EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

# Day-ahead electricity price forecasting model based on artificial neural networks for energy markets

S. Anbazhagan[1,]* and Bhuvaneswari Ramachandran[2]

[1] Department of EEE, Faculty of Engineering and Technology, Annamalai University, Annamalai Nagar, India 608002
[2] Department of Electrical and Computer Engineering, University of West Florida, Pensacola, FL 32514, USA

## Abstract

Day-ahead electricity price forecasting is still an open problem in electricity markets. One major method is used in solving this problem is artificial neural networks (ANN). But they are usually trained slowly and need large numbers of patterns. NN trained using Levenberg-Marquardt (LM) learning is proposed and partial autocorrelation is applied on time series data to get correct input values. The functionality of the NN-LM is higher than the traditional ANN and some other hybrid approaches. To show the effectiveness and accuracy of the NN-LM method, the Indian and the Austrian energy exchange markets are considered. It is significant to note that for the very first time, the NN-LM based approach is being tested on both the energy markets. Finally, the flexibility of the proposed approach is checked using a 4-fold cross-validation technique. The 4-fold cross-validation strategy is capable of improving the generalization ability of the model and accomplishing higher forecast precision.

**Keywords:** Autocorrelation, back propagation neural network, Deregulated electricity market, Leverberg-Marquardt learning.

Received on 05 December 2019, accepted on 01 December 2020, published on 23 December 2020



doi: 10.4108/eai.23-12-2020.167660

*Corresponding author. Email:s.anbazhagan@gmx.com

## 1. Introduction

After the 80s, numerous nations have changed the economics of their electricity markets from monopolies to oligopolies in an effort to increase competition. A significant feature of this change is to allow competition among generators and create market conditions in the industry which are necessary to decrease the cost of energy production and distribution, eliminate certain inefficiencies and increase customer choices [1]. The meaning of deregulation is the reduction or elimination of government control over a particular industry. The purpose of deregulation is to promote more competition within the same industries in the same geographical jurisdiction. It is generally believed that a fewer and simpler regulation will lead to a raised level of competitiveness and the overall result will be higher productivity and more efficiency at lower price. [1].

Deregulation of electrical markets calls for the restructuring of the electricity industry. The traditional vertically integrated system is broke down into three separate businesses are generation, transmission and distribution company. These three businesses are operated by three different entities. Deregulators advocate that deregulated electric market will bring cheaper electricity and in the meantime provide more choices for the customers [3]. In a deregulated market, instead of one generation provider, there are several generation providers in a local area. The local regulatory body can no longer fix the electricity price. The consumers have a choice regarding their local electricity providers. They can choose different electricity providers depending on their requirements and demand.

The deregulation of power industry creates new challenges in the electricity market due to forecasting of price which has become a major issue globally. The price is volatile in nature. From an economic point of view, electricity is non-storable goods which make balance between demands and supply a herculean task. The electricity prices in competitive markets are directly or indirectly affected by a number of factors which are interlinked to each other. Uncertainty factors like load, weather, market forces, bidding strategy etc arc fluctuating, and hence prediction of price is difficult. Accurate forecasting of price is not a trivial task. The electricity price forecasting is important for all market players and its behaviour is different from other commodities. Through competition, market deregulation strives to improve generation availability and efficiency.

Electricity price prediction is more complex than load forecasting because of uncertainties in operation as well as bidding strategies of market participants. The volatility and non-linearity of the system directly affects the accuracy of price prediction. Significance of price prediction and its complexity have motivated the researchers to be more innovative and propose numerous strategies. Among these methodologies are the time series and artificial neural network (ANN) models [3, 4].

Time series models, for example, dynamic regression and transfer function, autoregressive integrated moving average (ARIMA), generalized auto-regressive conditional heteroskedastic and hybrid methods based on wavelet transform (WT) and ARIMA (WT-ARIMA) have been proposed in the literature. Time series models are linear predictors, and they experience issues in foreseeing non-linear behaviour of electricity price. Hence ANN is utilized to solve this problem. The advantage of ANN is their non-linear modelling ability and ability to capture the high volatility prices [5]. When the supplier offers a price equal to or below market clearing price (MCP), it is set to that price at that hour. In deregulated power market based pools, purchaser organizations submit bids for selling and purchasing power for the next 24hr time period. In electricity price forecasting, ANN takes previous day's prices as input factors. Accuracy in forecasting MCP relies upon natural and extraneous components [5].

ANN with modified Levenberg–Marquardt (LM) learning algorithm is implemented in Penn-Jersey-Maryland (PJM) market. The approach using ANN predicts the 24hr locational marginal price (LMP) of day-ahead energy market [6]. The results obtained are compared with dynamic regression, transfer function, ARIMA, WT, simple application of neural networks (NN). Also fuzzy c-mean method is used to classify three clusters. NN are used to forecast MCP for day-ahead energy market. Three layered back-propagation (BP) network was chosen for structure of NN. The result showed 16% error on week days and less than 20% error on a week end. The accuracy can be improved by combining several techniques such as fuzzy logic, NN and dynamic clustering together. NN in open market was presented [7] by researchers.

Many authors have published their work in electricity price forecasting utilizing different computational intelligence techniques, for example, feed-forward NN, extreme learning machines, recurrent neural networks, bat optimized NN and particle swarm optimization (PSO) based ANN [8-15]. In the literature, the greater part of the NN-based research papers have utilized day-ahead electricity price forecasting approach [16-18].

Among the existing approaches, [19] proposes to develop a hybrid ANN forecast engine solely to Indian energy exchange, since only meagre work is carried out in this market. Hybrid ANN models, which combine heuristic search algorithms, such as ANN-ANN-PSO, Wavelet-based ANN and Wavelet-based ANN-ANN-PSO were developed to forecast MCP [19]. In the process of training ANN, the weights were updated based on conventional gradient descent method. It should be noted that the weights can be updated either in incremental or in batch modes [19]. However, the conventional gradient descent method [19] had an obvious drawback of getting stuck in local minima.

The proposed research work in this paper develops a NN trained electricity price forecasting model using LM (NN-LM) network and is used for forecasting the MCP. The proposed work uses a novel approach to eliminate the drawbacks mentioned in the earlier paragraphs and is implemented on two test systems, one on an Indian market and the other on an Austrian energy market to compare the forecast results. The findings show the accuracy and efficacy of the proposed approach. The legitimacy and versatility of the proposed approach are verified by comparing the obtained results with that from 4-fold cross-validation.

The rest of the paper is organized as follows. The methodology and detailed NN-LM learning are presented in Section 2. The experimental results and discussion are provided in Section 3. The conclusion is described in Section 4.

