for time series prediction.

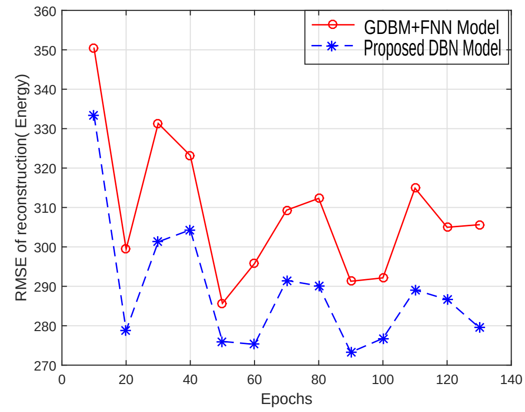
TABLE II
THE RESULTS OF THREE CRITERIA ON DIFFERENT VALUES OF EPOCH FOR ENERGY

Times Series	Criterion	FNN	GDBM+FNN	Our DBN
$[Energy]$	$RMSE$	320.5760	304.4497	289.3716
	MAE	258.9582	226.5737	211.8005
	$MAPE$	2.0825	1.7931	1.6689
$[Dollar]$	$RMSE$	0.0147	0.0140	0.0106
	MAE	0.0122	0.0119	0.0089
	$MAPE$	1.8793	1.8369	1.3689
$[TAIEX]$	$RMSE$	72.7805	68.1714	57.6623
	MAE	55.2538	50.1753	41.4932
	$MAPE$	0.9368	0.8492	0.7034

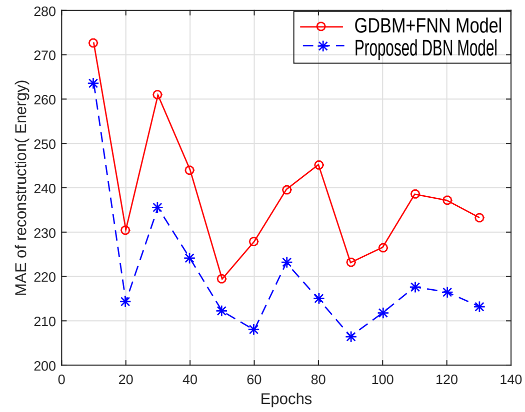
In terms of RMSE, the results for three time series under FNN model, GDBM+FNN model and proposed DBN model are shown in Table II.

It can be seen from that Table II, for time series *[Energy]*, FNN model and GDBM+FNN model achieve 320.5760 and 304.4497 of RMSE values respectively. The RMSE of our proposed DBN model is 289.3716. Also, for the time series *[Dollar]* and *[TAIEX]*, our proposed model shows its priority compared to the other two models. Similarly, smaller values of MAE and MAPE could be achieved by our proposed model based on time series *[Dollar]* and *[TAIEX]* compared to FNN model and GDBM+FNN model.

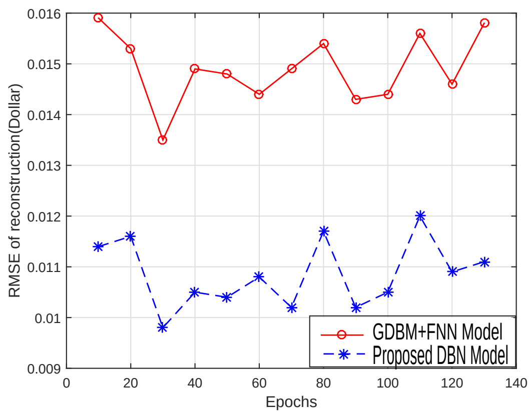
Next, to further evaluate the performance of the proposed DBN model, the evolution of each criterion is investigated as the number of the training epochs increases from 10 to 130 for the GDBM+FNN model and our proposed DBN model based on the three time series and the results are shown in Fig.4.



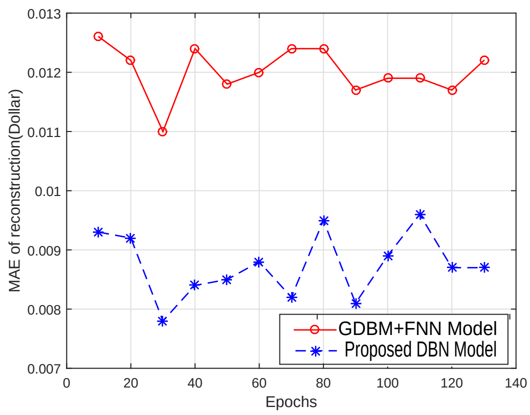
(a) RMSE with [Energy]



