IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 3, Ver. IV (May - June 2017), PP 33-40
www.iosrjournals.org

Comparison between Adaptive and Conventional RBFNN Based Approach for Short-Term Load Forecasting

Eyad Almaita[1]
[1](*Electrical Power and Mechatronics Engineering Department, Engineering/ Tafila Technical University, Jordan)*

Abstract: *In this paper, a comparison between novel adaptive Radial Basis Function Neural Networks (RBFNN) algorithm and conventional RBFNN is conducted. Both algorithms are used to forecast electrical load demand in Jordan. The Same forecasting features are used in both algorithms. Most of the forecasting models need to be adjusted after a period of time, because the change in the system parameters. The data used in this paper is real data measured by National Electrical Power co. (Jordan). The data is divided into two sets. Set for a training and the other for testing. The results illustrated that the adaptive RBFNN model outperformed conventional RBFNN. The proposed adaptive RBFNN model can enhance the reliability of the conventional RBFNN after embedding the network in the system. This is achieved by introducing an adaptive algorithm that allows the change of the weights of the RBFNN after the training process is completed, which will eliminate the need to retrain the RBFNN model again.*

Keywords: *Load Forecasting, Neural Network, Radial Basis Function, Short-Term, adaptive.*

I. Introduction

T Short-Term Load Forecasting (STLF) is a crucial part in power system efficient operation, this includes economic dispatch, fuel scheduling and unit maintenance [1]. With a small country like Jordan, that live almost with no natural resources, his imports from energy represent around 95% of his consumption [2] , around every ten years has a waves of refugees come in, and with the big increase in energy prices, STLF represents really an essential problem needs to be addressed.

Different techniques have been used to tackle the short-term load forecasting problem. These techniques can be classified into two models: (i) Linear models such as ARX, ARMA, etc.[3][4] . (ii) Nonlinear models such as Artificial Neural Networks (ANN) [5], Support Vector Machine (SVM) [6], and Fuzzy logic [7]. Because of the nonlinear nature of the short-term forecasting problem there was a trend in recent years to utilize the power of nonlinear models to solve this problem [8]-[14]. ANNs have great capabilities in dealing with nonlinear prediction problems. They are nonlinear in nature, can deal with huge numbers of variables, and can minimize the effect of noisy and uncertain data [8]-[14].

ANNs architectures that have been used in LTLF problem can be classified into two architectures: (i) hybrid ANN architectures and (ii) pure ANN architectures. Pure ANNs uses either the famous Backpropagation Neural Network (BPNN) model [12] or the Radial Basis Function Neural Network model [13], on the other hand, hybrid models try to combine the power of ANNs and other algorithms such as fuzzy logic, support vector machine (SVM) , Wavelet, and grey model [15].

A lot of the previous models (Hybrid and pure) use many networks to build a single STFL model, some of them uses many algorithms in the same model, which increases the model complexity, another issue regarding some of previous models, is the large number of the hidden neurons [7]-[14]. A simple and effective STFL model using RBFNN was introduced in [16].

In this paper, a comparison between conventional RBFNN introduced in [15] and Adaptive RBFNN is investigated. The hour by hour load demand for the Jordanian power grid is forecasted. The Forecasting models for both conventional and adaptive RBFNN are based on a limited historical data and small number of factors. The Structure of both algorithms and the forecasting models are illustrated. Also, a comparison in the performance of both algorithms is carried out.

II. Conventional RBFNN Algorithm

A. Structure of RBFNN

The RBFNN structure consists of three main different layers as shown in Fig. 1; one input layer (source nodes with inputs $I_1, I_2,.., I_N$), one hidden layer has K neurons, and one output layer (with outputs $y_1, y_2,.., y_m$). The input-output mapping consists of two different transformations; nonlinear transformation from the input layer to the hidden layer and linear transformation from hidden to the output layer. The connections between the input and hidden layers are called centers and the connections between the hidden and output layers are called weights [17]–[18].

