

Short term load forecasting based on deep learning for smart grid applications

Ghulam Hafeez¹, Nadeem Javaid^{1,*}, Safeer Ullah¹, Zafar Iqbal²
Mahnoor Khan¹, Aziz Ur Rehman^{1,3}, Ziaullah⁴

¹ COMSATS Institute of Information Technology, Islamabad 44000, Pakistan

² PMAS Arid Agriculture University, Rawalpindi 46000, Pakistan

³ The University of Lahore, Gujrat Campus, Pakistan

⁴ Huazhong University of Science and Technology, China

* Correspondence: nadeemjavaidqau@gmail.com, www.njavaid.com

Abstract. Short term load forecasting is indispensable for industrial, commercial, and residential smart grid (SG) applications. In this regard, a large variety of short term load forecasting models have been proposed in literature spanning from legacy time series models to contemporary data analytic models. Some of these models have either better performance in terms of accuracy while others perform well in convergence rate. In this paper, a fast and accurate short term load forecasting framework based on stacked factored conditional restricted boltzmann machine (FCRBM) and conditional restricted boltzmann machine (CRBM) is presented. The stacked FCRBM and CRBM are trained using rectified linear unit (ReLU) and sigmoid functions, respectively. The proposed framework is applied to offline demand side load data of US utility. Load forecasts decide weather to increase or decrease the generation of an already running generator or to add extra units or exchange power with neighboring systems. Three performance metrics i.e., mean absolute percentage error (MAPE), normalized root mean square (NRMSE), and correlation coefficient are used to validate the proposed framework. The results show that stacked FCRBM and CRBM are accurate and robust as compared to artificial neural network (ANN) and convolutional neural network (CNN).

1 Introduction

The energy demand is increasing rapidly with the increase in population. The current net energy demand of commercial and residential sector is about 40%-50% in developed countries of the world [1]. In order, to meet this increasing energy demand the entire globe is transitioning from traditional grid to smart grid (SG). The SG has the ability to forecast, monitor, plan, schedule, and make real time decisions regarding generation and consumption of energy. One of the features of SG is advanced metering infrastructure which realized the active participation of both utility and consumers in electricity market. The goals and objectives of SG are to improve efficiency and effective utilization of electricity framework. Modeling and forecasting load realized the achievement these objectives.

For SG distribution side performance optimization a proper decision making is necessary which leads to reduce cost, alleviate peaks, and power losses. The current research keeping the aforementioned objectives in mind perform power scheduling using optimization techniques [2], [3]. However, prior to optimization techniques based scheduling, load forecasting is necessary for efficient energy management. Typically, load forecasting have three categories: a) short term load forecast, ranging from hours to week, b) medium term forecast, for a duration of weeks to year, and c) long term forecast, for a duration more than one year. Especially, the focus of this work is on short term load forecasting, it is a challenging task due to stochastic and non-linear consumption behavior of consumers. Many short term load forecasting models have been proposed in [4]-[11]. However, these models have either accuracy or convergence rate problems. In [12], authors used ANN based forecasting model to reduce the error performance and improve the forecast accuracy. However, while improving forecast accuracy the computational complexity and execution time is compromised due to tradeoff between accuracy and convergence rate.

The contribution of this paper is twofold, first, we propose a new way to adopt stacked factored conditional restricted boltzmann machine (FCRBM) model with aim to learn the non-linear and stochastic electrical patterns from offline data. Second, a short term load forecasting model is proposed based on the stacked FCRBM and CRBM to improve the relative forecast accuracy. The proposed model is validated by comparing with existing short term load forecasting models based on ANN and CNN in terms in terms of mean absolute percentage error (MAPE), normalized root mean square error (NRMSE), and correlation coefficient. We adopt a modular strategy for short term load forecasting in which the output of each former module is fed into the later module. In short, our system model comprises of three modules data processing and feature extraction module, deep learning-based training module, and deep learning-based forecasting module.