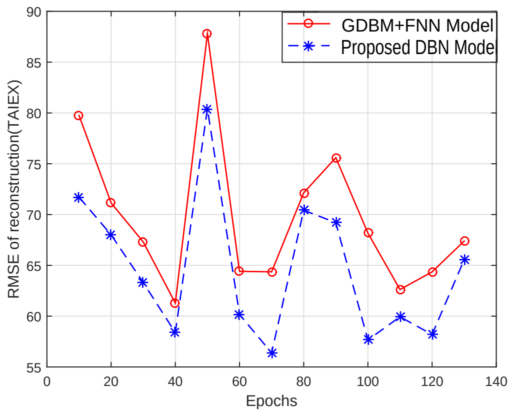
(d) MAE with [Energy]



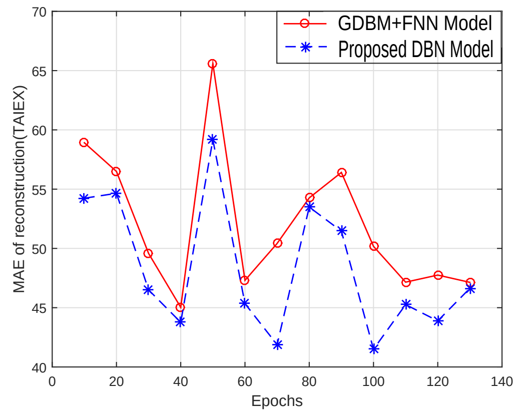
(b) RMSE with [Dollar]



(e) MAE with [Dollar]

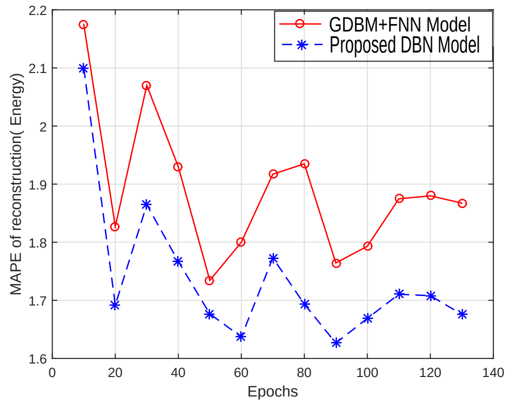


(c) RMSE with [TAIEX]

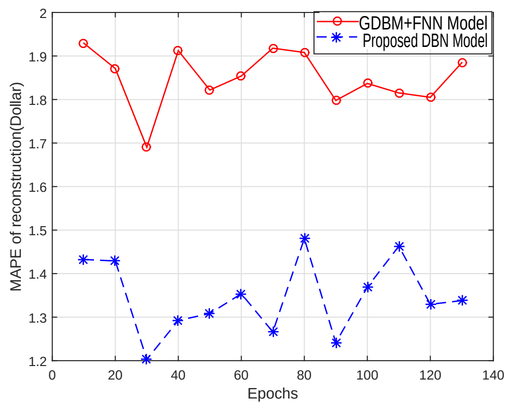


(f) MAE with [TAIEX]

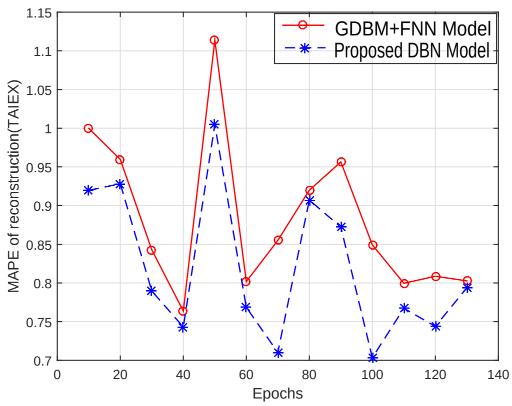
In Fig.4, (a), (b) and (c) represent the RMSE values of GDBM+FNN model and our proposed DBN model based on time series *[Energy]*, *[Dollar]* and *[TAIEX]* respectively. Fig.4 (d), (e) and (f) represent the MAP values of the two models based on three time series respectively. And Fig.4 (g), (h) and (i) represent the MAPE values accordingly. The horizontal axis represents the number of the training epochs and the vertical axis represents the corresponding error values. In Fig.4, the red solid lines represent results of GDBM+FNN model, while the blue dotted lines represent results of our proposed DBN model respectively. It is illustrated from Fig.4 that no matter which criterion is used, our proposed DBN model has smaller values. The forecasting performance of our proposed DBN model is thereby verified.