The most common radial basis function used in RBFNN is given by

$$\phi_i(x) = \exp\left[-\frac{(x-c_i)^T(x-c_i)}{2\sigma_i^2}\right], \qquad \text{i=1,2,...,K} \qquad (1)$$

This is a Gaussian basis function with φ_i as the output of the i^{th} hidden neoron , x is the input vector data sample (I_1, I_2,…,I_N) (could be training, actual, or test data), c_i is centers vector of the i^{th} hidden neuron $(c_{i1}, c_{i2}, .., c_{iN})$, σ_i is the normalization factor, and $(x\text{-}c_i)^T(x\text{-}c_i)$ is the square of the vector $(x\text{-}c_i)$ [16]–[17]. The i^{th} output node y_i is a linear weighted summation of the outputs of the hidden layer and is given by

$$y_i = w_i{}^T \emptyset(x), \qquad i=1, 2,...., m \qquad (2)$$

where w_i is the weight vector of the output node and $\Phi(x)$ is the vector of the outputs from the hidden layer (augmented with an additional bias which assumes a value of *1*).

B. Training Algorithm of RBFNN

The block diagram shown in Fig. 2 illustrates one of the RBFNN training processes called *hybrid learning* process [18]. The *hybrid learning* process has two different stages; (i) finding suitable locations for the radial basis functions centers of the hidden neurons [18], [19] and (ii) finding the weights between the hidden and output layers. In the first stage the K-means [18], [19] clustering algorithm is used to locate the centers in the input data space regions where a significant data are present (shown as I in Fig. 2). In the second stage (shown as II in Fig. 2) the weight matrix between the hidden and the output layers are found by linear matrix inversion algorithm based on the least-square solution, which minimizes the sum-squared error function [20].
The weights matrix w is calculated by

$$w = A^{-1}\Phi^T D \qquad (3)$$

where D is the desired output vector for l training data samples set and given by :

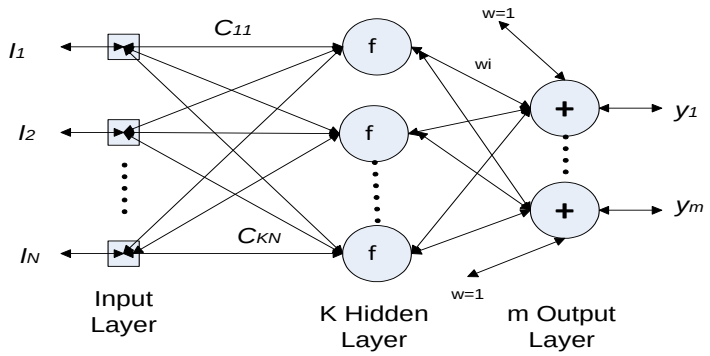

Fig. 1 Structure of Conventional RBFNN Network

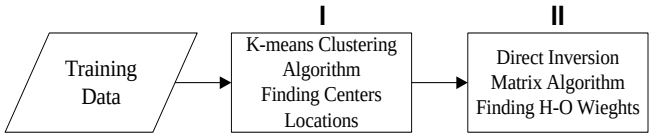

Fig. 2 Block Diagram for the RBFNN Hybrid Learning Process

$$D = \begin{bmatrix} d(x_1) \\ d(x_2) \\ . \\ d(x_j) \\ . \\ d(x_l) \end{bmatrix} \qquad (4)$$

where $d(x_j)$ describes the output vector corresponding to the j^{th} training data samples vector (x_j). Φ is a matrix where each element $\varphi_i(x_j)$, is a scalar value and represents the output of the i^{th} hidden neuron for the j^{th} training data samples vector (x_j). The Φ matrix for l training data samples is given by

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & .... & \phi_K(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & ... & \phi_K(x_2) \\ . & . & & . \\ & & ... & \\ . & . & & . \\ \phi_1(x_l) & \phi_2(x_l) & ... & \phi_K(x_l) \end{bmatrix} \qquad (5)$$

A^{-1}, the variance matrix and given by

$$A^{-1} = \left[\Phi^T \Phi\right]^{-1} \qquad (6)$$

One of the advantages of this method compare to other training algorithms is that it does not need iterations in the training phase; what it needs is the matrix inversion shown in (6), which needs negligible time to be calculated.