## 2. Methodology

This section of the research describes the data source, input feature selection for NN-LM model, NN-LM model for day-ahead price forecasting in the electricity markets of Indian as well as Austrian energy markets and their prediction performance evaluation.

### 2.1. Data source

In this paper, the data for electricity prices data are taken from the daily trading reports of Indian as well as Austrian energy markets and are presented on a monthly basis. The Indian and Austrian dataset consists of MCP [20, 21].

### 2.2. Input feature selection using correlation

Choosing the most suitable inputs to a model is the imperative initial phase in model building. It is particularly critical for NNs that are intense, non-linear processors. For time-series, inputs additionally incorporate lags (memory length). As the input dimensionality expands, multifaceted nature of the model increments and learning turns out to be more troublesome, prompting poor convergence. With fewer applicable sources of inputs, a network can concentrate on building up the required associations with more efficiency. The test is to choose from all the potential data sources, a subset of information sources that will prompt an unrivalled model. If there are several inputs in the time-series, then it is necessary to find the appropriate lags that are significant to the output for each time-series [22].

The electricity price information of the earlier day ($n^{th}$ day) is mapped with the following day $(n+1)^{th}$ day while modelling neural systems. The reason is that the strength of the correlation of both linear and nonlinear parameters between $n^{th}$ and $(n+1)^{th}$ day is more stronger [19]. For instance, the MCP profile on Monday of the earlier day is mapped to Tuesday of the following day. So when a test contribution of the $n^{th}$ day is fed into the prediction model, the MCP of the $(n+1)^{th}$ MCP is predicted. However, it should be noted that researchers discussed short-term electricity price forecasting in which the electricity price or MCP is forecasted for a day or a week [19].

The correlation gives the level of direct relationship between two variables, which involves how firmly the two variables are identified with each other [23]. The extent to which the variables are related can be determined by deciding the correlation coefficient, whose value is limited between −1 and 1. The three potential outcomes of being positively correlated, negatively correlated and not correlated correspond to the correlation coefficients with values close to 1, −1, and 0, respectively.

Correlation is the most prevalent investigative method for choosing inputs and the quantity of lags. Correlation examination has been utilized by a few researchers [22-24] for input determination in electricity price forecasting.

The lags (price of the previous hours) of Indian market (a week in June) $P_{h-1}$, $P_{h-2}$, $P_{h-8}$, $P_{h-22}$, $P_{h-23}$, $P_{h-24}$, $P_{h-25}$, $P_{h-34}$, $P_{h-35}$, $P_{h-36}$, $P_{h-47}$, $P_{h-49}$, $P_{h-52}$, $P_{h-62}$, $P_{h-63}$, $P_{h-70}$, $P_{h-72}$, $P_{h-77}$, $P_{h-79}$, $P_{h-81}$, $P_{h-83}$, $P_{h-84}$, $P_{h-87}$, $P_{h-89}$, $P_{h-91}$, $P_{h-94}$; (a week in September) $P_{h-1}$, $P_{h-2}$, $P_{h-8}$, $P_{h-19}$, $P_{h-20}$, $P_{h-21}$, $P_{h-22}$, $P_{h-23}$, $P_{h-24}$, $P_{h-25}$, $P_{h-40}$, $P_{h-45}$, $P_{h-46}$, $P_{h-48}$, $P_{h-52}$, $P_{h-58}$, $P_{h-64}$, $P_{h-65}$, $P_{h-68}$, $P_{h-74}$, $P_{h-76}$, $P_{h-77}$, $P_{h-80}$, $P_{h-83}$, $P_{h-88}$, $P_{h-102}$, $P_{h-103}$, $P_{h-107}$, $P_{h-111}$, $P_{h-112}$, $P_{h-113}$, $P_{h-115}$, $P_{h-122}$; (First week of October) $P_{h-1}$, $P_{h-2}$, $P_{h-4}$, $P_{h-14}$, $P_{h-18}$, $P_{h-24}$, $P_{h-25}$, $P_{h-39}$, $P_{h-54}$, $P_{h-59}$, $P_{h-67}$, $P_{h-72}$, $P_{h-86}$, $P_{h-89}$; (Second week of October) $P_{h-1}$, $P_{h-2}$, $P_{h-16}$, $P_{h-20}$, $P_{h-21}$, $P_{h-23}$, $P_{h-24}$, $P_{h-52}$, $P_{h-59}$, $P_{h-61}$, $P_{h-67}$, $P_{h-80}$, $P_{h-81}$, $P_{h-82}$, $P_{h-83}$, $P_{h-90}$, $P_{h-92}$, $P_{h-95}$, $P_{h-96}$, $P_{h-98}$, $P_{h-108}$, $P_{h-116}$, $P_{h-118}$, $P_{h-119}$, $P_{h-129}$, $P_{h-136}$ are considered as the input features.

The lags of Indian market in the year 2019 (a week in June) $l \varepsilon L = \{1, 2, 21, 22, 23, 24, 48, 74, 84, 88, 90, 92, 93, 95, 96, 97, 98, 101, 103, 106, 107, 110, 112, 113, 117\}$; (a week in September) $l \varepsilon L = \{1, 2, 23, 24, 25, 47, 55, 63, 66, 67, 69, 70, 78, 79, 81, 82, 83, 84, 89, 90, 92, 97, 100, 102, 104, 109, 110, 115, 116, 121\}$; (First week in October) $l \varepsilon L = \{1, 2, 23, 24, 37, 48, 85, 94, 96, 97, 100, 121, 140, 142, 144, 145, 146\}$; (Second week in October) $l \varepsilon L = \{1, 2, 4, 8, 17, 23, 24, 59, 68, 72, 83, 85, 87, 112\}$ are considered as the input features.

