International Journal of Electrical and Electronic Engineering & Telecommunications

ISSN 2319 – 2518 www.ijeetc.com
Vol. 3, No. 3, July 2014

Research Paper

SHORT TERM FORECASTING OF MARKET CLEARING PRICE IN INDIAN ENERGY EXCHANGE USING ANN-PSO MODEL

Smitha Elsa Peter[1]* and I Jacob Raglend[2]

*Corresponding Author: Smitha Elsa Peter, ✉ smitha_peter@yahoo.com

This paper proposes a new ANN-ANN-PSO model solely for forecasting Market Clearing Price (MCP) in Indian Energy Exchange (IEX). IEX is one of the India's electricity power trading platforms where more than 2600 participants across utilities from 27 states, 5 Union Territories, more than 500 private generators and more than 2300 open access consumers are doing business with IEX. In such a competitive electricity market, generating companies (Gencos) assess bidding strategies to maximize their profits. Gencos have to make an intelligent decision to bid the MCP beforehand based on limited information available. The accuracy in the forecasted MCP will aid Gencos in enhancing the chances of winning bids, since the MCP depends on the bidding participant behavior of both seller and buyer in the market. Thus, a most favorable bidding strategy is a challenging task for GenCos. This paper uses a similar-day approach for forecasting the MCP. The recent available historical data from January 1, 2014 to March 16, 2014 is used in this research work. This paper also investigates the performance related issues of the proposed ANN-ANN-PSO model with respect to ANN and ANN-PSO models.

Keywords: Error variance, Market clearing price, Mean absolute percentage error, Neural network, Particle swarm optimization

INTRODUCTION

Electricity price (Conejo *et al.*, 2004) forecasting is an important function in an electricity market. There are various markets around the world such as California Independent Systems Operator (CASIO), Midwest ISO (MISO), New England (ISO-NE), New York ISO (NYISO), PJM-RTO (regional transmission organisation), National Electricity Market (NEM) Australia, Ontario Electricity market-Canada, etc. However, each market has its own method of operation and regulations. Therefore, it is necessary to develop an accurate MCP forecasting model

[1] Department of ECE, PRIST University, Thanjavur 613403, India.
[2] Department of EEE, Noorul Islam University, Tamil Nadu, India.

relevant to a particular electricity market. It should also be noted that the accuracy in forecasting MCP depends on the intrinsic and extrinsic factors that depend on each market. A precise price forecasting helps suppliers to set up bidding strategies, make investment decisions and be cautious against risks. Conversely, consumers can use price forecasting to exploit appropriate power purchasing strategies for maximum utility utilization. Therefore, MCP is varying price signals that mostly depend on the dynamic behaviour of buyer and seller, analogous to the demand and supply in the market, respectively. When electricity MCP is determined, every supplier whose offering price is below or equal to the electricity MCP will be picked up by the Independent System Operator (ISO) to supply electricity at that hour. They will be paid at the same price, the electricity MCP, not the price they offered. The reason for this is to keep fairness of the market and to avoid market manipulation. The accuracy of the forecast depends on the availability of the data and further depends on other influential price drivers such as volatility in fuel price, load uncertainty, fluctuations in hydroelectricity production, generation uncertainties, transmission congestion, behaviour of market participants, etc.

Several methods have been proposed by researchers for short-term price forecasting. Among these methods, two extensively used approaches are time series models and Artificial Neural Networks (ANNs). Models such as WT model (Aggarwal *et al.*, 2008) and k-GIGARCH model (Diongue *et al.*, 2004) have been proposed for this purpose. Similarily, Areekul *et al.* (2010) had proposed a hybrid methodology that combined both Autoregressive Integrated Moving Average (ARIMA) and Artificial Neural Network (ANN) models for predicting short-term electricity prices and it was validated by using the data of Australian national electricity market, New South Wales from the year 2006. Khashei *et al.* (2008) proposed a new hybrid model that overcame the limitations in both the ANNs and fuzzy regression models to yield more accurate results with incomplete data sets. Singhal and Swarup (2011) developed a three-layer Back Propagation (BP) neural network method to forecast the Market-Clearing Prices (MCPs) for day-ahead energy markets. The forecasting results showed that the model was efficient for days with normal trend but however for days with price spikes the model displayed a gradual degradation on performance.

Pindoriya *et al.* (2008) proposed an Adaptive Mexican hat Wavelet Neural Network (AWNN) for Short-Term Price Forecasting (STPF) in the electricity markets. The model was implemented to predict the day-ahead prediction of MCP of Spain market, a duopoly market with a dominant player, and the Locational Marginal Price (LMP) forecasting in PJM electricity market was also considered. The forecasted results clearly showed that AWNN had good prediction properties compared to other forecasting techniques, such as wavelet-ARIMA, Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) neural networks as well as Fuzzy Neural Network (FNN). Catalao *et al.* (2011) had proposed a novel hybrid approach, combining wavelet transform, particle swarm optimization, and adaptive-network-based fuzzy inference system for short-term electricity prices forecasting in a competitive market. The results from the case study were based on the electricity market of mainland Spain. Pany and Ghoshal (2013) proposed a particle swarm optimization based Local Linear Wavelet Neural Networks (LLWNN) model that smoothly bases function of hidden layer neurons according to training data set, maps the input-output space by adapting the shape of wavelet, was examined for electricity price prediction of the Ontario electricity market.