III. Novel Adaptive RBFNN Algorithm

One of the major disadvantages of the feed forward neural networks (BPNN and conventional RBFNN) techniques is that; the obtained parameters do not changed once the training process is completed. In the presence of the noise, these fixed parameters can degrade the performance of the neural networks. The main objective of the adaptive RBFNN algorithm is to enhance the reliability of the conventional RBFNN after embedding the network in the system. This can be achieved by introducing an adaptive algorithm for RBFNN structure that allows the change of the weights of RBFNN after the training process is completed. As shown in II, the RBFNN adjustable parameters that will affect the output is the centers and the weights. This algorithm assumes that the noise present in the system can be mitigated only by adjusting the weights between the hidden and the output layers, without the need of adjusting the values of the centers between the input and hidden layers.

Fig.3 shows the general structure of the adaptive RBFNN algorithm. It has the same conventional RBFNN structure regarding input layer, hidden layer, and output layer. But it has two extra components; (i) Summation component, which is located after the outputs of the RBFNN. The goal of this component is to calculate the error signal between the estimated outputs y and the reference (actual) signal (R). (ii) Weights updating component. The goal of this component is to adjust the weights in order to reduce the error signal. In the absence of the noise δ(k) in the input side, the summation of the outputs of the RBFNN model is equal to the reference signal R(k). In this case the error E(k) equal to zero and no change in the RBFNN weights.

$$E(k)=R(k)-\{y\_1(k)+ y\_2(k)+\cdots+y\_m(k)\} \qquad (7)$$

In the presence of noise in the input side, the jth output node of the RBFNN will be affected by this noise as

$$y\_j(k)=y\_oj(k)+ \delta\_j(k) \qquad (8)$$

where y_oj (k) is the jth output node without noise and δ_j (k) is the added noise error to the jth output node.
In order to mitigate the effect of the noise in the performance of the RBFNN, the error E(k) is used to update the weights vectors based on the least-mean- square-error algorithm [7] as:

$$w\_1new=w\_1old+ \eta\_1 \emptyset(k)E(k) \qquad (9)$$

$$\vdots$$

$$w\_mnew=w\_mold+ \eta\_m \emptyset(k)E(k) \qquad (10)$$

where ηj is the regulation factor for the jth output node.
The weights updating will continue until the error E(k) become zero again.

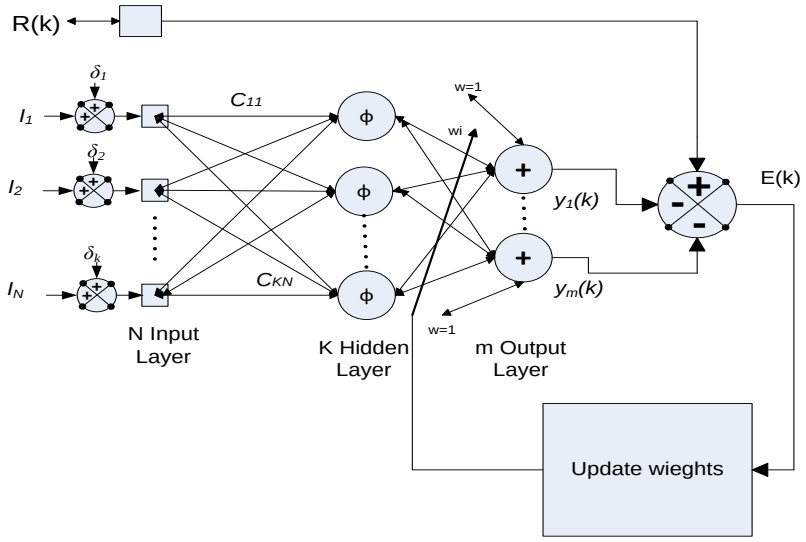

Fig. 3 Structure of the Adaptive RBFNN Algorithm

The above algorithm has several advantages including the following; (i) It has a fast convergence time because it adjusts only the weights between the hidden and output layers, which is a linear relationship. Therefore, fast convergence can be achieved. (ii) The updating process could be initiated based on threshold value for E(k) (different from zero), which gives the flexibility to the algorithm and saves excessive computations. (iii) This algorithm has greater capabilities compare to the popular neural linear adaptive algorithm (ADALINE) because, the RBFNN structure can be used to realize linear and nonlinear functions.

