# A Comparative Study of Different Machine Learning Methods for Electricity Prices Forecasting of an Electricity Market

Elham Foruzan, *Member, IEEE*, Stephen D. Scott, and Jeremy Lin, *Senior Member, IEEE*

***Abstract*—Generally, it is difficult to accurately forecast electricity prices because they are unpredictable. Yet, accurate price forecasting is expected to provide crucial information, needed by power producers and consumers to bid strategically, thereby decreasing their risks and increasing their profits in the electricity market. In this paper, two models using artificial neural networks (ANN) and support vector machines (SVM) were developed for electricity price forecasting. In addition, ant colony optimization (ACO) was used to reduce the feature space and give the best attribute subset for ANN model. Using ACO for feature selection significantly reduced the training time for ANN-based electricity price forecasting model while the results were almost as accurate as those from ANN model.**

***Keywords*—Electricity price forecasting, Artificial Neural Networks, Support Vector Machines, Machine Learning, Ant Colony Optimization**

## I. INTRODUCTION

THE restructuring of the electric power industry has brought about the change to the traditional structure of vertically integrated utilities. The generation, transmission, and distribution sectors that were once controlled by a single utility before this change were separated and are now owned by different entities after the restructuring. The competition exposed to the generating companies in the electricity market increased the risks for these and other entities in the electricity industry. Since a single company no longer holds the entire supply chain, risk management has become an important aspect of the electricity business.

To manage such risks in the electricity market, forecasting of different market indicators such as the hourly electricity price of the spot market has become necessary. Accurate forecast of the market prices is an important input to the decision-making activities of a generating company for producing energy. Electricity is a special commodity because it is extremely expensive to store electricity. Thus, almost all generated electricity must be consumed instantaneously. Therefore, both producers and consumers need accurate price forecasts in order to establish their own strategies for increasing their own surpluses.

The electricity price is generally influenced by many factors such as hour of the day, day of the week, historical electricity price, and load demand. Forecasting problems for electricity demand and price have been previously solved by using various techniques such as fuzzy inference, fuzzy-neural models, artificial neural networks (ANN) and support vector machines (SVM). Among the different forecasting techniques, the application of ANNs and SVMs for forecasting in power systems has received much greater attention in recent years [1, 2, 3, 4, 5]. The main reason for the popularity of these models is the ANN's ability to learn complex nonlinear relationships that are difficult to model with conventional techniques.

A new model for short-term electricity price forecasting was developed by *Fan et al.* [6] by combining two machine learning technologies known as Bayesian clustering by dynamics (BCD) and SVM. The model works in two distinct steps. In the first step, a BCD classifier is applied to cluster the input data set into several subsets in an unsupervised manner. Then, groups of 24 SVMs for the next day's electricity price profile are used to fit the training data of each subset in a supervised way.

A simple methodology based on the weighted nearest neighbors (WNN) technique was used by Lora rt al. [7] to forecast hourly prices in deregulated electricity markets. Forecast results corresponding to the Spanish market for the entire year 2002 are reported. Results from the proposed method are also compared with that of other techniques such as ANN, neuro-fuzzy systems, GARCH, and ARIMA (with and without wavelet transform). An adaptive wavelet neural network (AWNN) is proposed by Pindoriya et al. [8] for short-term price forecasting in the electricity markets. A Mexican hat wavelet was chosen as the activation function for hidden-layer neurons of feed-forward neural networks (FFNNs). Day-ahead prediction of market clearing price (MCP) of Spanish market, and locational marginal price (LMP) forecasting in PJM electricity market, were considered for the method's applications. The authors claimed that their proposed method can produce good forecast compared with other existing forecasting techniques.

Zhao et al. [9] proposed a novel data mining-based approach to accurately forecast both the value and prediction interval of the electricity price series. In their proposed approach, an SVM was used to forecast price values. To forecast the prediction interval, authors proposed a statistical model in which a heteroscedastic variance equation was introduced for the SVM. Maximum likelihood estimation (MLE) is used to estimate model parameters.