Although the main advantage of ANN is their non-linear modelling capability, they cannot capture the characteristics of high volatility in electricity price. In general, the performance of the ANN forecasting model will be more accurate if the training of weights in the neural network is carried out in the best possible manner. However, it is found from the literature, while training, the weights gets caught in the local minima and thereby the training of the neural network leads to premature convergence. In order to overcome this limitation, instead of conventional gradient based training of weights, heuristic approaches are used in optimization of weights during neural network training. Hybrid-ANN models, which combine heuristic search algorithms such as genetic algorithms, Particle Swarm Optimisation (PSO), artificial bee colony algorithms, etc., for updating the weights, show some better performance. Among these optimization algorithms, particle swarm optimization is regarded as a promising method in several engineering applications as is evident in the literature. Therefore, the proposed research work develops a novel ANN based training scheme using PSO for forecasting the MCP.

PROPOSED WORK

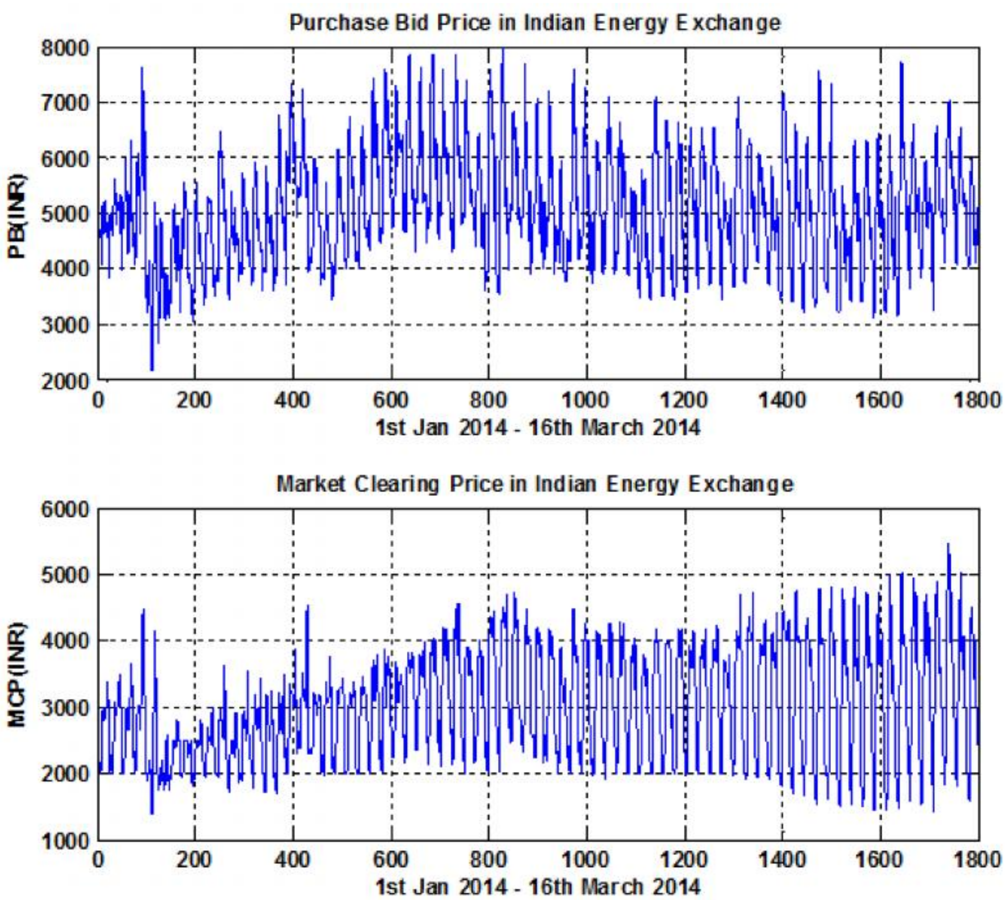
The proposed research work focuses to develop a suitable forecast engine solely to Indian Energy Exchange (IEX). There are two main reasons to carry out this proposed work. Firstly, not so many literatures are available for the forecast of MCP in the IEX. Secondly, there is a possibility in improving the performance of ANN-PSO during training of weights with PSO. The proposed model involves two sequential phases. In the first phase, ANN is used to train historical patterns. More number of trials is carried out and the final weights that give the least training error are stored. In the second phase the stored weights that are obtained from various trials are used as the initial population for the ANN-PSO model. The random initialization of weights in the ANN-PSO is avoided; thereby the chances of better training of the patterns are explored by the PSO in training the weights of the ANN. It should be noted that the ANN model that is used in the proposed investigation is a feed forward back propagation neural network.

INDIAN ENERGY EXCHANGE AND ITS DATABASE

The operational activities of IEX is to manage power portfolio in the most competitive and reliable way. Day-Ahead and Term-Ahead market is followed in the IEX. Day-Ahead-Market (DAM) is a physical electricity trading market for deliveries for any/some/all 15 minute time blocks in 24 hours of next day starting from midnight. The prices and quantum of electricity to be traded are determined through a double sided closed auction bidding process. Term-Ahead-Market (TAM) provides a range of products allowing participants to buy/sell electricity for contracts beyond day-ahead market, besides intraday contracts [www.iexindia.com]. The proposed work concentrates in forecasting the hourly Week-Ahead Market Clearing price which is the part of TAM using a similar day approach using Feed Forward Back Propagation Neural Network (FFBPNN). The activities of the consumers are found to be similar on the same week days. So, in this case study, MCP of similar days is correlated for training the historical MCP data. For example, the MCP profile on Monday of the previous week is correlated to Monday of the present week. So when a test input is fed into the forecast model, a week-ahead consumption profile is forecasted.