IV. Methodology

A. Forecasting model structures

In this section, the forecasting model structures using conventional and adaptive RBFNN is illustrated in Fig.4 and Fig.5 respectively. Both models depend on the same features. The forecasted next day hourly load, for both models, will be estimated based only on six features. These six input features are: (1) 24-hour (previous day) delayed hourly load, this input is to cover 24-hour cyclic pattern of the load, (2) 168-hour (previous week) delayed hourly load, this input is to cover weekly cyclic pattern of the load, (3) hourly forecasted temperature, this input is intended to compensate for the effect of the temperature on the hourly load, (4) hour of the day, (5) week day number, where Sunday is given *1* and Saturday is given *7*, and (6) the type of the day. The feature type of the day is concerned about addressing holidays and Ramadan month, so if the day is holiday or Ramadan day the value of this feature is *1* otherwise it is *0*. The difference between the two models is that the furcating model using adaptive RBFNN will compare the forecasted (expected) hourly load, for a given hour, with the actual load at that hour (when it becomes available). The difference (error) between the forecasted load and the actual load will be used to update the adaptive RBFNN model based on methodology that was illustrated in III.

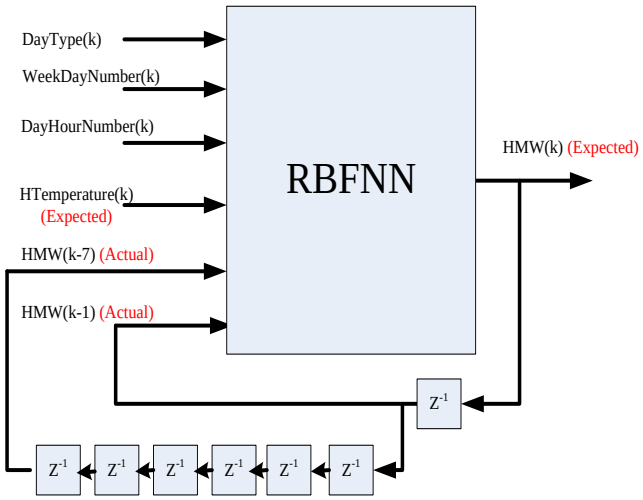

Fig. 4 Forecasting Model Structure for the RBFNN

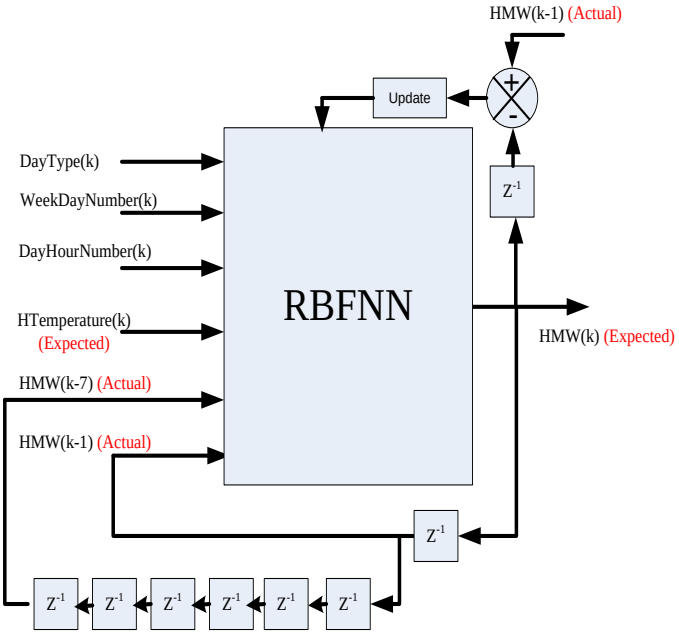

Fig. 5 Forecasting Model Structure for the Adaptive RBFNN

B. Data preparation

The data used in this study is a real data measured in the Jordanian electrical power grid. The data from Jan/1/2012 to April/30/2013 is used to train the RBFNN model. The test data is from *May/1/2013* to *Sep./30/2013*. In order to make the inputs of the RBFNN model homogenous, all the input data is normalized between *0* and *1* as:

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (11)$$

- Where X_n, X is the normalized and non-normalized feature value, respectively. X_{max}, X_{min} is the maximum and the minimum value for a given feature, respectively.