Ant colony optimization (ACO) along with SVM was used by Niu et al. [10] in load forecasting. The method of colony optimization was employed to reduce a large amount of SVM training data and speed up the processing time. The ACO method also improved the accuracy of data mining. The authors claimed that this new method can achieve greater forecasting accuracy in short-term load forecasting, compared with a single SVM and a back-propagation (BP) neural network.

A fast method of forecasting electricity market prices was proposed in [11] based on a recently emerged learning method for single hidden layer feed-forward neural networks, known as the extreme learning machine (ELM). The new approach has also improved price intervals forecast accuracy by incorporating a bootstrapping method for uncertainty estimations. Authors used case studies based on chaos time series and Australian National Electricity Market price series. The authors claimed that their methods can capture the nonlinearity from the highly volatile price data series with much less computation time compared with other methods.

In this paper, three intelligent models, namely, SVM, ANN, and ACO-ANN for electricity price forecasting were introduced and implemented using MATLAB. The ant colony optimization algorithm is applied to find optimal feature subsets. These three models were trained and tested on data that contains hourly locational marginal prices (LMPs) for a one-year period. Results showed that the proposed ANN method has a lower error rate compared with the prediction methods used in [8]. The proposed ACO-ANN method for electricity price prediction reduced input data set intelligently without sacrificing result accuracy. SVM implementation for electricity price forecasting shows reasonably accurate results with very fast performance compare to ANN. Therefore these methods are practical and useful. The data set used for forecasting electricity prices is derived from New York ISO (NYISO), which is available on its website [12]. This paper is organized as follows. In Section 2 , we discuss forecasting methods, namely SVM, ANN and ACO-ANN, for electricity prices. Features selection and comparison methods are presented in section 3. Results are presented and discussed in section 4. Finally, conclusions are drawn in section 5.

## II. Intelligence-Based Forecasting Methods

The basic concepts of three intelligence-based forecasting methods that were used in this paper are introduced bellow.

### A. Support Vector Machine (SVM)

The regression SVM problem can be stated as follows [13]. Given the training data set $(x_i, r_i)$ where $X_i$ denotes the space of the input patterns (e.g. $X = R^d$ for $i = 1, ..., n$; the goal is to find function y that has at most $\epsilon$ deviation from the actually obtained targets $r_i$ for all the training data. A data set that is not linearly separable in a single-dimensional space may be linearly separable in a higher-dimensional space. Therefore one method is to convert the data set into the higher dimension and find the SVM linear function. Using function $g_j(x), j = 1, ..., m$, input $x$ is mapped into the $m$-dimensional feature space to find the SVM function $y(x)$ as a linear function of inputs in the new space is shown as:

$$y(x, w) = \sum_{j=1}^{m} w_j g_j(x) + b \tag{1}$$

where $w_j$ is the weight of input $g_j(x)$ and $b$ is the bias term. SVM regression solves the problem to estimate the parameters $w_j, j = 1, ..., m$ and bias term. In the SVM, $\epsilon$-insensitive loss function is considered as an error.

$$e_\epsilon(r, y(x,w)) = \begin{cases} 0, & \text{if} \quad |r - y(x,w)| \le \epsilon \\ |r - y(x,w)| - \epsilon, & \text{otherwise} \end{cases} \tag{2}$$

Thus SVM regression is formulated to minimize the error function and problem complexity:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \\ \text{s.t.} & \begin{cases} r_i - y(x_i, w) \le \epsilon + \xi_i^* \\ y(x_i, w) - r_i \le \epsilon + \xi_i \\ \xi_i, \xi_i^* \ge, i = 1, ..., n \end{cases} \end{aligned} \tag{3}$$

where $\xi_i^*$ and $\xi_i$ are the upper and lower training error respectively. in most cases the optimization problem (3) can be solved more easily in its dual formulation. Therefore optimization problem in eq. (3) can be transformed into the dual problem, and its solution is given by maximizing the dual function as shown below:

$$\begin{aligned} y(x) = -\frac{1}{2} \sum_t \sum_s (\alpha_+^t - \alpha_-^t)(\alpha_+^s - \alpha_-^s) K(x_i, x) \\ -\epsilon \sum_s (\alpha_+^t + \alpha_-^t) - \sum_s r^t (\alpha_+^t - \alpha_-^t) \\ \text{s.t.} \quad 0 \le \alpha_+^t \le C, \\ 0 \le \alpha_-^t \le C, \\ \sum_t (\alpha_+^t - \alpha_-^t) = 0 \end{aligned} \tag{4}$$

where

$$K(x, x_i) = \sum_{j=1}^{m} g_j(x) g_j(x_i) \tag{5}$$

Parameter $C$ determines the tradeoff between the model complexity and the degree of acceptable error. Increasing the value of $C$ will increase the effect of minimizing error.