The organisation of the paper is as follows: In Section 2, we introduce proposed methods. Section 3 demonstrates proposed architecture. In Section 4, simulation results are provided. Finally, the paper is concluded in Section 5.

2 Proposed methods

In this section, we introduce deep learning techniques CRBM and stacked FCRBM for short term load forecasting. For the aforementioned deep learning techniques, we describe three ingredients i.e., error function, conditional probability, and update/learning rules. Error function of a given network provides scalar values that are essential for the configuration. Conditional probability calculates the probability of an event over specific condition. Update/learning rules are required for tuning of the free parameters.

2.1 CRBM

CRBM [13] is an amendment in the RBM [14]. It is a probabilistic model used to model human activities, weather data, collaborative filtering, classification, and time series data [15]. In this paper, we use CRBM having three layers visible layer, hidden layer, and conditional history layer. The generic infrastructure of CRBM is shown in Figure 1. The detailed description of CRBM three ingredients such as error function, conditional probability, and learning rules are as follows.

Error function: The error function express the possible correlations between input, conditional history layer, hidden layer, and output. The error function is calculated as:

$$E(v, u, h; w) = -(v^T w^{vh} h + u^T w^{uv} v + u^T w^{uh} h + u^T a + h^T b), \quad (1)$$

where $v = [v_1, v_2, \dots, v_n]$ is the real valued vector having visible unit neurons from 1 to n neuron, $u = [u_1, u_2, \dots, u_n]$ shows the real valued vector having history neurons from 1 to n , $h = [h_1, h_2, \dots, h_n]$ denotes the binary vector having hidden neurons from 1 to n , w is the weight matrix, a is the visible layer bias, and b is the hidden layer bias. The weight matrix w^{vh} is bidirectional while the weight matrices w^{uh} and w^{uv} are unidirectional.

Conditional probability: Conditional probability in CRBM determines the probability distribution over two inferences. First inference $p(h/v, u)$, is to determine the probability of hidden layer inferenced on all the layers, while the second inference $p(v/h, u)$, is to determine the probability of visible layer conditioned on all the other layers. The two inferences are leading to:

$$p(h/v, u) = \text{sigmoid}(u^T w^{uh} + v^T w^{vh} + b) \quad (2)$$

$$p(v/h, u) = \text{sigmoid}(w^{uv} u^T + w^{vh} h + a) \quad (3)$$

Weights and biases learning and update rules: We use a stochastic gradient decent method for learning and updating the weights and biases of the layers because other methods some time have the problem of vanishing gradient which made the network hard to train. The parameters are fine tuned by maximizing the probability function and the weights and biases metrics are updated to minimize the gap between real and forecasted value.

The weights are updated as follows:

$$\begin{aligned} w_{t+1}^{uh} &= w_t^{uh} - \eta \frac{\partial E}{\partial w^{uh}} \\ w_{t+1}^{uv} &= w_t^{uv} - \eta \frac{\partial E}{\partial w^{uv}} \\ w_{t+1}^{vh} &= w_t^{vh} - \eta \frac{\partial E}{\partial w^{vh}}. \end{aligned} \quad (4)$$

The biases are updated by the following Equation:

$$\begin{aligned} a_{t+1} &= a_t - \eta \frac{\partial E}{\partial a^v} \\ b_{t+1} &= b_t - \eta \frac{\partial E}{\partial b^h} \end{aligned} \quad (5)$$

where η is the learning rate and t is the iteration number. The aforementioned procedure is repeated for the number of epochs until to converge the model.

2.2 Stacked FCRBM

FCRBM is an extension of the CRBM introduced by Taylor and Hinton in [15]. In FCRBM [16], they add the concept of factor and styles to mimic multiple human actions. We propose a new way to adopt deep learning technique i.e., stacked FCRBM for short term load forecasting, where the successive layers take output from the previous trained layers to overcome the problem of overfitting and to improve the forecast accuracy. The stacked FCRBM comprised of four layers as shown in Figure 2: a) visible layer v , b) history layer u , c) hidden layer h , and d) style layer y .