C. Model effectiveness

The effectiveness of the model is measured by using the Mean Absolute Percentage Error (MAPE) which is calculated as:

$$MAPE = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - E_i|}{A_i}\right] \qquad (12)$$

where
A_i : is the actual value
E_i : is the Expected (Forecasted) value
n: is the number of the sample

V. Results

The used data in the forecasting model is an actual data and it was provided by National Electrical Power Company (NEPCO) / Jordan. The MATLAB® software is used to build the model. The value of σ in the RBFNN models depends on the input training data. This value is obtained by running the simulation several times and selecting the value that minimizes the RBFNN network error. The number of hidden neuron is small comparing to the others models. The number of hidden neurons used in this model is *11* neurons.

One of the main features of any short-term forecasting model is the addressing of the daily and weekly periodicity of the load. Fig.6 shows two weeks load. It is clear from this figure that there are two periodic patterns. The first pattern is the daily load, where in each day there are two peaks. One peak happen in during the day light and the other is in the nigh. The second pattern is the weekly periodicity of the load. It is clear the every seven days the load pattern repeat itself. Another feature of this model is the day type. This feature tackles the difference in the daily load profile between working days and holidays. Some of these holidays do not have constant dates like Christmas. Actually their dates are moving, because they depend on lunar calendar. These moving holidays can cause errors in any forecasting models if not taken in consideration. Fig.7 shows a profile

load for the same day but one of them was holiday. The load profile for the holiday is much lower than the normal day. Fig.8 shows three curves at the same figure. These curves represent the actual load hourly loads for one week of the test data (dotted), the forecasted hourly load based on conventional RBFNN (dashed), and the forecasted hourly load based on adaptive RBFNN (solid). The MAPE for the test data based on the adaptive RBFNN ***(0.0128)*** is around one fourth of the MAPE based on conventional RBFNN ***(0.04)***. Also the MAPE for individual hours is shown in Fig.9. It can been seen that the adaptive RBFNN performance is much better than the conventional RBFNN in individual hour MAPE. And this is very important for forecasting the peak load for that day.