As can be seen from equation (4), we need to solve an optimization problem that computes the pair-wise dot product of two training instants. A *Kernel* is a function that computes a dot product between two inputs as a measure of their similarities. Using the kernel concept, we can solve the pair-wise dot product of instants in the higher dimension without explicitly transforming them to the higher dimension. The common kernel function that is applied in this work is RBF kernel which is shown below as:

$$K(x_i, x) = \exp\big( - \frac{\| x - x_i \|^2}{2p^2} \big) \tag{6}$$

By solving equation 3, the coefficients will be calculated. Finally, the function $y(x)$ can be written as a weighted sum of the support vectors.

$$y(x) = \sum_{i=1}^{n_{sv}} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{7}$$

### B. Artificial Neural Network (ANN)

Neural networks have a considerable ability to obtain meaning from complicated data. They can be used for extracting and detecting the patterns from trends that are too complex to be noticed by humans. An artificial neural network (ANN) is a structure of layers and inter-connected processing nodes. Multi-layer perceptrons (MLPs) are the most widely used network architecture in artificial neural networks.

Neural networks for supervised learning showed good results in forecasting tasks [13]. In this study, we use classic ANN, which uses feed forward and back-propagation (BP) algorithms to find the best set of network coefficients.

### C. Features for Price Forecasting

A major concern of an electric power utility in the deregulated power market is to accurately forecast the electricity price. Various factors influence electricity price. On one side, generation capacity and cost are changing due to the exogenous variables such as technical limitation and oil price. On the other side, energy demand is variant and depends on the time of the day, day of the week, season and weather conditions.

To build an acceptable price forecasting model, choosing appropriate input variables is one of the most important tasks. Generally, historical prices and system load data are two of the most influential factors in the price forecasting. Historical price and demand data could be found in ISO websites, for each hour [12]. Using this data, in Table 1, we tried to capture the volatility of price by using an extensive set of hourly electricity prices for one year.

TABLE I. ATTRIBUTES SELECTED FOR ELECTRICITY PRICE FORECASTING

| Attribute Number | Electricity Price Attributes |
|---|---|
| 1 | Day of the week (1,2,...,7) |
| 2 | Hour of the day (1,2,...,24) |
| 3 | Forecast demand |
| 4 | Change of demand to one hour ago |
| 5-13 | Price: 23, 24, 25, 167, 168, 336, 504, 672 hours ago |
| 14-20 | Temperature: Maximum 1,2,3,4,5,6,7 days ago |
| 21-27 | Temperature: Minimum 1,2,3,4,5,6,7 days ago |
| 28-34 | Temperature: Average 1,2,3,4,5,6,7 days ago |
| 35 | Humidity |

### D. Ant Colony Optimization (ACO)

In many ant species, when ants are shuttling between their nest and food, they deposit a substance called *pheromone* on the ground. Other ants perceive the presence of pheromone and tend to follow paths where pheromone concentration is higher. Through this mechanism, named ant colony optimization (ACO), ants are able to find a shortest path from the nest to the food [14].

Since the number of features in the price forecasting task is high, it would be helpful in utilizing ACO method to select the most helpful features among thirty-five features that were selected for price forecasting. Appropriate feature selection will help reduce the computational time and simultaneously keep prediction error small.

In order to utilize ACO, at first, the problem should be represented as a set of nodes and edges between the nodes. In this paper, each node was assigned to one feature. The procedure for selecting the subset of features is summarized as follows.