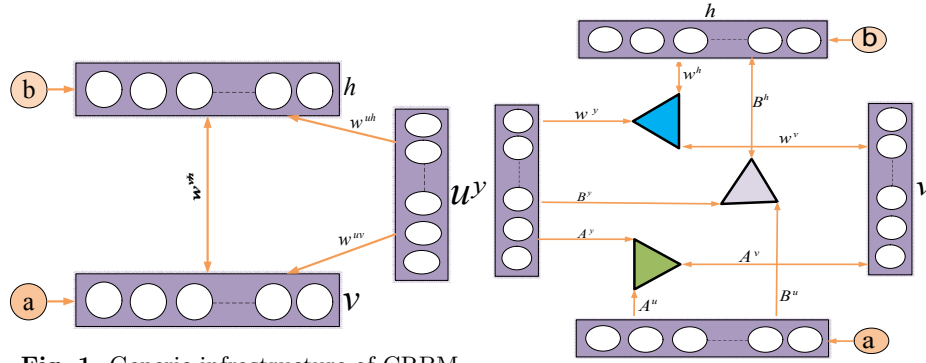


Fig. 1. Generic infrastructure of CRBM

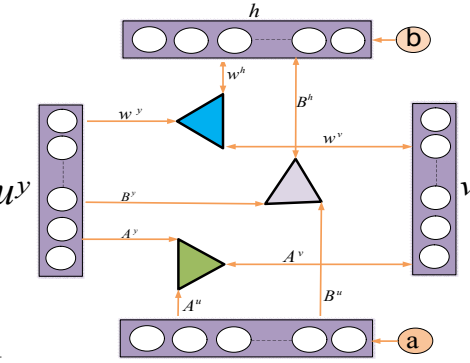


Fig. 2. Generic infrastructure of stacked FCRBM

The visible and history layers are real valued while the hidden layer is binary. The visible layer is responsible for encoding the present time series data to forecast the future value, while the history layer will encode historical time series data. Hidden layer is responsible for the discovery of significant features required for analysis. The different styles and parameters essential for forecasting are embedded into the style layer. The relation and interaction between the layers, weights, and factors is expressed by an error function as:

$$E(v, u, h; w) = -v^T \hat{a} - h^T \hat{b} - \sum \{(v^T w^v) \circ (y^T w^y) \circ (h^T w^h)\} \quad (6)$$

where E is the error function, $v^T w^v$ is the visible factored, $y^T w^y$ is the style factored, and $h^T w^h$ is the hidden factored. It $() \circ ()$ is the product known as

hadamard product, in which the product operation is element wise. The elements \hat{a} and \hat{b} represent dynamic biases for visible and hidden layers, respectively, which are defined as follows:

$$\begin{aligned}\hat{a} &= a + A^v \{ (u^T A^u) \circ (y^T A^y) \}^T \\ \hat{b} &= b + B^h \{ (u^T B^u) \circ (y^T B^y) \}^T\end{aligned}\tag{7}$$

where w^v , w^y , w^h are weights of the corresponding layers and A^v , A^u , A^y , B^h , B^u , B^y are the connections of the corresponding layers to factors, are known as model free parameters.

2.3 Conditional probability

In case of stacked FCRBM, conditional probability determines probability distribution of one layer conditioned over all the remaining layers. In first case, we define probability distribution of hidden layer conditioned over all the remaining layers $p(h|v, u, y)$. The restriction is that there is no intra-connection between the neurons in any layer while there is inter-connection between the neurons of different layers. The conditional probability of hidden layer can be calculated as:

$$p(h|v, u, y) = \text{ReLU} \left[\hat{b} + w^h \{ (v^T w^v) \circ (y^T w^y) \} \right]\tag{8}$$

For all inputs probability of hidden layer neurons is evaluated using rectified linear unit (ReLU) activation function.