The historical data that are available in the IEX website is the hourly Purchase Bid (MW), Sell Bid (MW), Market Clearing Volume (MW), Cleared Volume (MW) and Market Clearing Price (INR). Market Clearing Volume (MCV) is carried out before transmission congestion whereas Cleared Volume (CV) is carried out after transmission congestion. However, in Peter *et al.* (2014), when using ANN forecast model, it is observed that the Purchase Bid data is closely related to MCP among the other available historical data. Therefore, in this work, PB and MCP are only used in the

Figure 1: Market Snapshot of Historical Data (1st 75 Days of the Year 2014)



proposed ANN-ANN-PSO forecast model. The market snapshot data for PB and MCP for the first 75 days is presented in the Figure 1. The total number of samples for 70 days is 1680. The training data, validation data and the testing data for the FFBNN is considered only from the 1680 samples. The source and the target training data for FFBNN training is taken from January 1, 2014 to February 19, 2014, and from January 8, 2014 to February 26, 2014, respectively. The validation data is taken from February 12, 2014 to February 19, 2014 and is compared with the actual data from February 20, 2014 to February 26, 2014. The testing or verification data is taken from February 27, 2014 to March 5, 2014, and is compared with the actual data from March 6, 2014 to March 12, 2014. It should be noted that the testing data is not used in the training set, whereas the validation data is used in the training set. Validation is carried out while training to check that the network do not over train, thereby the forecast accuracy will not deteriorate.

PROPOSED METHODOLOGY

The proposed methodology involves two phases (Phase A and Phase B) of operation. In the first phase, the training of feed forward back propagation neural network is carried out in the batch mode using the conventional gradient based neural network. Before the start of the training, the weights are initialized randomly and the training is continued till the training error gets minimized and no further improvement is possible. Then the final weights obtained for the network is recorded. This phase is repeated for a sufficient number of trials and the final weights obtained in each of the trials are recorded. In the second phase, the final weights obtained from the trials will be the initial population for the ANN based training using PSO. The optimal number of population is fixed from the number

Figure 2: Block Diagram of the Proposed ANN-ANN-PSO Model

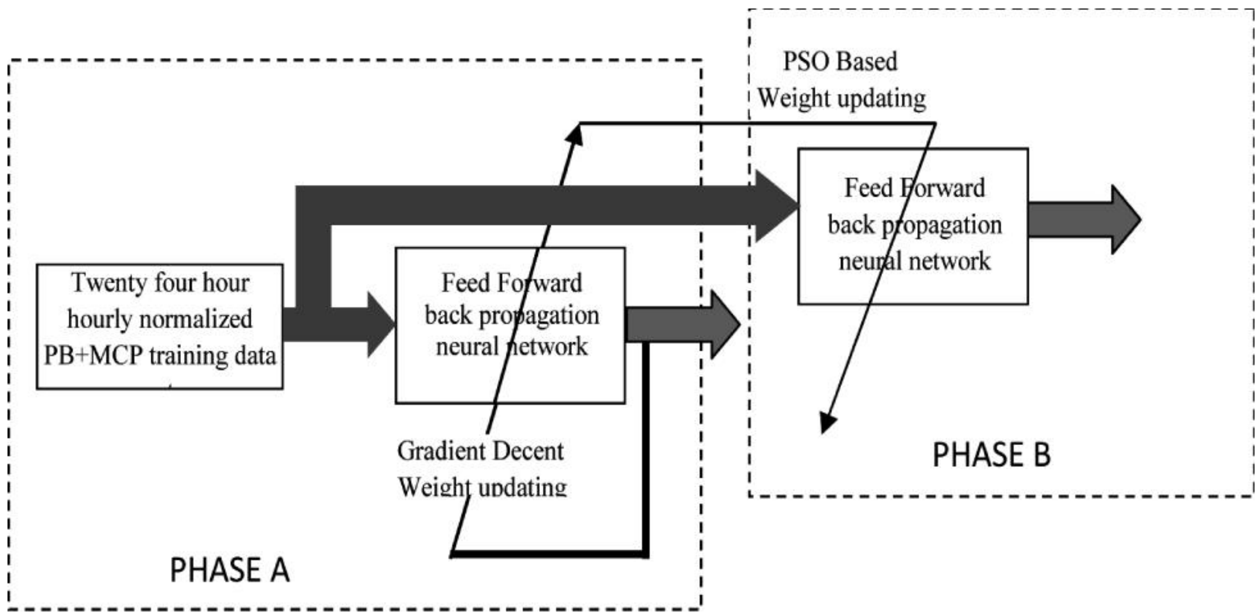
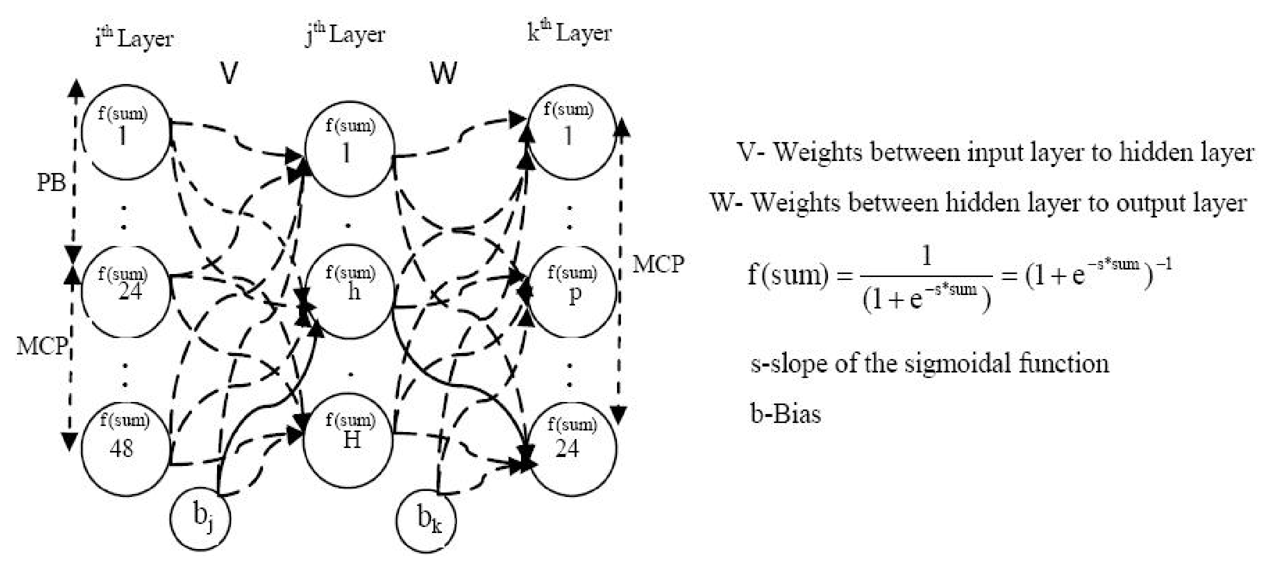