In the first step, each ant selects its own path according to two indices. The first index is the amount of pheromone in each path. Pheromone matrix is an $n \times n$ matrix in which $n$ is the number of features. The entry $\tau_{ij}$ of this matrix is the amount of pheromone between two edges $i$ and $j$. The pheromone matrix is dynamically changing during the optimization process.

The second index for selecting the next feature is the heuristic desirability of each path. The heuristic desirability for each path is a constant number during optimization process and shows the amount of desirability for selecting the next feature according to the prior information. The matrix of heuristic desirability is an $n \times n$ matrix in which the entry $\eta_{ij}$ of this matrix is the heuristic desirability for path $i$ to $j$. Equation 8 shows how these two indices contribute for choosing the next feature:

$$p_{mn}^k(t) = \begin{cases} \frac{[\tau_{mn}(t)]^\alpha [\eta_{mn}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{mk}(t)]^\alpha [\eta_{mk}]^\beta} & \text{if} \quad j \in \text{allowed} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $\alpha$ and $\beta$ are two constant numbers. After all ants complete their paths, pheromone updating rules will be applied. Their effect is to dynamically change the desirability of paths according to following rules:

$$\tau_{mn}(t+n) = (1-\rho)\tau_{mn}(t) + \Delta\tau_{mn} \tag{9}$$

$$\Delta\tau_{mn} = \sum_{k=1}^{n} \Delta\tau_{mn}^k \tag{10}$$

$$\Delta\tau_{mn}^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ uses path } (i,j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $\rho$ is the pheromone evaporation factor. The first element of pheromone updating causes the features in one ant's tour to be chosen with lower probability than in the next ants tour.

Consequently, ants will favor the exploration of features which have not been visited yet. $\Delta\tau_{mn}$ is the quantity that depends on the total distance that ant $k$ passed during its tour, $L_k$ , if ants $k$ uses edge $(i, j)$ in its path. $Q$ is a constant during the optimization.

## III. Features selection and Comparison Method

### A. Feature Selection for ANN Model Using ACO

The combination of ANN and ACO pseudo-code is shown in Fig. 1. In this implementation of ACO-ANN, the best subset of attributes is the smallest subset in which the error of training data set is smaller than the predefined stopping criterion. In the first step of the implementation, ACO runs to find the most important features for electricity price forecasting among the whole feature space. In step 2 of implementation, ANN trains with the features that have been selected with ACO. This algorithm repeats until the feature subset obtained from ACO gives acceptable error on the training data set.

During the ACO simulation, each element $(i, j)$ of the heuristic desirability matrix is the weighted sum of correlation between feature $i$ and $j$ and correlation of feature $j$ and actual price:

$$\eta = \alpha M_1 + \sigma M_2 \tag{12}$$

where $M_1$ is the matrix of correlation factors between every pair of attributes and $M_2$ is matrix of correlation factors between every attribute and actual label. $\alpha$ and $\sigma$ are constant coefficients.

### B. Performance Evaluation and Comparison Method

The performance of each model is evaluated using root-mean-square error criteria as follows.

$$\text{Error} = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}(r_i - y_i)^2\right)} \tag{13}$$

where $r_i$ and $y_i$ are the actual and forecasted price for sample $i$, respectively.

The comparison of three models ACO-ANN. ANN and SVM were performed based on $k$-fold cross validation with $k$ = 11. Therefore the data set was divided into 11 subsets, and each solution method was repeated 11 times. Each time, one of the 11 subsets is used as the test set and the other 10 subsets are put together to form a training set. And the paired t-test were also performed to compare the accuracy of the results of each method.

## IV. Results and Discussions

In order to test the performance of the discussed methods, hourly electricity price data for one specific node of the network is downloaded from the New York ISO website [12]. Note that the data is collected by means of sensory network and it is possible to have invalid price records.