The lags of Austrian market (a week during Winter) $P_{h-1}$, $P_{h-2}$, $P_{h-4}$, $P_{h-20}$, $P_{h-21}$, $P_{h-22}$, $P_{h-23}$, $P_{h-52}$, $P_{h-62}$, $P_{h-73}$, $P_{h-79}$, $P_{h-80}$, $P_{h-81}$, $P_{h-82}$, $P_{h-83}$, $P_{h-84}$, $P_{h-86}$, $P_{h-88}$, $P_{h-93}$, $P_{h-96}$, $P_{h-97}$, $P_{h-99}$, $P_{h-100}$, $P_{h-104}$, $P_{h-110}$, $P_{h-111}$, $P_{h-115}$, $P_{h-119}$, $P_{h-127}$, $P_{h-128}$, $P_{h-129}$; (a week during Spring) $P_{h-1}$, $P_{h-2}$, $P_{h-4}$, $P_{h-8}$, $P_{h-9}$, $P_{h-10}$, $P_{h-23}$, $P_{h-42}$, $P_{h-44}$, $P_{h-73}$, $P_{h-87}$, $P_{h-88}$, $P_{h-93}$, $P_{h-94}$, $P_{h-95}$, $P_{h-97}$, $P_{h-100}$, $P_{h-106}$, $P_{h-109}$, $P_{h-116}$, $P_{h-123}$, $P_{h-128}$, $P_{h-129}$, $P_{h-132}$, $P_{h-140}$, $P_{h-144}$, $P_{h-145}$; (a week during Summer) $P_{h-1}$, $P_{h-2}$, $P_{h-7}$, $P_{h-18}$, $P_{h-73}$, $P_{h-86}$, $P_{h-87}$, $P_{h-88}$, $P_{h-90}$, $P_{h-99}$, $P_{h-101}$, $P_{h-110}$, $P_{h-111}$, $P_{h-113}$, $P_{h-117}$, $P_{h-122}$, $P_{h-123}$, $P_{h-131}$, $P_{h-136}$; (a week during Fall) $P_{h-1}$, $P_{h-2}$, $P_{h-17}$, $P_{h-20}$, $P_{h-22}$, $P_{h-34}$, $P_{h-39}$, $P_{h-56}$, $P_{h-68}$, $P_{h-85}$, $P_{h-90}$, $P_{h-92}$, $P_{h-93}$, $P_{h-98}$, $P_{h-99}$, $P_{h-102}$, $P_{h-106}$, $P_{h-108}$, $P_{h-116}$, $P_{h-132}$, $P_{h-141}$, $P_{h-143}$, $P_{h-144}$, $P_{h-148}$ are considered as the input features.

The inputs for the proposed models depend on this correlation investigation. The chosen inputs (lagged prices) demonstrate the impact of short-run trend, every day periodicity and week after week periodicity. There are different inputs for the proposed prediction model in Indian and Austrian power markets during all of the test periods.

## 2.3. Neural network trained using Levenberg–Marquardt learning

ANN is made up of neurons organized in layers, as illustrated in Figure 1. The data is fed into the network through an input layer. This is followed by setting up at least one intermediate (hidden) layer. The output data comes out of the network's last layer [4]. The transfer function contained in the individual layers can be nearly anything. It describes the mathematics behind the NN with LM training algorithm.

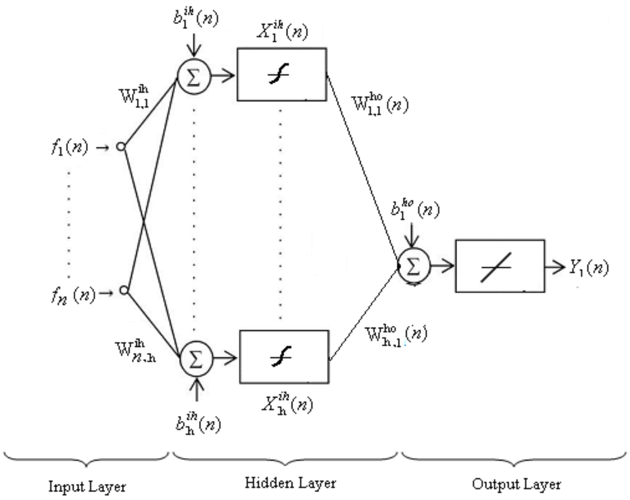


**Figure 1.** Implementation of NN-LM for electricity price forecasting

When an input vector is presented to the NN, the output error can be computed by a squared error. The squared error is calculated as the sum of the squared differences between the target values and the output values.

$$e(n)=\frac{1}{2}\sum_{k=1}^{m}\left(t_k(n)-y_k(n)\right)^2 . \quad (1)$$

where $t_k(n)$ is the target output for the $k^{th}$ in the output layer when pattern $n$ is presented and $y_k(n)$ is the net output for the $k^{th}$ in the output layer when pattern $n$ is presented. The output error for all input vectors presented to the feed-forward is given by

$$e=\sum_{n=1}^{N}e(n) . \quad (2)$$

The goal of training algorithm is to iteratively adjust the weights in the network to generate the preferred output by minimizing the output error. BP is a gradient-descent approach in that it utilized the minimization of first-order derivatives to locate an ideal solution. It works with a training set of input vectors *f*, and target output vectors *t*. The training algorithm iteratively tries to constrain the created outputs described by vector y to sought after target vector *t*, by modifying the weights in the network through a training calculation.

Quasi-Newton methods are popular algorithms for nonlinear optimization. They use second-order derivatives to find the optimal solution, so they generally converge faster than the first-order techniques such as the gradient-descent method used in BP [25]. Quasi-Newton methods can be used to train NNs, and they can be used in most configurations that work for BP [26]. The second-order partial derivatives are computed in a Hessian matrix, *H*. The weight update is the product of the inverse Hessian matrix *H*, and the direction of the steepest descent, *g*. Since it works on the average gradient of the error surface, a batch update of weights is performed at the end of each epoch [27, 28].

$$\Delta w=-H^{-1}g . \quad (3)$$

Since determining the weight updates involves the use of a Hessian matrix with all the second-order derivatives, the computation is difficult and time consuming. By using approximations to the Hessian matrix, speed can be increased. In general, Quasi-Newton techniques can become stuck in local minima more often than the other optimization techniques [27, 28].

The LM algorithm is nonlinear optimization based on the use of second-order derivatives [25-28]. It has been adapted for use on training NNs. The main weakness of the LM algorithm is that it desires the storage of several matrices that can be quite large for definite problems. It also works only with summed squared error functions, so it is often used for estimation (i.e., regression) applications.

The LM algorithm is a succession of the features of gradient descent found in BP and the Newton method [27, 28]. It assumes that the underlying function being modelled is linear and that the minimum error can be found in one step. It calculates weight change to make this single step. It tests the network with these new weights to determine whether the new error is lower. A change in weights is only accepted if it improves the error. When the error decreases, the weight change is accepted and the linear assumption is reinforced by decreasing a control parameter, $\mu$. When the error increases, the weight change is rejected and like BP, it follows a gradient descent by increasing the control parameter to de-emphasize the linear assumption. Along these lines, the LM calculation is a bargain between a Newton and gradient-decent process [25-28]. Close to a base, the linear supposition is roughly genuine so the LM calculation gains exceptionally fast ground by utilizing this second-order Newton-like feature. The procedure is repeated until the stage when the desired error is reached or maximum number of iterations is reached.