Figure 3: Architecture of the Feed Forward Back Propagation Neural Network (FFBPNN)



of trials carried out in the first phase of operation.

The PSO algorithm will improve upon the training of weights from the point where the conventional gradient based training gets stagnated. The proposed ANN-ANN-PSO model improves the PSO based training in the right direction. The block diagram shown in Figure 2, give an outline picture about the proposed model.

The architecture of the feed forward back propagation neural network is given in Figure 3. This ANN model consist of 24 input nodes and 24 output nodes representing the 24 hours hourly MCP. It consists of one hidden layer with 5 numbers of hidden nodes which is set based on trial and error method. The hidden layer and the output layer nodes consist of log-sigmoid transfer function whose output value will in the range between 0 and 1.

The historical dataset is usually not used directly in process modelling of ANNs due to the difference in magnitude of the process variables. Therefore, the data needs to be scaled to a fixed range to prevent unnecessary domination of certain variables, and to prevent data with larger magnitude from overriding the smaller and impede the premature learning process. The choice of range depends on transfer function of the output nodes in ANN. Typically [0, 1] for sigmoid function and [-1, 1] for hyperbolic tangent function. However, due to nonlinear transfer function has asymptotic limits; the range of dataset is always set slightly less than the lower and upper limits. In this work, since the sigmoid function is adopted, the data is normalized in the range of [0.1-0.9], i.e., if x_1 and x_2 is the maximum and minimum value of the training set, respectively, then the normalised data is given by $N(x)$ as in (1).

$$N(x) = \left(\frac{(x - x_1) \times (0.1 - 0.9)}{(x_2 - x_1)}\right) + 0.9 \qquad ...(1)$$

Step by Step Procedure for the Proposed ANN-ANN-PSO Model

Phase A (ANN Training)

Step by Step Algorithm of FFBPNN Architecture

Nomenclature

I = Input training vector

$$I = (i_1, ..., i_n, ..., i_{48})$$

T = Output target vector

$$T = (t_1, ..., t_y, ..., t_{24})\ *$$

$¢_y$ = Error correction weight adjustment for w_{hy} due to an error at output unit K_y

$¢_h$ = Error correction weight adjustment for v_{nh} due to an error at hidden unit J_h

$\sim$ = Learning rate

$$f(sum) = \frac{1}{1 + \exp(-sum)}$$

Activation function or Threshold function

Step 1: Set the trial number $tr = 1$.

Step 2: Set the epoch $ep = 1$.

Step 3: Generate the weights randomly to small random values between 0 and 1 to ensure that the network is not saturated by large values of weights. Let I and T be the normalized input and target training vector from set of P number of training patterns.

Step 4: Choose a training pair from the training set.

Step 5: For each training pair, do steps 6-11.

Step 6: Each input unit receives input signal in and broadcasts this signal to all units in the hidden layer J.

Step 7: Each hidden unit J_h sums its weighted input signals and the net input to the hidden unit is given as in (2) and the output at the hidden layer (J) is given as in (3). Send the output of the hidden layer signals to all units in the output units.

$$sum_{Jh} = b_J + \sum_{n=1}^{48} i_n \times V_{nh}, \quad ...(2)$$

$$f(sum_{Jh}) = \frac{1}{1 + \exp(-sum_{Jh})} \quad ...(3)$$

Step 8: Each output unit K_y sums its weighted input signals and the net input to the output unit is given as in (4) and the output at the output layer (K) is given as in (5).

$$sum_{Ky} = b_K + \sum_{h=1}^{H=5} J_h \times W_{hy}, \quad ...(4)$$

$$f(sum_{Ky}) = \frac{1}{1 + \exp(-sum_{Ky})} \quad ...(5)$$

Back Propagation of Error

Step 9: Each output unit K_y receives a target pattern corresponding to the input training pattern, computes its error information term as in (6) and calculates its weight correction term as in (7) which is used to update w_{hy} later.

$$¢_y = (t_y - K_y) \times f'(sum_{Ky}) \quad ...(6)$$

$$\Delta w_{hy} = \sim \times ¢_y \times f(sum_{Jh}) \quad ...(7)$$

The bias correction term is given in (8)

$$\Delta b_K = \sim \times ¢_y \quad ...(8)$$

Step 10: Each hidden unit J_h sums its delta inputs as in (9), multiplies by the derivative of its activation function to calculate its error information term as in (10) and calculates its weight correction term as in (11).