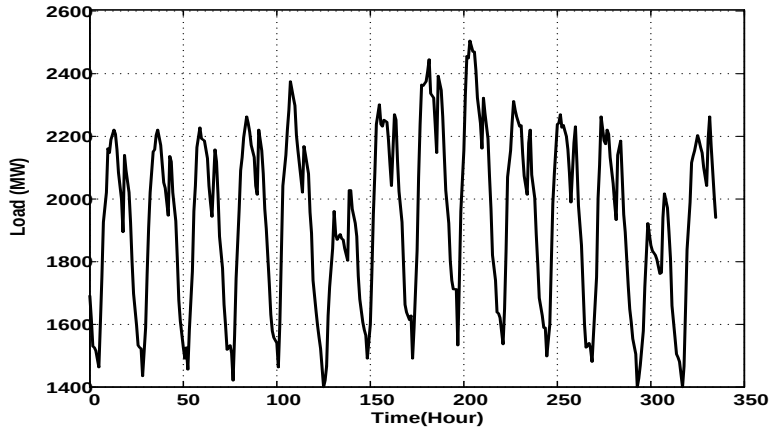

Fig. 6 Daily and weekly load profile

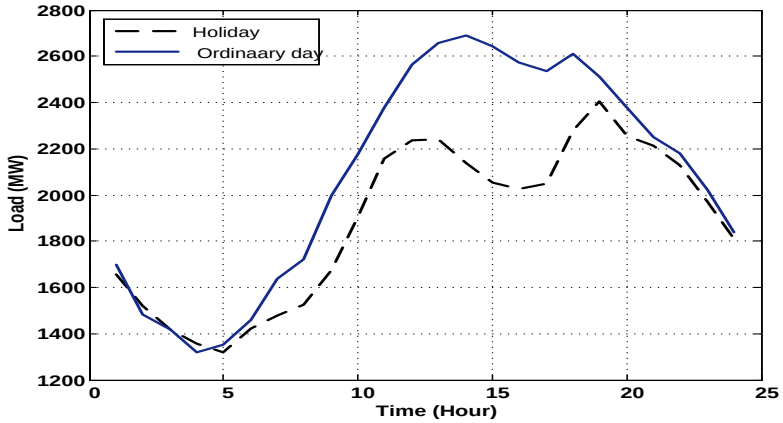

Fig. 7 Holiday (dashed) Vs. normal day (solid) load profile

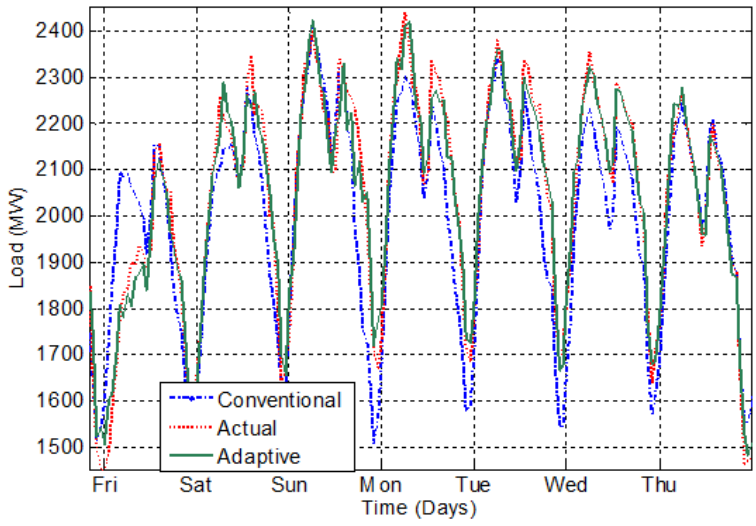

Fig. 8 Next -24 hour load forecasting (seven days period), Actual (solid) and forecasted (dashed)

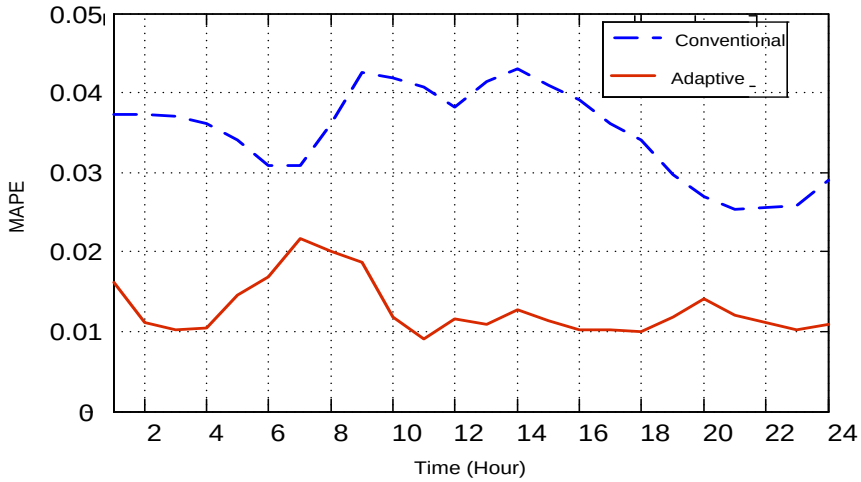

Fig. 9 Hourly MAPE for adaptive RBFNN model (solid) and conventional RBFNN model (dashed)

VI. Conclusion

In this paper, a comparison between novel adaptive Radial Basis Function Neural Networks (RBFNN) algorithm and conventional RBFNN was conducted. Both algorithms are used to forecast the hour by hour electrical load demand in Jordan. Both algorithms were under the same conditions; the same training, the same test data and the same forecasting parameters. The adopted forecasting models for both algorithms utilized six simple features. These features are; the load in the previous day, the load in the same day in the previous week, the temperature in the same hour, the hour number, the day number, and the day type. Most of the forecasting models need to be adjusted after a period of time, because the change in the system parameters. The results illustrated that the adaptive RBFNN model outperformed conventional RBFNN. The proposed adaptive RBFNN model can be enhance the reliability of the conventional RBFNN after embedding the network in the system. This was achieved by introducing an adaptive algorithm that allows the change of the weights of the RBFNN after the training process is completed, which will eliminate the need to retrain the RBFNN model again. This feature make adaptive RBFNN is more suitable for forecasting models and this algorithm can be extended to the other applications such as control systems and signal processing.

Acknowledgements

This research was supported by National Electrical Power Co. (NEPCO) in Jordan.

References

[1] Abderrezak Laouafi, Mourad Mordjaoui, Farida Laouafi and Taqiy Eddine Boukelia, "Daily peak electricity demand forecasting based on an adaptive hybrid two-stage methodology", *International Journal of Electrical Power & Energy Systems*, vol. 77, pp. 136, 2016, ISSN 01420615.