The LM calculation approximates the Hessian matrix utilized as a part of the Quasi-Newton technique as the result of a Jacobian matrix of the first-order partial derivatives, with its transpose as appeared in Eq. (4). Since it utilizes a Jacobian matrix *J*, rather than Hessian matrix *H*, the estimation is simpler [28].

$$H\approx J^T J . \quad (4)$$

The gradient is calculated as the result of the Jacobian containing the first-order partial derivatives and a vector e that contains the errors being minimized.

$$g=J^T e . \quad (5)$$

This gives us a weight-update formulation in Eq. (6), where $I$ is the identity matrix and $\mu$ is the control parameter.

$$\Delta w=-\left(J^T J+\mu I\right)^{-1}J^T e . \quad (6)$$

From Eq. (6), it is revealed that $\mu$ is 0, and this is a Newton routine with an approximated Hessian matrix. The larger values of $\mu$ make it look more like a gradient-descent method.

The LM training procedure is follows:

(i) Initialize weights. Set $W^{ih}(n)$ and $W^{ho}(n)$ to small random values,
(ii) Present each pattern to the input of the network,
(iii) Propagate data forward and produce the output pattern. Determine the error between the target output and the actual output,
(iv) If there are more patterns (i.e., n < N) in the training set, loop back to step (ii),
(v) Now estimate the error vector *e*, between the target and actual output for all patterns presented by using summed squared error as in Eq. (1),
(vi) Calculate the Jacobian matrix, *J*, from the first-order partial derivatives,
(vii) Calculate the weight update as given in Eq. (3),
(viii) Recalculate the sum of squared errors. If the new error is lower, reduce $\mu$ by some factor, update the weights by $\Delta W$, and go to step (ix). If the new error is higher, increase $\mu$ by some factor and go back to step (vii),

(ix) If the norm of the gradient $g$, is less than the preferred amount, stop; otherwise loop back to step (i).

### 2.4. Prediction performance evolution

The input features and the target output (actual electricity price) are linearly normalized in the range of {-1, 1} before being presented to the NN-LM model and the output from the NN-LM model was de-normalized before being presented in performance evaluation. The performance of the trained network is then evaluated by comparison of the network output with its actual value via statistical evaluation indices.

The mean absolute percentage error (MAPE), the normalized mean square error (NMSE) and the error variance (EV) are used to evaluate the performance of forecasting in electricity prices.

The MAPE can be defined as

$$MAPE = \frac{1}{N}\sum_{h=1}^{N}\left|\frac{A_h - F_h}{A_h}\right| \times 100, \quad (7)$$

where $A_h$ and $F_h$ are the actual and forecasted electricity prices of $h^{th}$ hour, respectively, and $N$ is the number of forecasted hours.

The NMSE is given by

$$NMSE = \left[\frac{1}{\Delta^2 N}\sum_{h=1}^{N}(F_h - A_h)^2\right]. \quad (8)$$

where

$$\Delta = \frac{1}{N-1}\sum_{h=1}^{N}(A_h - A_{Ave})^2. \quad (9)$$

where $A_{Ave}$ is average of actual ata.

The EV is given by

$$\sigma^2 = \frac{1}{N}\sum_{h=1}^{N}\left(\left|\frac{F_h - A_h}{A_h}\right| - MAPE\right)^2. \quad (10)$$

The MAPE, NMSE and EV were used in the experimental results in this case study. If a model has smaller MAPE, NMSE, and EV, then that means that it is well performing both in space and in times as well as more precise will be the prediction of prices. The detailed discussion on three error indices are presented here [19].

## 3. Numerical results

This section presents the case study of energy exchanges in Indian and Austrian electricity markets which were forecasted by the proposed NN-LM model.

### 3.1. Case studies

The day-ahead electricity market of the Indian energy exchange and energy exchange of Austria are considered in this real-world case study.

In the energy market of an Indian market, price changes are identified by key behaviour of the dominating player, which are difficult to predict. It could be observed that the series introduced in Figs. 2–5 (four weeks of June, September and two weeks of October) have shaky mean and variance. This temperamental conduct makes forecasting hard. Along these lines, it could be obviously observed that the Indian power market is a genuine case study with adequate unpredictability. Hence, researchers have used the Indian electricity market as a benchmark case study [19].

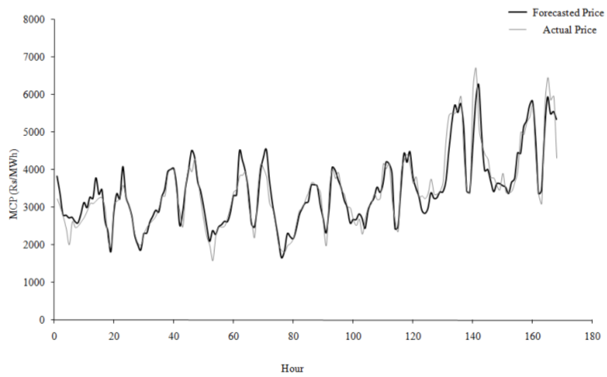


**Figure 2.** Forecasted MCP for a week in June (Indian market in the year 2014)

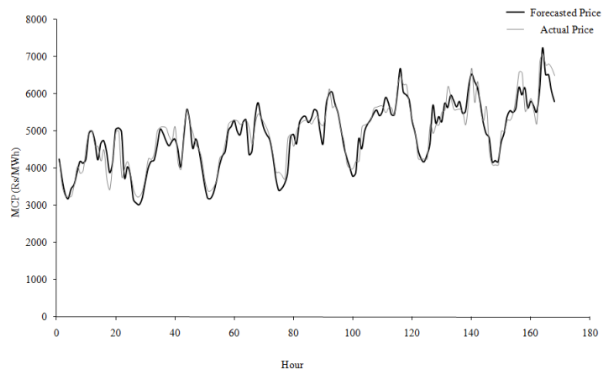


**Figure 3.** Forecasted MCP for a week in September (Indian market in the year 2014)

Therefore, day-ahead energy exchange Indian market, during the year 2014 is utilized as a case study in price forecasting [19]. For correlation, four weeks are chosen, i.e., weeks with especially good price behaviour were purposely not picked. The most unstable prices were utilized for forecasting [9]. It is significant to note that for the very first time, an NN-LM based approach is being tested on the Indian energy market price data

To construct the forecasting model for each of the forecasted weeks, the input data incorporates hourly historical prices of the 42 days prior to the day of the week whose prices are to be predicted. Large training sets are not used to stay away from overtraining amid the learning procedure and when training is attempted for over 42 days, it doesn't give viable and better forecasting accuracy [4, 5, 9]. Also it is tedious for training the NN-LM beyond the 42 days worth of data. Hence, 42 days worth of data set is utilized. The previous 42 days electricity price data are used for training and the following 7 days electricity price are predicted.