In this study, we examined two approaches to predict the hourly price. In the first approach, we fed a matrix of 3000

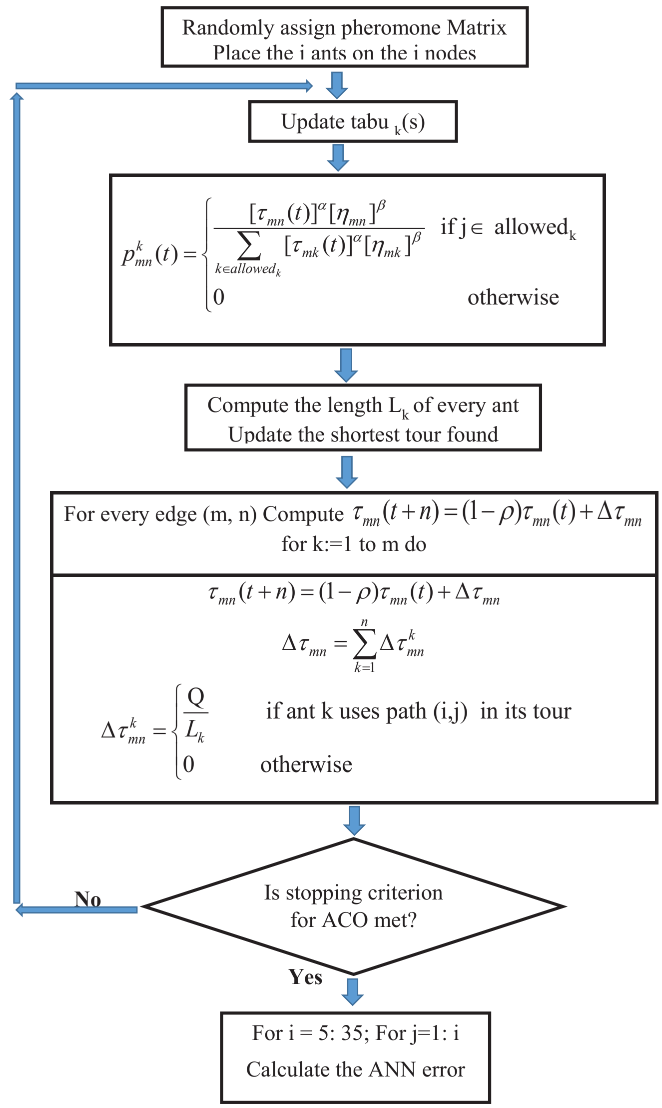


Fig. 1. Algorithm for ACO-ANN model

instances to the prediction network. In the second approach, we collected the data for each hour in 300 days and constructed 24 data sets respectively. Then we used the data set pertaining to each hour to find the forecast price for the corresponding hour of the day.

In all model implementations, input data were first normalized between zero and one. This helps to obtain a better model, since the range of attribute values are different.

A support vector machine implementation in Matlab was used to build the models for hourly electricity price forecasting. Quadratic programming optimization was used to minimize Lagrange multipliers of the dual problem. A Gaussian kernel with $p^2 = 0.5$ was chosen in this implementation. Other parameters were chosen as: $C = 400$ and $\epsilon = 0.25\text{E-}8$. The 24-hour ahead actual and forecasted price using SVM implementation is shown in Fig. 2. Error between actual and prediction price for 24-hour ahead using SVM implementation was 6.4%. It has been observed from Fig. 2 that the predicted price followed the actual price very well. However, at time $t$=19 h, the error between predicted and actual price reached its maximum and decreased afterward. This maximum error has remarkably increased average error for twenty four hours.

In the ANN model, we selected a 3 layer network. Also, Sigmoid function was used in this implementation. The number of nodes in input layer, hidden layer and output layer are 35, 15 and 1 respectively. The error is calculated according to Equation 5.1. The 24-hour actual and forecasting price using ANN implementation is shown in Fig 3. As can be seen in Fig. 3, the predicted price followed the actual price very well. Calculated error value for this 24 hours is slightly less than 4%. The proposed ANN method was applied on the data set for PJM market. ANN method has 3.5% error for this data set. Compared to [8], the electricity price prediction error rate was decreased from 4.5% to a lower value.

In the other implementation, The ACO optimization was used. According to ACO, vector A was selected as the optimal set of attributes. Therefore, instead of using all thirty-five attributes, the attributes of matrix A are used as the inputs of ANN model. Each element of A is the designation number of selected attributes according to Table 1: A = [7 6 5 8 9 10 11 12 13 16 14 15 18 19 17].