Finally, we determine the probability of the visible layer conditioned on all the remaining layers such as history, hidden, and style layers $p(v|h, u, y)$. The visible layer probability is defined as:

$$p(v|h, u, y) = \text{ReLU} \left[\hat{a} + w^v \{ (h^T w^h) \circ (y^T w^y) \} \right]\tag{9}$$

2.4 Stacked FCRBM weights and biases learning rules

We adopt stochastic gradient decent for learning and updating rules because to overcome the problem of vanishing gradient. Moreover, the stochastic gradient decent on a large dataset converge faster and avoid overfitting as compared to mini-batch.

The weights of corresponding layers are updated as:

$$\begin{aligned}w_{t+1}^h &= w_t^h - \eta \frac{\partial E}{\partial w^h} \\ w_{t+1}^y &= w_t^y - \eta \frac{\partial E}{\partial w^y} \\ w_{t+1}^v &= w_t^v - \eta \frac{\partial E}{\partial w^v}\end{aligned}\tag{10}$$

The connections and biases are updated as follows:

$$\begin{aligned}
A_{t+1}^u &= A_t^u - \eta \frac{\partial E}{\partial A^u} \\
A_{t+1}^v &= A_t^v - \eta \frac{\partial E}{\partial A^v}
\end{aligned} \tag{11}$$

$$\begin{aligned}
A_{t+1}^y &= A_t^y - \eta \frac{\partial E}{\partial A^y} \\
B_{t+1}^u &= B_t^u - \eta \frac{\partial E}{\partial B^u} \\
B_{t+1}^h &= B_t^h - \eta \frac{\partial E}{\partial B^h}
\end{aligned} \tag{12}$$

$$\begin{aligned}
B_{t+1}^y &= B_t^y - \eta \frac{\partial E}{\partial B^y} \\
\hat{a}_{t+1} &= \hat{a}_t - \eta \frac{\partial E}{\partial v} \\
\hat{b}_{t+1} &= \hat{b}_t - \eta \frac{\partial E}{\partial h}
\end{aligned} \tag{13}$$

3 Proposed architecture

In this section, we introduce our proposed fast and accurate short term load forecasting model based on deep learning techniques such as stacked FCRBM and CRBM as shown in Figure 3. We adopt the modular strategy for short term load forecasting based on deep learning techniques, in which the output of each former module is fed into the later module. In short, our system model consists of three modules data processing and feature extraction module, deep learning-based training module, and deep learning-based forecasting module. The detailed description is as follows:

3.1 Data processing and feature extraction module

First the twenty zones historical data of US utility consisting of hourly load and weather data is taken from the Kaggle repository. This data is given as an input to the data processing and feature extraction module. The three data operations: cleansing, normalization, and structuring are performed on the received data. The cleansing operation is performed in order to replace the missing and defective values by the mean of the previous values. After cleansing, the data is normalized in order to reduce and eliminate the redundancy. Moreover, the data has large values, the normalization is performed to make the weighted sum within in the limits of the sigmoid function. At the end, we denormalize the data to achieve the desired load predictions. After cleansing and normalization, the data is structured in ascending or descending order. The desired features from the dataset is extracted by feature extraction process and finally the data is split into training and testing dataset. The training data have hourly load and weather data to train the deep learning techniques such as CRBM and stacked FCRBM. The testing data is used to evaluate the accuracy of the proposed deep learning techniques in terms of MAPE, NRMSE, and correlation coefficient. For the proper parameters tuning, we construct validation dataset.