$$sum_{¢J} = \sum_{n=1}^{24} ¢_y \times W_{hy}, \quad ...(9)$$

$$¢_h = sum_{¢J} \times f'(sum_{Jh}) \quad ...(10)$$

$$\Delta v_{nh} = \sim \times ¢_h \times i_n \quad ...(11)$$

The bias correction term is given in (12)

$$\Delta b_J = \sim x\, ¢_h \quad ...(12)$$

Update Weights and Biases

Step 11: Each output unit K_y updates its weights and bias as in (13) and (14). Also each hidden unit J_h updates its weights and bias as in (15) and (16).

$$w_{hy}(new) = w_{hy}(old) + \Delta w_{hy} \quad ...(13)$$

$$b_K(new) = b_K(old) + \Delta b_K \quad ...(14)$$

$$w_{nh}(new) = w_{nh}(old) + \Delta w_{nh} \quad ...(15)$$

$$b_J(new) = b_J(old) + \Delta b_J \quad ...(16)$$

Go to Step 5, till all the training pairs in the training set are sent into the input layer *I* (one epoch is over). Otherwise go to Step 12.

Step 12: Do again Step 4 to Step 8 till all the training pairs in the training set are sent into the input layer *I*. Calculate the error (£), the difference between the network output and the desired output, for all the training pairs as in (17) and then the Average Mean Squared Error (AMSE) as in (18), which is calculated for every epoch. Update $ep = ep + 1$.

$$£_p^y = T_p^y - K_p^y \quad ...(17)$$

$$AMSE = \frac{\sum_{p=1}^{P}\left(\frac{\sum_{y=1}^{O} £_p^y}{O}\right)}{P} \quad ...(18)$$

Step 13: Repeat steps 2-12, if $ep < TE$ (total number of epochs), else go to step 14. The total number of epochs is fixed based on trial and error approach such that the AMSE obtained is the least. Record the final weights and biases obtained for the trial number $tr = 1$. Update $tr = tr + 1$. Also if the validation error is increasing and if the number validation checks are greater than the Validation Count (VC), then stop the training for the current trial and update $tr = tr + 1$.

Step 14: Do sufficient numbers of trials (TR) equal to the number of particle size selected in Phase B and record the final weights obtained in each of the trials. If $tr < TR$, go to step 1, else stop the execution.

Phase B (ANN-PSO Training)

In PSO system, each particle in the swarm represents a candidate solution to the optimization problem. Each particle moves with an adaptable velocity through the search space, adjusting its position in the search space according to own experience and that of neighbouring particles, then it retains a memory of the best position it ever encountered, a particle therefore makes use of the best position encountered by itself and the best position of neighbours to position itself towards the global minimum. The effect is that particles move towards the global minimum, while still searching a wide area around the best solution (Kennedy and Eberhart, 1995). In Phase B, PSO is used to update weights and biases of the neural network. The performance of each particle (i.e., the closeness of a particle to the global minimum) is measured according to AMSE which is related to the performance of the ANN training. For the purposes of this research, a particle represents the weight vector of NNs, including biases. The dimension of the search space is therefore the total number of weights and biases.

The iterative approach of PSO for ANN training can be described by the following:

Step 1: Initialize the population size which is also equal to the number of trials performed in Phase A. Get the previously stored recorded weights and biases obtained in Phase A. Also, initialize the positions and velocities of agents.

Step 2: The current best fitness achieved by particle *p* is set as *pbest*. The *pbest* with best value is set as *gbest* and this value is stored.

Step 3: Evaluate the desired optimization fitness function *fp* for each particle as the Average Mean Square Error (AMSE) over a given data set.

Step 4: Compare the evaluated fitness value of each particle with its value. If $fp < pbest$, then $pbest = fp$ and $best_{xp} = fp$, *xp* is the current coordinates of particle *p*, and $pbest_{xp}$ is the coordinates corresponding to particle *p* as the best fitness so far.

Step 5: The objective function value is calculated for new positions of each particle. If a better position is achieved by an agent, *pbest* value is replaced by the current value. As in step 2, *gbest* value is selected among *pbest* values. If the new *gbest* value is better than the previous *gbest* value, the *gbest* value is replaced by the current value and this value is stored. If $fp < gbest$, then $gbest = p$, where is the particle having the overall best fitness over all particles in the swarm.

Step 6: Change the velocity and location of the particle according to Equations (19) and (20), respectively (Kennedy, 1997)

$$V_i = wV_{i-1} + acc \times rand() \times (best_{xp} - xp) + acc \times rand() \times (best_{xgbest} - xp) \quad ...(19)$$

where, *acc* is the acceleration constant that control how far particles *p* move from one another, and *rand* returns a uniform random number between 0 and 1.

$$xp = xpp + V_i \quad ...(20)$$

V_i is the current velocity, V_{i-1} is the previous velocity, *xp* is the present location of the particle, *xpp* is the previous location of the particle, and *i* is the particle index. Here, the coordinates $best_{xp}$ and $best_{xgbest}$ are used to pull the particles towards the global minimum.

Step 7: Fly each particle *p* according to Equation (20).

Step 8: If the maximum number of predetermined iterations is exceeded, then stop; otherwise go to step 3 until convergence.

Performance Evaluation

The accuracy of the results in this case study is evaluated based on three error indices. They are: Mean Absolute Percentage Error (MAPE), Normalized Mean Square Error (NMSE) and Error Variance (EV). The Mean Absolute Percentage Error (MAPE) is defined by the following Equation (21).

$$MAPE = \frac{1}{NH} \sum_{i=1}^{NH} \left| \frac{P_i - A_i}{A_i} \right| \quad ...(21)$$

NMSE (Amjady *et al.*, 2011) is defined as:

$$NMSE = \left[\frac{1}{\Delta^2 NH} \sum_{i=1}^{NH} \left(P_i - A_i \right)^2 \right] \quad ...(22)$$

where,

$$\Delta = \frac{1}{NH-1} \sum_{i=1}^{NH} \left(A_i - A_{Ave} \right)^2$$

EV (Amjady *et al.*, 2011) is defined as:

$$\sigma^2 = \frac{1}{NH}\sum_{i=1}^{NH}\left(\left|\frac{P_i - A_i}{A_i}\right| - MAPE\right)^2 \quad ...(23)$$

where, P_i and A_i are the i^{th} predicted and actual values respectively, A_{Ave} is the mean of the actual value and *NH* is the total number of predictions.