For the Indian market the testing data are the week in June (from June 22 to June 28, 2014), the week in September (from September 21 to September 27, 2014), the first week in October (from October 5 to October 11, 2014) and the second week in October (from October 26 to November 1, 2014). The historical data available includes hourly prices from May 11 to June 21, 2014, from August 10 to September 20, 2014, from August 24 to October 4, 2014 and from September 14 to October 25, 2014 and they are used to forecast the price for the above test data [19]. In order to evaluate the forecasting accuracy of the proposed model, the electricity prices in India during the year 2019 and energy exchange of Austria during the year 2015 are considered. For reasonable correlation, the fourth week of June, fourth week of September, first week of October and the second week of October are also selected for the Indian market during 2019.

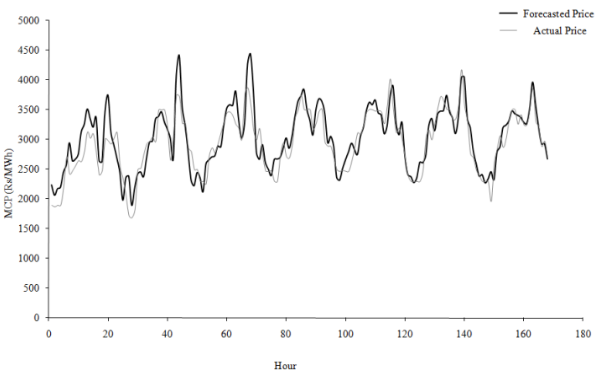


**Figure 4.** Forecasted MCP for first week in October (Indian market in the year 2014)

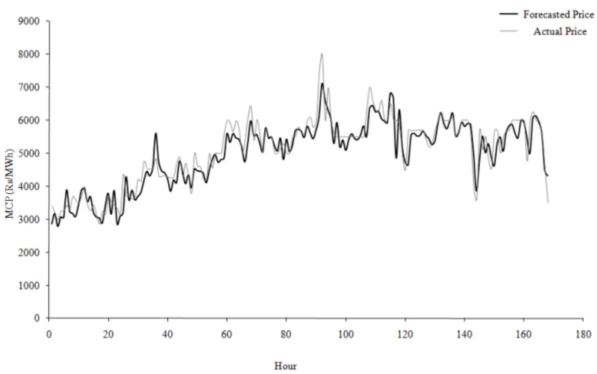


**Figure 5.** Forecasted MCP for second week of October (Indian market in the year 2014)

Real data of the Austrian energy exchange market in the year 2015 are considered in the case study as well. For the sake of fair comparison, the fourth weeks of February, April, July and November (months 2, 4, 7 and 11) are selected, i.e., weeks with particularly good price behaviour are deliberately not chosen. The winter week is from February 15 to February 21, 2015, the historical data available includes hourly prices from January 4, 2015 to February 14, 2015. The spring week is from April 19 to April 25, 2015, historical data available includes hourly prices from March 8 to April 18, 2015. The summer week is from July 19 to July 25, 2015, historical data includes prices from June 7 to July 18, 2015. The fall week is from November 1 to November 7, 2015, historical data includes prices from September 20 to October 31, 2015.