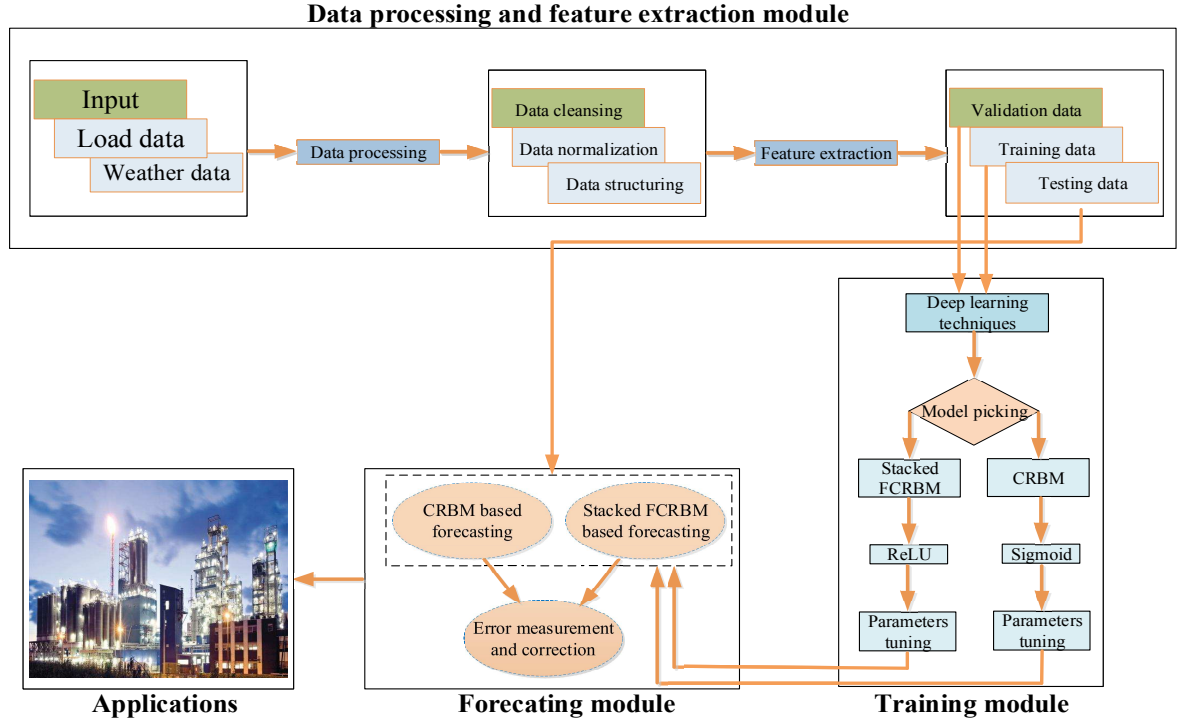


Fig. 3. Proposed architecture

3.2 Training module

Deep learning techniques such as CRBM and stacked FCRBM based training is the main part of this architecture. These techniques are trained with the training data, they learn non-linear relationship between demand load profile and historical observations. The output of the data processing and feature extraction module is given as an input to the training module. This module takes the training data and chose one of the models CRBM or stacked FCRBM for training. If the chosen model is stacked FCRBM, training module will train the stacked FCRBM using ReLU activation function because it overcomes the problems of vanishing gradient and curse of dimensionality. If the model chosen for training is CRBM training module will train it with sigmoidal activation function. In this way, the deep learning-based training module is enabled via learning to forecasts the future load.

3.3 Forecasting module

The output of the training module is fed into the forecasting module on the basis of trained deep learning models, the forecasting module forecasts the future load. The accuracy of the proposed deep learning techniques is evaluated in

terms of MAPE, NRMSE, and correlation coefficient using the testing data. The forecasted results are used for SG applications such as power generation planning, economic operation and unit commitment, power system maintenance and planning, load switching, power purchasing, demand side management, and contract evaluation.