RESULTS AND DISCUSSION

The proposed ANN-ANN-PSO model is used to forecast MCP from March 6 to March 12, 2014. The architecture of the proposed model consists of 48 input nodes and 24 output nodes with 5 hidden nodes in the hidden layer. The hidden nodes are fixed based on trial and error. The parameter settings such as learning rate (0.9), momentum factor (0.9), slope factor (0.05) and validation count (10) are kept the same so as to have a fair comparison on the same reference among other ANN models to be compared. The weights and bias are initialized randomly between zeros to one. The neural network training is carried out for the following 3 ANN models.

1. Feed forward back propagation neural network.
2. ANN-ANN-PSO (Proposed model).
3. Feed forward back propagation neural network with PSO (ANN-PSO).

The first model (a) is the conventional feed forward back propagation neural network where the weights are updated using standard gradient descent algorithms. The second model, (b) which is the proposed model, updates the weights using PSO. Here the final weights obtained from the various trials of ANN training are kept as the initial weights. In the third model (c), weights are updated using PSO where the weights are initially generated randomly. All the simulations are carried out using codes developed in MATLABR2009a software.

Feed Forward Back Propagation Neural Network

The training for the source and the target data is carried out for 50 trials. The number of epochs for the training is fixed to 10,000. The training will be stopped at any time if the MAPE during validation at every epoch does not improve upon the convergence within a fixed number of epochs. The validation check helps to understand that whether the network is over trained. This information will give an idea about the forecasting accuracy for the sample set of data considered for validation which is related to the forecasting accuracy of the typical input test pattern not present in the training set. It should also be noted that it is not always necessary that if the MAPE during validation decreases, then the MAPE during verification will also decrease. The reason is due to the dynamic nature and uncertainties involved in the activities of the power system market. Table 1 gives the statistical results of the performance indices of the feed forward back propagation neural network. The convergence of the training and validation plots for the 50 trials is presented in Figures 4 and 5, respectively. The best result of the forecasted MCP is shown in Figure 6. The training termination of every trial is carried out when the validation checks fails and can be observed from Figure 5. The best forecasted MCP in Figure 6 is closer to the actual MCP with an EV of 139.8866.

Figure 4: Convergence Plot for 50 Trials

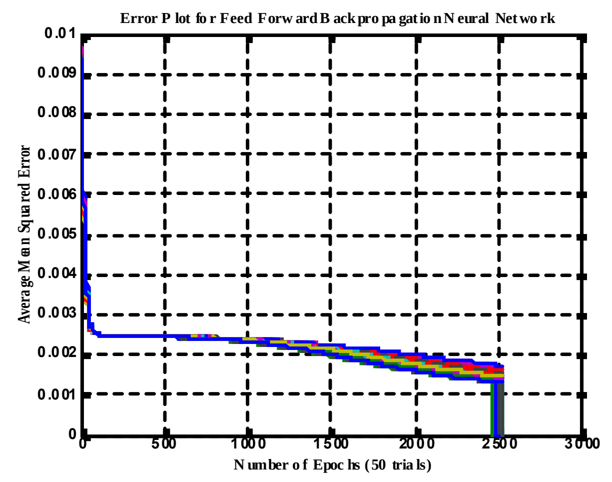
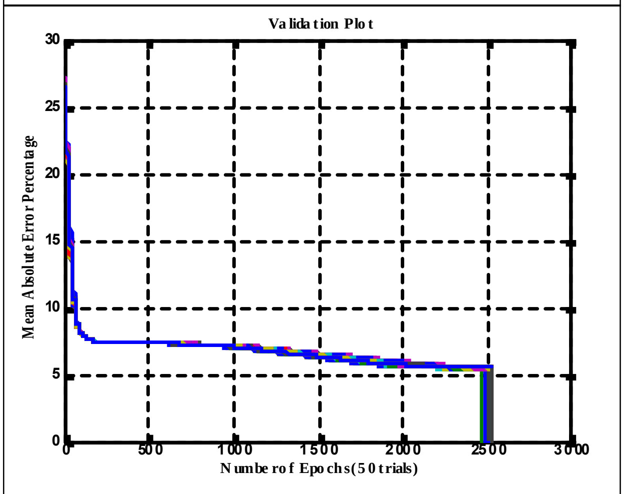


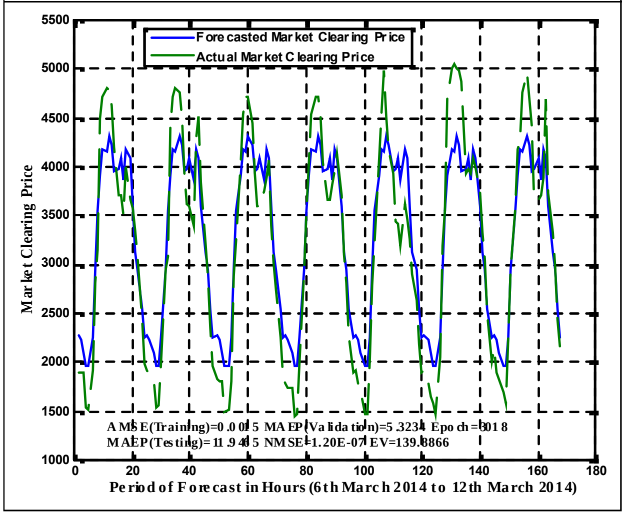
Figure 5: Validation Plot for 50 Trials



ANN-ANN-PSO (Proposed Model)