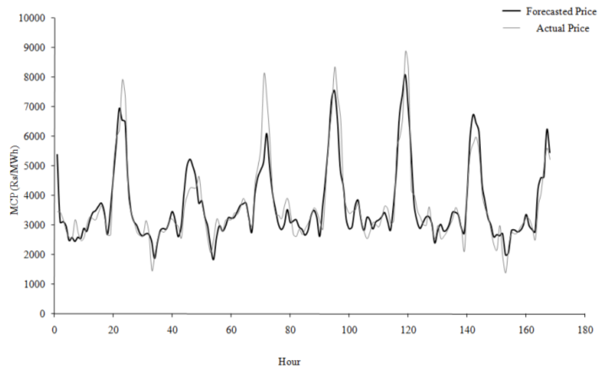


**Figure 6.** Forecasted MCP for a week in June (Indian market in the year 2019)

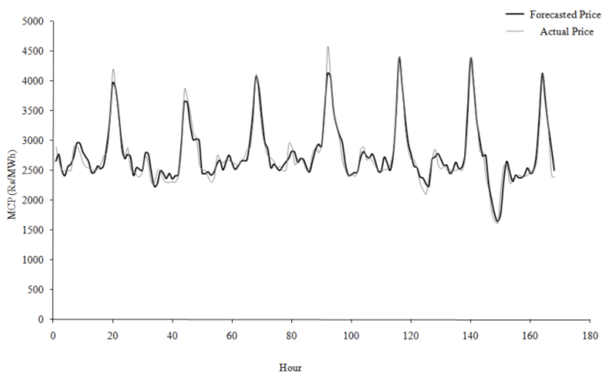


**Figure 7.** Forecasted MCP for week in September (Indian market in the year 2019)

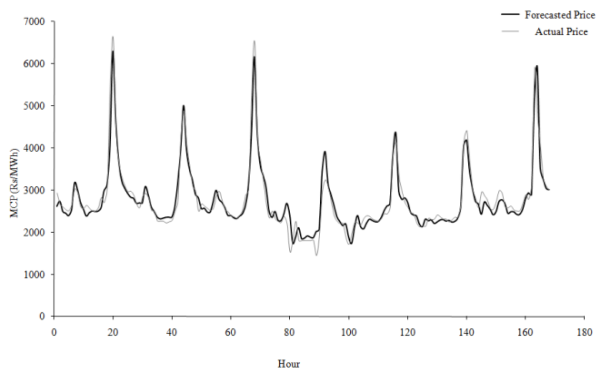


**Figure 8.** Forecasted MCP for first week in October (Indian market in the year 2019)

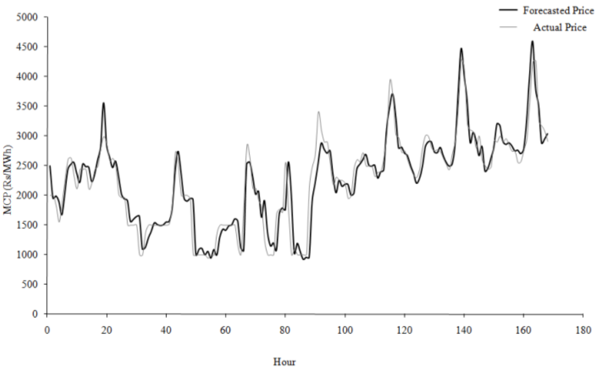


**Figure 9.** Forecasted MCP for second week in October (Indian market in the year 2019)

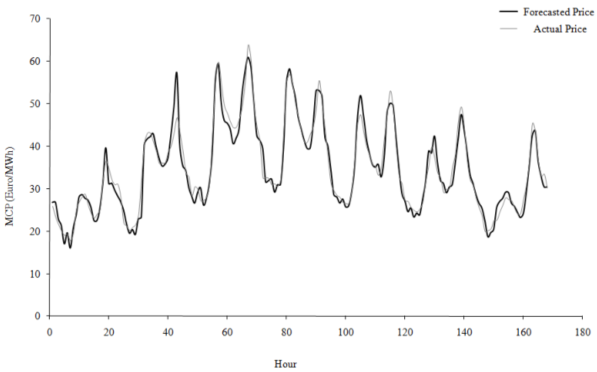


**Figure 10.** Forecasted MCP for a winter week (Austrian market in the year 2015)

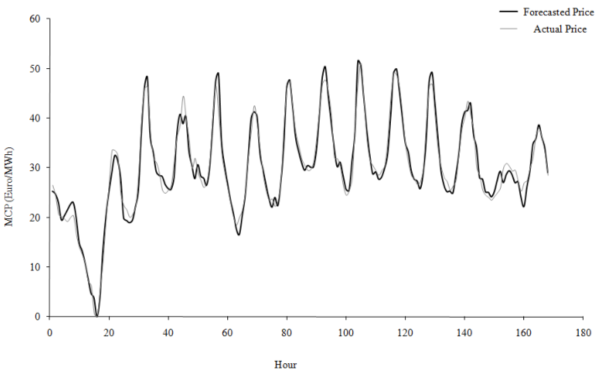


**Figure 11.** Forecasted MCP for a spring week (Austrian market in the year 2015)

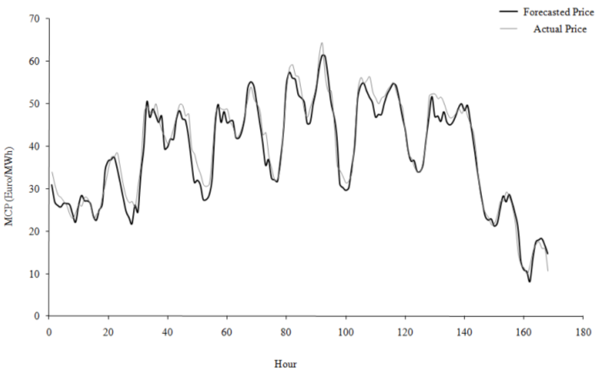


**Figure 12.** Forecasted MCP for a summer week (Austrian market in the year 2015)

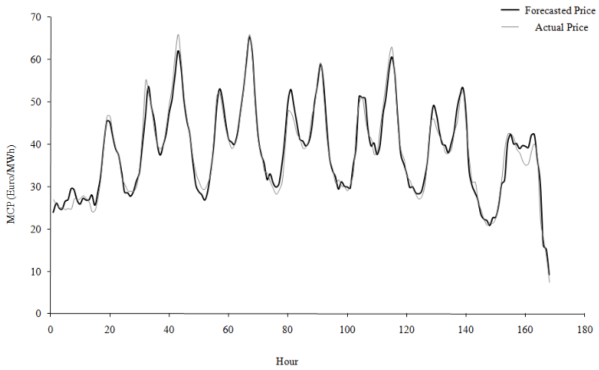


**Figure 13.** Forecasted MCP for a fall week (Austrian market in the year 2015)

The NN use LM training. In this model, the input pattern (selected using correlation analysis) x 1008 (42 days training period x 24 h), and the target pattern is 1 (forecast price) x 1008. The NN model is structured by using the Matrix Laboratory (MATLAB) signal processing and NN toolboxes.

The actual and estimated price values for the four weeks in the Indian market in the years 2014 and 2019, and Austrian energy markets using NN-LM model are shown in Figs. 2–13. Each figure shows the forecasted prices (black line) and the actual prices (gray line) in Rupees per megawatt hour (for the Indian market) and Euro per megawatt hour (for the Austrian market).

## 3.2. Electricity price forecasting with NN-LM model

In NN-LM, the architecture of the neural network is determined using stochastic approach. More and more number of simulations were made until the best number of hidden layers, and their corresponding number of neurons were obtained [4, 8, 9]. The network architecture is typically decided on when both training and testing gives minimal MAPE.

The resultant number of neurons in the input, hidden, and output layers for testing weeks of the Indian and Austria energy exchange markets that produced minimal MAPE error in both training and testing are shown in Table 1.

## 3.3. Comparison with other approaches

Distinctive methodologies are tested for Indian and Austrian energy markets and results of these investigations are discussed in this section. Table 2 shows the statistical analysis and metrics used to assess the accuracy of the proposed NN-LM display in forecasting the electricity prices in both Indian and Austria energy markets. The first column shows the deregulated power market, the second demonstrates the forecast week, the third demonstrates MAPE, the fourth presents the NMSE, and the fifth column shows the EV. It is observed that the MAPE for the Indian power market in the year 2014 has an average estimation of 6.3021% obtained by utilizing the proposed NN-LM model. From Table 2, an average weekly MAPE which is close to 6.3406% for the Indian power market for the year 2019, and Austria for the four weeks of the year 2015 is reported, yielding an average weekly MAPE which is close to 6.1627%, results being obtained by using NN-LM model. It shows the effectiveness of the proposed model for the recent year and under other market environments.

Table 1. The best number of neurons in the input, hidden, and output layers obtained with the NN-LM model for both the markets

| Market | Forecast week | Input layer neurons | Hidden layer neurons | Output layer neurons |
|---|---|---|---|---|
| Indian (2014) | June | 26 | 10 | 1 |
| | September | 33 | 12 | 1 |
| | First October | 14 | 6 | 1 |
| | Second October | 26 | 10 | 1 |
| Indian (2019) | June | 25 | 10 | 1 |
| | September | 30 | 11 | 1 |
| | First October | 17 | 7 | 1 |
| | Second October | 14 | 6 | 1 |
| Austria (2015) | Winter | 31 | 12 | 1 |
| | Spring | 27 | 10 | 1 |
| | Summer | 19 | 8 | 1 |
| | Fall | 24 | 9 | 1 |

Table 3, demonstrates the comparison between the NN-LM model and four different models (ANN, ANN-ANN-PSO, Wavelet-based ANN and Wavelet-based ANN-ANN-PSO). The proposed NN-LM is a single compact and robust architecture (without hybridizing the different hard and soft computing models) tool that consistently performs better than the other models. The four different models with the exception of ANN are a hybrid of soft computing models. As observed from Table 3, the NN-LM has excellent forecast accuracy with less calculation time, making it a single compact and robust model better than hybrid approaches such as ANN, ANN-ANN-PSO, Wavelet-based ANN and Wavelet-based ANN-ANN-PSO. From Table 3, it is also observed that with very low NMSE, and smaller the EV, the proposed model is well-performing both in terms of accuracy and time, and more precise are the prediction of prices.