4 Simulation results

In this section, simulation results of our proposed fast and accurate short term load forecasting model based on deep learning techniques such as stacked FCRBM and CRBM are presented. To validate accuracy of the proposed short term load forecasting model based on deep learning technique i.e., stacked FCRBM is compared with CRBM, ANN, and CNN in terms of performance metrics MAPE, NRMSE, and correlation coefficient. The detailed description is as follows: First, we define the forecast accuracy, in terms of NRMSE as:

$$RMSE = \sqrt{\frac{1}{\tau} \sum_{t=1}^{\tau} (R_t - F_t)^2} \quad (14)$$

$$NRMSE = \frac{RMSE}{\max(R_t - F_t)}$$

where τ is the number of steps forecasted in future, R_t represents the real value, and F_t is the forecasted value. Second, we present MAPE performance metric to get statistical significance for accuracy assessment. For minimum MAPE accuracy will be maximum and vice versa. The MAPE is calculated as:

$$MAPE = \left(\frac{1}{\tau} \sum_{t=1}^{\tau} \frac{|R_t - F_t|}{|R_t|} \right) * 100 \quad (15)$$

Finally, the correlation coefficient is implemented to check the accuracy that how close is the forecasted value to real value and is defined by the Equation 16. The correlation coefficient returns a value between -1 and 1. If the returned value is close to 1 the real and forecasted values are positively correlated, if the returned value is closed to -1 shows that the real and forecasted values are negatively correlated, and if zero is returned then real and forecasted values are not correlated.

$$r = \frac{E\{(R_t - \mu_R)(F_t - \mu_F)\}}{\sqrt{\sum (R_t - \mu_R)^2 * \sum (F_t - \mu_F)^2}} \quad (16)$$

where E is the expected value operator, μ_R is the mean of real values, and μ_F is the mean of forecasted value.

4.1 Historical load data

The historical data is taken from publicly available repository Kaggle of global energy forecasting competition 2012 [17]. The dataset consists of hourly load

(kW) of twenty zones of US utility and temperature of eleven stations. This historical dataset (load and weather) is of four years ranges from 1st hour of 1/1/2004 to 6th hour of 30/6/2008. The dataset is divided into training and testing data. The three years data is used to train the network and one year data is used to test the network. In summer during daytime there is a significant load increase of consumers as compared to night time. In winter during daytime there is a slight increase in the load as compared to the night. The electricity consumption from 2004 to 2008 maximum and minimum 540393 kWh and 149 kWh.

4.2 Learning curve

Learning curve describes the error rate across the number of epochs. It examines the difference between the training and testing data, when the gap between the training and testing data is more the forecasting results will be in accurate and vice versa. Generally, at some point where test error starts to increase as compared to training error, this simply means that overfitting is occurred. In such situation the model memorizes the given training data rather than learning and the forecasted results will be inaccurate. This problem is solved by several methods such as drop out and early stopping. However, we observed learning curves of both deep learning techniques stacked FCRBM and CRBM, we did not notice overfitting because test error decreases as training error does as shown in Figure 4. In this situation, the network is learning rather than memorizing and the forecasted results will be accurate.

4.3 Cumulative distribution function of errors

The NRMSE is expressed in terms of cumulative distribution function (CDF) as shown in Figure 5. The stacked FCRBM has 50% better CDF of NRMSE as compared to ANN, CNN, and CRBM because the stacked FCRBM has more computational capability. Stacked FCRBM based prediction is reliable, even if the load is uncertain with high error of predication. The results show that deep learning technique stacked FCRBM is robust and accurate as compared to CRBM, ANN, and CNN. Moreover, the stacked FCRBM would be the best choice for the consumers load forecasting as compared to CRBM because it has more computational power to capture the highly abstract data.