After the Phase B of the ANN training, the final weights obtained from the 50 trials is

Figure 6: Forecasted Market Clearing Price



initialized for the ANN training of weights using PSO. The architecture of the ANN-PSO is same as that of the Phase A ANN architecture. The controlling parameters for the PSO algorithm are presented in Table 2. The parameter setting for the PSO algorithm is fixed based on trial and error approach. Here, the convergence of the error plot is not as smooth as that of the ANN Phase A training. The reason is that the stochastic nature of the particle search in finding the optimal weights for minimizing the MAPE while training and also satisfying the validation check. In Phase B, a maximum of 25 trials is carried out and the statistical analysis is performed and presented in Table 3. The

Table 1: Statistical Analysis for 50 Trials (ANN Mode)

Performance Indicesm	Average	Best	Worst
Average Mean Square Error (Training)	0.001374	0.0015	0.0013
Mean Absolute Error Percentage (Validation)	5.451778	5.3234	5.6551
Stopping Epoch	2695 (Approx)	3018	2570
Mean Absolute Error Percentage (Testing)	12.29162	11.9465	12.7742
Normalized Mean Square Error	1.24E-07	1.20E-07	1.29E-07
Error Variance	148.1191	139.8866	159.9431

Table 2: PSO Parameter Settings

Parameters	No. of Population	Acceleration Constant (Local)	Acceleration Constant (Global)	Max. Velocity	Inertial Weights	
					Initial Iteration	Final Iteration (3500)
Value	50	0.00005	0.99999	100000	0.9	0.1

Table 3: Statistical Analysis for 25 Trials (ANN-ANN-PSO Mode)

Performance Indicesm	Average	Best	Worst
Average Mean Square Error (Training)	2.41E-01	0.001296	0.001136
Mean Absolute Error Percentage (Validation)	5.514823	5.541751	5.559755
Stopping Epoch	364 (Approx.)	266	455
Mean Absolute Error Percentage (Testing)	12.2114	10.91058	15.59951
Normalized Mean Square Error	1.35E-07	8.61E-08	1.40E-07
Error Variance	1.48E+02	1.17E+02	2.39E+02

Figure 7: Convergence Plot for 25 Trials

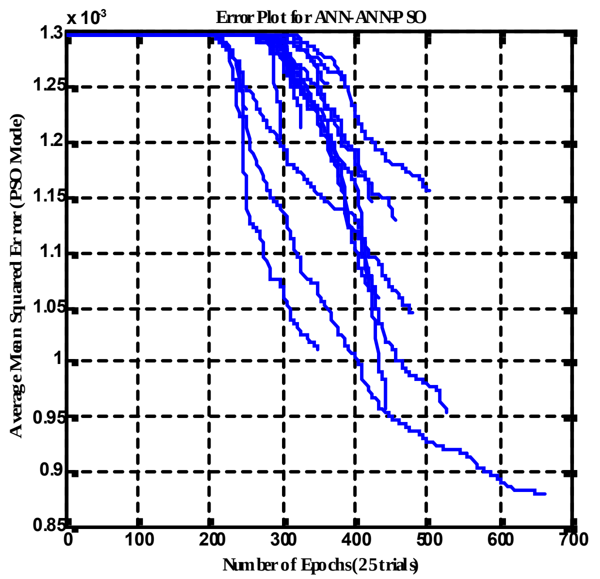
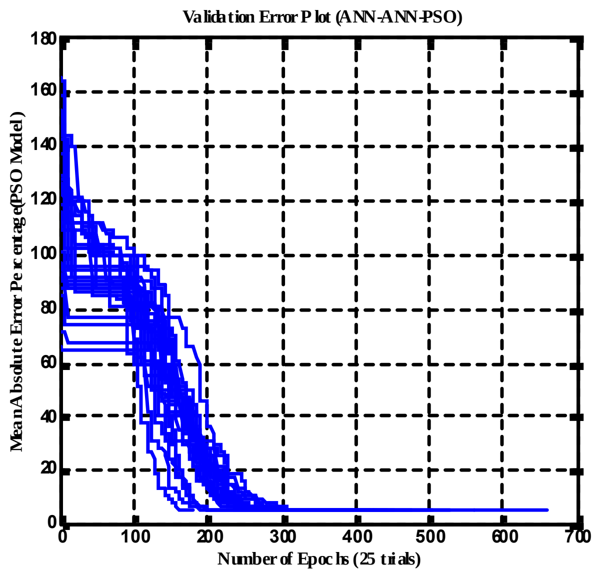


Figure 8: Validation Plot for 25 Trials



convergence and validation plots and the best forecasted result out of 25 trials are shown in Figures 7, 8 and 9, respectively. The stochastic search of the PSO algorithm strives to search the optimal weights in the search space where the conventional gradient search of the ANN training of Phase A fails. It is observed that the convergence in most of the trials is better than the best MAPE of the training error (0.0015) observed in the previous model (a). The training termination of every trial is carried out when the validation checks fails and is observed in Figure 8. The best forecasted MCP in Figure 9 is much closer to the actual MCP with an EV of 1.17E+02.