The aggregate setup time of the proposed technique including the execution of pre-processing (normalization), training of NN (tweaking utilizing experimentation approach), testing of NN and post processing (de-normalization) was around 3 secs on an AMD processor with 2 GHz and 1 GB RAM memory. In the wake of training, average computation (response) time of the NN-LM was around 15 ms (since it just includes the forward propagation of the NN). This approach is an efficient and accurate method for forecasting electricity prices in a deregulated power market. Consequently, the NN-LM presents the best combination of forecasting accuracy and computation time, and furthermore brings down modelling complexity, which is essential for real-time applications. In a deregulated power market, the faster and accurate forecast of prices is likewise essential for real-life applications.

Table 2. Statistical analysis for the four weeks of Indian and Austria energy exchange markets

| Market | Forecast week | MAPE | NMSE | EV |
|---|---|---|---|---|
| Indian (2014) | June | 7.4658 | 1.2468E-07 | 54.6338 |
| | September | 4.5317 | 1.5538E-07 | 20.1299 |
| | First October | 7.1617 | 1.0583E-06 | 50.2738 |
| | Second October | 6.0492 | 1.5091E-07 | 35.8675 |
| Indian (2019) | June | 7.8640 | 1.7044E-07 | 59.3711 |
| | September | 3.8451 | 2.9718E-07 | 14.4918 |
| | First October | 5.6864 | 1.4771E-07 | 31.6965 |
| | Second October | 7.9669 | 2.4363E-07 | 67.3825 |
| Austria (2015) | Winter | 5.1865 | 5.3307E-04 | 26.3695 |
| | Spring | 8.7266 | 4.6843E-04 | 74.8370 |
| | Summer | 5.9292 | 3.0039E-04 | 34.4451 |
| | Fall | 4.8085 | 4.0043E-04 | 22.6633 |

## 3.4. Effect of 4-fold cross-validation on forecasting accuracy

The NN-LM network is taken as an example to illustrate the effectiveness of cross-validation in the proposed training model. In this research, the 4-fold cross-validation is applied to the Indian energy market for the year 2014 for training and testing. The training and testing sets are divided into four disjoint sets of equal sizes. Let each set of data be labelled as week 1 (w1), week 2 (w2), week 3 (w3) and week 4 (w4). The group 1 represents cross-validation w2w3w4 (training) Vs w1 (testing), group 2 represents cross-validation w1w3w4 Vs w2, group 3 represents cross-validation w1w2w4 Vs w3 and finally group 4 represents cross-validation w1w2w3 Vs w4. The statistical results of 4-fold cross-validation are presented in Table 4.

Table 4 shows that 4-fold cross-validation has the most noteworthy accuracy, yielding an average weekly MAPE which is near 5.7717%. Also, the forecast accuracy of week 4 is lower than that of NN-LM, and week 1, week 2, and week3 have higher forecast precision than NN-LM, however, it increases the processing time of the model. Besides, we determined that the NMSE and EV of the test system took a long time by 4-fold cross-validation, and the results are shown in the Table 4. This phenomenon shows that the 4-fold cross-validation strategy can improve the generalization ability of the model.

Table 3. Comparative results between the various methods

| Method | Forecast week | MAPE | NMSE | EV |
|---|---|---|---|---|
| ANN [19] | June | 24.3832 | 7.1227E-07 | 5.8273E+02 |
| | September | 13.0963 | 1.1638E-06 | 1.6811E+02 |
| | First October | 10.1464 | 4.0563E-07 | 1.0091E+02 |
| | Second October | 14.7888 | 5.9369E-06 | 2.1437E+02 |
| Average, MAPE % | | 15.6036 | | |
| ANN -ANN PSO [19] | June | 24.3855 | 6.5888E-07 | 5.8284E+02 |
| | September | 12.9346 | 3.4333E-07 | 1.6398E+02 |
| | First October | 9.9469 | 2.8291E-08 | 9.6982E+01 |
| | Second October | 14.7534 | 2.0740E-07 | 2.1335E+02 |
| Average, MAPE % | | 15.5051 | | |
| Wavelet -based ANN [19] | Winter | 24.3809 | 7.1219E-07 | 5.8262E+02 |
| | Spring | 13.0329 | 1.2132E-06 | 1.6648E+02 |
| | Summer | 10.0097 | 4.1598E-07 | 9.8211E+01 |
| | Fall | 14.1191 | 4.1001E-06 | 1.9539E+02 |
| Average, MAPE % | | 15.3856 | | |
| Wavelet -based ANN-ANN -PSO [19] | Winter | 24.3802 | 6.5859 E-07 | 5.8259E+02 |
| | Spring | 12.8764 | 3.5436E-07 | 1.6251E+02 |
| | Summer | 9.2523 | 2.5696E-08 | 8.3911E+01 |
| | Fall | 14.0773 | 1.4129E-07 | 1.9424E+02 |
| Average, MAPE % | | 15.1465 | | |
| Proposed Method | June | 7.4658 | 1.2468E-07 | 54.6338 |
| | September | 4.5317 | 1.5538E-07 | 20.1299 |
| | First October | 7.1617 | 1.0583E-06 | 50.2738 |
| | Second October | 6.0492 | 1.5091E-07 | 35.8675 |
| Average, MAPE % | | 6.3021 | | |

Table 4. Statistical analysis for the four weeks of Indian energy exchange market in the year 2014 using 4-fold cross-validation

| Market | Forecast week | MAPE | NMSE | EV |
|---|---|---|---|---|
| Indian (2014) | June (w1) | 6.9344 | 1.0922E-07 | 47.1328 |
| | September (w2) | 3.8443 | 1.1309E-07 | 14.4861 |
| | First October (w3) | 5.7044 | 1.3085E-07 | 31.8954 |
| | Second October (w4) | 6.6035 | 9.6907E-07 | 42.7417 |
| | Average, MAPE % | 5.7717 | | |

## 4. Conclusion

An exact determination of electricity price is an essential issue of concern for all energy market players and stock holders, either for creating bidding systems or for settling on investment choices. In the past research, no single accessible model has been applied across data from a broad spectrum of power markets. There is a need to put forth more research attempts in different markets also; this will help in interpreting and understanding the price development and behaviour in various power markets from an advanced point of view. Hence, in this paper, input parameters were selected using correlation analysis of raw data, by removing redundant components and LM training method was implemented which helped the neural network to train better. Forecast results of the benchmark energy market of India for the four weeks of the year 2014 were analyzed, yielding an average weekly MAPE near 6.3021%, with a very low NMSE, and smaller EV.