4.4 Deep learning based short term load forecasting

The short term load forecasting for one week time horizon with hourly resolution is described in Figure 6. We forecast load of one week in middle of each season. It is observed that FCRBM model closely follow the real curve as compared to the CRBM, ANN, and CNN. Moreover, the stacked FCRBM has more stable and accurate forecasted curve as compared to ANN, CNN, and CRBM. The observation numerical values in terms of performance metrics i.e., NRMSE,

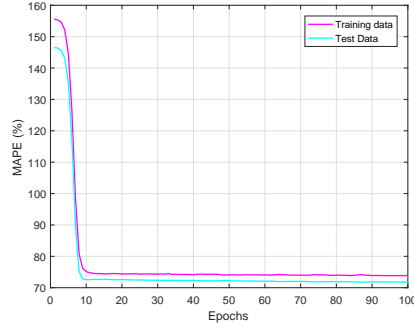


Fig. 4. Learning curve for both training and testing data

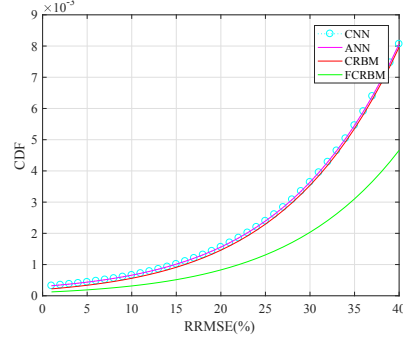


Fig. 5. CDF analysis in terms of error performance

MAPE, and correlation coefficient are listed in Table 1. It is ensured from the results listed in Table 1 that stacked FCRBM forecasted are more accurate as compared to other techniques in terms of NRMSE, MAPE, and correlation coefficient. Moreover, the adopted stacked FCRBM have 99.62% accuracy with affordable execution time and complexity. Also, we have observed that deep learning techniques stacked FCRBM and CRBM performance are dependent on the type, number, and size of dataset chosen. The performance of deep learning techniques improves as the size of the training data increases and vice versa.

Table 1. Accuracy analysis in terms of performance metrics

Techniques	MAPE (%)	NRMSE	correlation coefficient
ANN	3.500	2.048	0.4189
CNN	1.700	1.325	0.5300
CRBM	1.225	1.032	0.7287
Stacked FCRBM	0.377	0.6570	0.9987

5 Conclusion

A short term load forecasting model based on deep learning techniques i.e., stacked FCRBM and CRBM is proposed in this paper. The proposed model consists of data processing and feature extraction module, deep learning-based training module, deep learning-based forecasting module, and forecast weakly load profile on the basis of past energy consumption. We found that the stacked FCRBM and CRBM are effective to learn from past energy consumption and exhibit better performance compared to literature forecasting models. The performance evaluation of the proposed model is performed in terms MAPE, NRMSE,

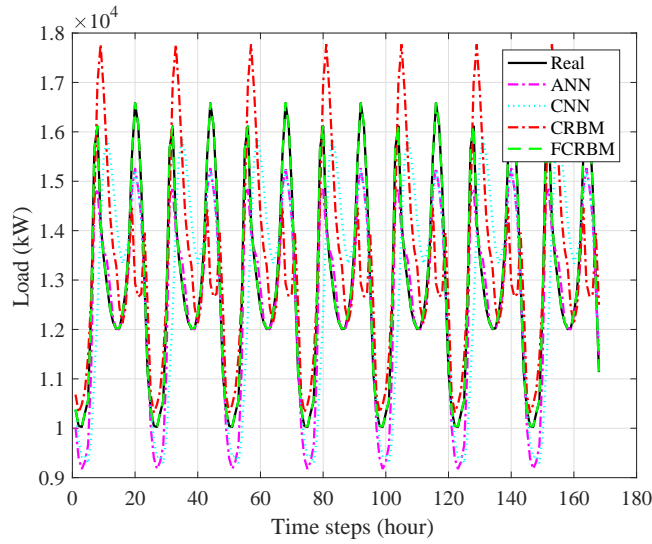


Fig. 6. Short term energy forecasting based on deep learning technique stacked FCRBM

and correlation coefficient. Simulation results demonstrate that stacked FCRBM and CRBM are accurate and robust as compared to ANN and CNN. Moreover, the adopted stacked FCRBM achieved 99.62% accuracy with affordable execution time and complexity. In future, we will work in medium term forecasting and long term forecasting.