Figure 9: Forecasted Market Clearing Price

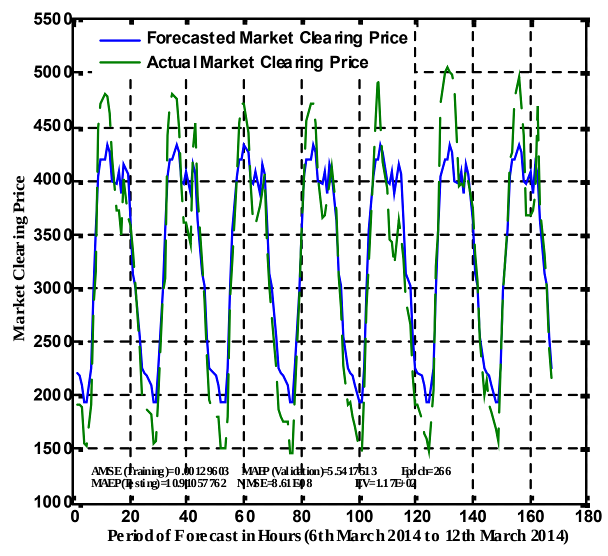


Table 4 gives the statistical information about the performance indices for the 15 trials. The poorer performance of the ANN-PSO is due to the random initialization of weights which does not reinforce the optimal tuning of weights.

CONCLUSION

This research work concludes that the proposed ANN-ANN-PSO sequential and auxiliary hybrid model performs better than the ANN-PSO auxiliary hybrid model. The proposed model is suitably developed for the IEX in forecasting MCP. The concept of validation check by selecting a sample set of training patterns at every epoch prevents the neural network training from overtraining. The proposed model can be used in other power system markets since the approach is simple and less complicated.

Table 4: Statistical Analysis for 15 Trials (ANN-PSO Mode)

Performance Indicesm	Average	Best	Worst
Average Mean Square Error (Training)	0.043766	0.035076	0.02664
Mean Absolute Error Percentage (Validation)	42.95142	36.55854	36.06834
Stopping Epoch	1000	1000	1000
Mean Absolute Error Percentage (Testing)	56.50932	41.39489	1.07E+02
Normalized Mean Square Error	1.68E-01	1.79E-06	9.84E-06
Error Variance	3.41E+03	1.68E+03	1.12E+04

Feed Forward Back Propagation Neural Network with PSO (ANN-PSO)

The feed forward back propagation neural network is trained with PSO for 15 trials. The same ANN architecture is used as given in the previous models. Here, the initialization weights are randomly chosen. The parameter settings for the PSO algorithm are kept similar to that of the previous model (b). This is done in order to compare the models having the same reference platform. The performance of the ANN-PSO model is comparatively poor in performance and therefore, the number of iterations considered for analysis is also less.

REFERENCES

1. Aggarwal S K, Saini L M and Kumar A (2008), “Electricity Price Forecasting in Ontario Electricity Market Using Wavelet Transform in Artificial Neural Network Based Model”, *International Journal of Control, Automation, and Systems*, Vol. 6, No. 5, pp. 639-650.

2. Amjady N, Keynia F and Zareipour H (2011), "Short Term Wind Forecasting Using Ridgelet Neural Network", *Electrical Power Systems Research*, Vol. 81, pp. 2099-2107.
3. Areekul P, Senjyu T, Toyama H and Yona A (2010), "A Hybrid ARIMA and Neural Network Model for Short-Term Price Forecasting in Deregulated Market", *IEEE Transactions on Power Systems*, Vol. 25, pp. 524-530.
4. Catalao J P S, Pousinho H M I and Mendes V M F (2011), "Hybrid Wavelet-PSO-ANFIS Approach for Short-Term Electricity Prices Forecasting", *IEEE Transactions on Power Systems*, Vol. 26, No. 1, pp. 137-144.
5. Conejo A J, Contreras J, Espý´nola R and Plazas M A (2004), "Forecasting Electricity Prices for a Day-Ahead Pool-Based Electric Energy Market", *International Journal of Forecasting*, Vol. 21, pp. 435-462.
6. Diongue K A, Dominique G and Bertrand V (2004), "A k-Factor GIGARCH Process: Estimation and Application on Electricity Market Spot Prices", Proceedings of 2004 International Conference on Probabilistic Methods Applied to Power Systems, pp. 1-7.
7. Kennedy J (1997), "The Particle Swarm: Social Adaptation of Knowledge", Proceedings of IEEE International Conference on Evolutionary Computation, pp. 303-308.
8. Kennedy J and Eberhart R (1995), "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks IV, pp. 1942-1948.
9. Khashei M, Hejazi S R and Bijari M (2008), "A New Hybrid Artificial Neural Networks and Fuzzy Regression Model for Time Series Forecasting", *Fuzzy Sets and Systems*, Vol. 159, No. 7, pp. 769-786.
10. Pany P K and Ghoshal S P (2013), "Day-Ahead Electricity Price Forecasting Using PSO Based LLWNN Model", *International Journal of Energy Engineering*, Vol. 3, No. 4, pp. 99-106.
11. Peter S E, Raglend I J and Simon S P (2014), "An Architectural Frame Work of ANN Based Short Term Electricity Price Forecast Engine for Indian Energy Exchange Using Similar Day Approach", *IMPACT: International Journal on Research & Engineering Technology*, Vol. 2, No. 4, pp. 111-122.
12. Pindoriya N M, Singh S N and Singh S K (2008), "An Adaptive Wavelet Neural Network-Based Energy Price Forecasting in Electricity Markets", *IEEE Transactions on Power Systems*, Vol. 23, No. 3, pp. 1423-1432.
13. Singhal D and Swarup K S (2011), "Electricity Price Forecasting Using Artificial Neural Networks", *Electrical Power and Energy Systems*, Vol. 33, pp. 550-555.