(g) MAPE with [Energy]



(h) MAPE with [Dollar]



(i) MAPE with [TAIEX]

Fig. 4. Evolution of criteria based on different time series with *[Energy]*, *[Dollar]* and *[TAIEX]* for GDBM+FNN Model and Proposed DBN Model

V. Conclusion

A novel DBN model composed of two RBMs and an ANN is proposed for time series forecasting in this paper, in which GBRBM for continuous input is adopted. Moreover, to further improve the forecasting accuracy, a novel error correction algorithm is proposed. The forecasting performance of the proposed DBN model is verified based on three time series such as *[Energy]*, *[Dollar]* and *[TAIEX]*. Experiments results show that the proposed model is effective and able to achieve higher forecasting accuracy compared to FNN model and GDBM+FNN model.

References

[1] Abd-Elaal A K, Hefny H A, Abd-Elwahab A H. Forecasting of egypt wheat imports using multivariate fuzzy time series model based on fuzzy clustering[J]. *IAENG International Journal of Computer Science, 2013, 40(4): 230-237.*

[2] Khiabani K, Aghabozorgi S R. Adaptive time-variant model optimization for fuzzy-time-series forecasting[J]. *IAENG International Journal of Computer Science, 2015, 42(2): 107-116.*

[3] Box G E P, Pierce D A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models[J]. *Journal of the American statistical Association, 1970, 65(332): 1509-1526.*

[4] Sapankevych N and Sankar R. Time series prediction using support vector machines: a survey. *Computational Intelligence Magazine 4 (2) (2009) (24-38).*

[5] Kuremoto T, Kimura S, Kobayashi K and Obayashi M. Time series forecasting using a deep belief network with restricted Boltzmann machines. *Neurocomputing 137 (2014) 47-56.*

[6] G. E. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science, vol. 313, no. 5786, pp. 504-507, July 2006.*

[7] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. *Neural computation, 2006, 18(7): 1527-1554.*

[8] Y. Bengio. Deep learning of representations: Looking forward[M]. *Statistical language and speech processing. Springer Berlin Heidelberg, 2013. 1-37.*

[9] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(8): 1798-1828.*

[10] Bengio Y. Learning Deep Architectures for AI[J]. *Foundations & Trends in Machine Learning, 2009, 2(1):1-127.*

[11] M. A. Salama, A. E. Hassanien, and A. A. Fahmy. Deep Belief Network for clustering and classification of a continuous data. *in Proceedings of the 10th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 10), pp. 473-477, December 2010.*

[12] A. Fischer and C. Igel. An introduction to restricted Boltzmann machines. *In L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, editors, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, volume 7441 of Lecture Notes in Computer Science, pages 14-36. Springer Berlin Heidelberg, 2012.*

[13] Kuremoto T, Kimura S and Kobayashi K. Time series forecasting using a deep belief network with restricted Boltzmann machines[J]. *Neurocomputing, 2014, 137: 47-56.*

[14] Schlkopf. B, Platt. J and Hofmann. T. Greedy layer-wise training of deep networks[C]. *In NIPS, 2007:153-160.*

[15] Qiu X, Zhang L and Ren Y. Ensemble deep learning for regression and time series forecasting[C]. *IEEE Symposium on Computational Intelligence in Ensemble Learning, 2014:1-6.*

[16] Zhang C X, Zhang J S, Ji N N, et al. Learning ensemble classifiers via restricted Boltzmann machines[J]. *Pattern Recognition Letters, 2014, 36: 161-170.*

[17] Kuremoto T, Obayashi M, Kobayashi K, et al. Forecast chaotic time series data by DBNs[C]. *Image and Signal Processing (CISP), 2014 7th International Congress on. IEEE, 2014: 1130-1135.*

[18] Nie S, Wang Z, Ji Q. A generative restricted Boltzmann machine based method for high-dimensional motion data modeling[J]. *Computer Vision and Image Understanding, 2015, 136: 14-22.*

[19] Hirata T, Kuremoto T, Obayashi M, et al. Time Series Prediction Using DBN and ARIMA[C]. *Computer Application Technologies (CCATS), 2015 International Conference on. IEEE, 2015: 24-29.*

[20] K. Cho, A. Ilin, and T. Raiko. Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. *in Proceedings of the 20th International Conference on Artificial Neural Networks (ICANN 2010), 2011.*

[21] Cho K H, Raiko T and Ilin A. Gaussian-bernoulli deep boltzmann machine[C]. *Neural Networks (IJCNN), The 2013 International Joint Conference on. IEEE, 2013: 1-7.*

[22] Rumelhart D E, Hinton G E and Williams R J. Learning representations by back-propagating errors. *Cognitive modeling 323 (9) (1986) 533-536.*

[23] Hrasko R, Pacheco A G C and Krohling R A. Time Series Prediction Using Restricted Boltzmann Machines and Backpropagation[J]. *Procedia Computer Science, 2015, 55: 990-999.*

[24] Taiwan Stock Exchange Capitalization Weighted StockIndex (TAIEX). *[Online].Available:http://www.twse.com.tw/en/products/indices/tsec/taiex.php*