References

1. <https://www.eia.gov/tools/faqs/faq.php?id=86&t=1> (accessed on 19 April 2018).
2. Hafeez, Ghulam, Nadeem Javaid, Sohail Iqbal, and Farman Ali Khan. "Optimal Residential Load Scheduling Under Utility and Rooftop Photovoltaic Units." *Energies* 11, no. 3 (2018): 611.
3. Hafeez, Ghulam, Nadeem Javaid, Saman Zahoor, Itrat Fatima, and Zahoor Ali Khan. "Energy efficient integration of renewable energy sources in smart grid." In *International Conference on Emerging Internetworking, Data & Web Technologies*, pp. 553-562. Springer, Cham, 2017.
4. Marino, Daniel L., Kasun Amarasinghe, and Milos Manic. "Building energy load forecasting using deep neural networks." In *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*, pp. 7046-7051. IEEE, 2016.
5. Zeng, Nianyin, Hong Zhang, Weibo Liu, Jinling Liang, and Fuad E. Alsaadi. "A switching delayed PSO optimized extreme learning machine for short-term load forecasting." *Neurocomputing* 240 (2017): 175-182.
6. Cecati, Carlo, Janusz Kolbusz, Pawe Rycki, Pierluigi Siano, and Bogdan M. Wilamowski. "A novel RBF training algorithm for short-term electric load forecasting and comparative studies." *IEEE Transactions on industrial Electronics* 62, no. 10 (2015): 6519-6529.

7. Dedinec, Aleksandra, Sonja Filiposka, Aleksandar Dedinec, and Ljupco Kocarev. "Deep belief network based electricity load forecasting: An analysis of Macedonian case." *Energy* 115 (2016): 1688-1700.
8. Fan, Cheng, Fu Xiao, and Yang Zhao. "A short-term building cooling load prediction method using deep learning algorithms." *Applied energy* 195 (2017): 222-233.
9. Amjady, Nima, Farshid Keynia, and Hamidreza Zareipour. "Short-term load forecast of microgrids by a new bilevel prediction strategy." *IEEE Transactions on smart grid* 1, no. 3 (2010): 286-294.
10. Mocanu, Elena, Phuong H. Nguyen, Madeleine Gibescu, and Wil L. Kling. "Deep learning for estimating building energy consumption." *Sustainable Energy, Grids and Networks* 6 (2016): 91-99.
11. Ryu, Seunghyoung, Jaekoo Noh, and Hongseok Kim. "Deep neural network based demand side short term load forecasting." *Energies* 10, no. 1 (2016): 3.
12. Ahmad, Ashfaq, Nadeem Javaid, Mohsen Guizani, Nabil Alrajeh, and Zahoor Ali Khan. "An accurate and fast converging short-term load forecasting model for industrial applications in a smart grid." *IEEE Transactions on Industrial Informatics* 13, no. 5 (2017): 2587-2596.
13. V. Mnih, H. Larochelle, G. Hinton, Conditional restricted Boltzmann machines for structured output prediction, in: *Proceedings of the International Conference on Uncertainty in Artificial Intelligence* (2011).
14. Hinton, Geoffrey E. "A practical guide to training restricted Boltzmann machines." In *Neural networks: Tricks of the trade*, pp. 599-619. Springer, Berlin, Heidelberg, 2012.
15. G.W. Taylor, G.E. Hinton, S.T. Roweis, Two distributed-state models for generating high-dimensional time series, *J. Mach. Learn. Res.* 12 (2011) 10251068.
16. Mocanu, Decebal Constantin, Haitham Bou Ammar, Dietwig Lowet, Kurt Driessens, Antonio Liotta, Gerhard Weiss, and Karl Tuyls. "Factored four way conditional restricted boltzmann machines for activity recognition." *Pattern Recognition Letters* 66 (2015): 100-108.
17. <https://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting> (accessed on 26 March 2018).