[2] http://www.memr.gov.jo

[3] S. Kumar, S. Mishra and S. Gupta, "Short Term Load Forecasting Using ANN and Multiple Linear Regression," *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, Ghaziabad, India, 2016, pp. 184-186.

[4] T. Haida and S. Muto, "Regression based peak load forecasting using a transformation technique," IEEE Trans. Power Syst., vol. 9, no. 4, pp. 1788–1794, 1994.

[5] Taylor, J.W.; Buizza, R., "Neural network load forecasting with weather ensemble predictions," *Power Systems, IEEE Transactions on*, vol.17, no.3, pp.626,632, Aug 2002.

[6] B. J. Chen, M. W. Chang, and C. J. Lin, "Load Forecasting using Support Vector Machines: A Study on EUNITE competition 2001," IEEE Transactions on Power Systems, Vol. 19, No. 4, pp. 1821–1830, November 2004..

[7] A. C. Luna; N. L. Diaz Aldana; M. Graells; J. C. Vasquez; J. M. Guerrero, "Mixed-Integer-Linear-Programming based Energy Management System for Hybrid PV-wind-battery Microgrids: Modeling, Design and Experimental Verification," in *IEEE Transactions on Power Electronics* , vol.PP, no.99, pp.1-1.

[8] Xiaorong Sun; Luh, P.B.; Michel, L.D.; Corbo, S.; Cheung, K.W.; Wei Guan; Chung, K., "An efficient approach for short-term substation load forecasting," *Power and Energy Society General Meeting (PES), 2013 IEEE* , vol., no., pp.1,5, 21-25 July 2013

[9] Ramos, S.; Soares, J.; Vale, Z.; Ramos, S., "Short-term load forecasting based on load profiling," *Power and Energy Society General Meeting (PES), 2013 IEEE* , vol., no., pp.1,5, 21-25 July 2013

[10] Zeng Linsuo; Li Yanling, "A Method for Power System Short-Term Load Forecasting Based on Radial Basis Function Neural Network," *Intelligent Systems Design and Engineering Applications, 2013 Fourth International Conference on* , vol., no., pp.12,14, 6-7 Nov. 2013.

[11] Raza, M.Q.; Baharudin, Z.; Nallagownden, P.; Badar-Ul-Islam, "A comparative analysis of PSO and LM based NN short term load forecast with exogenous variables for smart power generation," Intelligent and Advanced Systems (ICIAS), 2014 5th International Conference on , vol., no., pp.1,6, 3-5 June 2014.

[12] Schachter, J.; Mancarella, P., "A short-term load forecasting model for demand response applications," *European Energy Market (EEM), 2014 11th International Conference on the* , vol., no., pp.1,5, 28-30 May 2014.

[13] Xia, Changhao, Jian Wang, and Karen McMenemy. "Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks." *International Journal of Electrical Power & Energy Systems* 32.7 (2010): 743-750.
[14] Wilamowski, B.M.; Cecati, C.; Kolbusz, J.; Rozycki, P.; Siano, P., "A Novel RBF Training Algorithm for Short-term Electric Load Forecasting and Comparative Studies," *Industrial Electronics, IEEE Transactions on* , vol.PP, no.99, pp.1,1,10.1109/TIE.2015.2424399.
[15] Raza, M.Q.; Baharudin, Z., "A review on short term load forecasting using hybrid neural network techniques," Power and Energy (PECon), 2012 IEEE International Conference on , vol., no., pp.846,851, 2-5 Dec. 2012.
[16] Almaita, E. "Next-24 Hour Load Forecasting of Jordanian Power Grid using Radial Basis Function Neural Networks". World Academy of Science, Engineering and Technology, International Science Index, Electrical and Computer Engineering, (2014), 1(10), 280.
[17] S. S. Haykin 1931-, Neural Networks : A Comprehensive Foundation /. Upper Saddle River, N.J. : Prentice Hall, c1999.
[18] N. K. Kasabov, Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering /. Cambridge, Mass. : MIT Press, c1996.
[19] J. Moody, "Fast learning in networks of locally-tuned processing units," Neural Comput., vol. 1, pp. 281, 1989. R. Yousef, "*Training radial basis function networks using reduced sets as center points,*" International Journal of Information Technology, vol. 2, pp. 21, 2005.