The NN-LM networks' implementation results illustrate that it has excellent forecasting accuracy than other forecast methodologies, such as ANN, ANN-ANN-PSO, Wavelet-based ANN, and Wavelet-based ANN-ANN-PSO. The technique is very straightforward and yields an average weekly MAPE close to 6.3406% for the Indian power market for the year 2019, and for Austrian energy market for the four weeks of the year 2015. The average weekly MAPE is close to 6.1627%. It shows the effectiveness of the proposed model for the current year and the under different market environments. The 4-fold cross-validation technique can improve the generalization ability of the proposed model. The error of the test weeks has decreased with average weekly MAPE which is near 5.7717%. However, it increases the processing time of the model and improves the accuracy of the test data. In a deregulated power market, the electricity price forecast model resulting in lower computation time and quicker forecast of prices is essential for real-life applications.

## References

[1] YUCEKAYA, A. (2008) *Electric Power Bidding Models for Competitive Markets,* (Doctoral dissertation).

[2] SINGH, S.N. (2008) *Electric Power Generation: Transmission and Distribution,* 2nd ed. (New Delhi: PHI Learning).

[3] LI, G., LIU, C.C., MATTSON, C., LAWARRÉE, J. (2007) Day-ahead electricity price forecasting in a grid environment. *IEEE Transactions on Power Systems* **22**(1): 266-274.

[4] ANBAZHAGAN, S., KUMARAPPAN, N. (2011). Day-ahead price forecasting in Asia's first liberalized electricity market using artificial neural networks. *International Journal of Computational Intelligence Systems*, **4**(4): 476-485.

[5] CATALÃO, J.P.D.S., MARIANO, S.J.P.S., MENDES, V.M.F., FERREIRA, L.A.F.M. (2007) Short-term electricity prices forecasting in a competitive market: A neural network approach. *Electric Power Systems Research* **77**(10): 1297-1304.

[6] Vahidinasab, V., Jadid, S., Kazemi, A. (2008) Day-ahead price forecasting in restructured power systems using artificial neural networks. *Electric Power Systems Research* **78**(8): 1332-1342.
[7] Singhal, D., Swarup, K.S. (2011) Electricity price forecasting using artificial neural networks. *International Journal of Electrical Power & Energy Systems* **33**(3): 550-555.
[8] Anbazhagan, S., Kumarappan, N. (2012) Day-ahead deregulated electricity market price forecasting using recurrent neural network. *IEEE Systems Journal* **7**(4): 866-872.
[9] Anbazhagan, S., Kumarappan, N. (2014). Day-ahead deregulated electricity market price forecasting using neural network input featured by DCT. *Energy Conversion and Management*, **78**: 711-719.
[10] Panapakidis, I.P., Dagoumas, A.S. (2016) Day-ahead electricity price forecasting via the application of artificial neural network based models. *Applied Energy* **172**: 132-151.
[11] Yang, Z., Ce, L., Lian, L. (2017). Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods. *Applied Energy*, **190**: 291-305.
[12] Zhang, Y., Li, C., Li, L. (2018). Wavelet transform and Kernel-based extreme learning machine for electricity price forecasting. *Energy Systems*, **9**(1): 113-134.
[13] Ugurlu, U., Oksuz, I., Tas, O. (2018). Electricity price forecasting using recurrent neural networks. *Energies*, **11**(5): 1255.
[14] Bento, P.M.R., Pombo, J.A.N., Calado, M.R.A., Mariano, S.J.P.S. (2018). A bat optimized neural network and wavelet transform approach for short-term price forecasting. *Applied Energy*, **210**: 88-97.
[15] Singh, N., Hussain, S., Tiwari, S. (2018). A PSO-based ANN model for short-term electricity price forecasting. In *Ambient Communications and Computer Systems*, Singapore, 21 Mar. 2018 (Singapore, Springer), 553-563.
[16] Anbazhagan, S. (2016) Forecasting models for vertically bundled electricity market prices—A review and future trend. *International Journal of Energy and Statistics* **4**(03): 1650011.
[17] Keles, D., Scelle, J., Paraschiv, F., Fichtner, W. (2016) Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. *Applied Energy* **162**: 218-230.
[18] Kabak, M., Tasdemir, T. (2020). Electricity Day-Ahead Market Price Forecasting by Using Artificial Neural Networks: An Application for Turkey. *Arabian Journal for Science and Engineering*, **45**(3): 2317-2326.
[19] Peter, S.E., Raglend, I.J. (2017) Sequential wavelet-ANN with embedded ANN-PSO hybrid electricity price forecasting model for Indian energy exchange. *Neural Computing and Applications* **28**(8): 2277-2292.
[20] Indian Energy Exchange (2020) “Market clearing price”. [Online]. Available: https://www.iexindia.com/. (in India)
[21] Energy Exchange Austria (2020) “Market clearing price”. [Online]. Available: https://www.exaa.at/en. (in Austria)
[22] Bastian, J., Zhu, J., Banunarayanan, V., Mukerji, R. (1999) Forecasting energy prices in a competitive market. *IEEE computer Applications in Power* **12**(3): 40-45.
[23] Nogales, F.J., Contreras, J., Conejo, A.J., Espínola, R. (2002) Forecasting next-day electricity prices by time series models. *IEEE Transactions on Power Systems* **17**(2): 342-348.
[24] Contreras, J., Espinola, R., Nogales, F.J., Conejo, A.J. (2003) ARIMA models to predict next-day electricity prices. *IEEE Transactions on Power Systems* **18**(3): 1014-1020.
[25] Johansson, E.M., Dowla, F.U., Goodman, D.M. (1991). Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems* **2**(04): 291-301.
[26] Hagan, M.T., Menhaj, M.B. (1994). Training feed-forward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks 5*(6), 989-993.
[27] Yu, H., Wilamowski, B.M. (2011). Levenberg-marquardt training. *Industrial electronics handbook* **5**(12): 1.
[28] Azar, A.T. (2013). Fast neural network learning algorithms for medical applications. *Neural Computing and Applications* **23**(3-4): 1019-1034.