Next, the ANN algorithm is used to make a prediction with sigmoid function. In this implementation, one hidden layer is used. The number of nodes in the input, hidden and output layer were 15, 15 and 1 respectively.

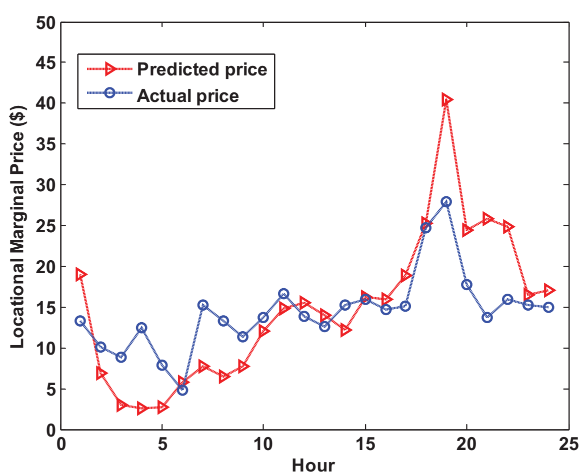


Fig. 2. Forecast and actual electricity prices for day-ahead using SVM

The 24-hour ahead actual and forecasting price using ACO-ANN implementation is shown in Fig. 4. From the figure, electricity price prediction with the ACO-ANN algorithm had almost the same pattern as ANN price prediction algorithm. However, its error rate is slightly bigger. The error for this test data set is 4.2%.

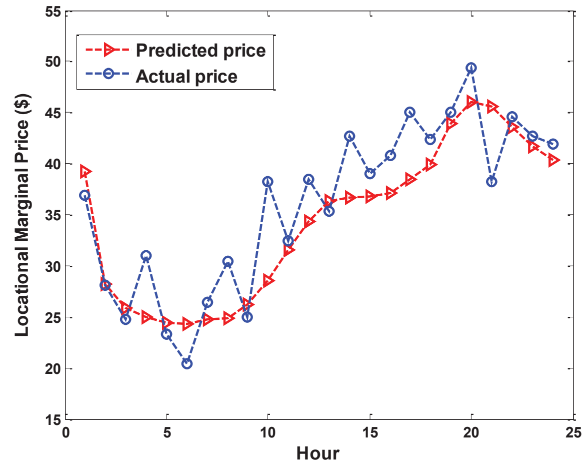


Fig. 3. Forecast and actual electricity prices for day-ahead using ANN

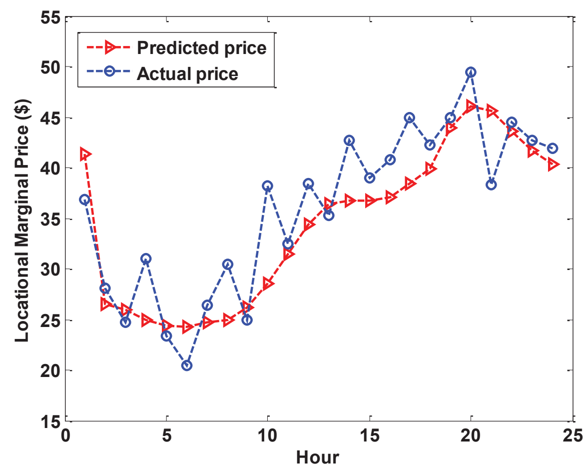


Fig. 4. Forecast and actual electricity prices for day-ahead using ACO-ANN

Hourly electricity price forecasting are simulated for two models, namely: ACO-ANN and SVM. In the hourly prediction scheme, we adopted 24 networks for 24 hours of the day and we used subset of 300 examples pertaining to each hour of the day, to train the network. Comparing to the previous results of SVM and ACO-ANN, hourly predictions have higher error. The average error rates for twenty-four hour are 12% and 15% for ANN and SVM respectively.

Using our second approach, we hoped that the prediction error decreases. Surprisingly the results show a smaller accuracy compare to cumulative version. One reason might be the reduced size of the training data set. In the first case, we used all 3000 examples for training one network and employed the resultant network to predict the price. But in hourly prediction scheme, sample set is decreased to 270 examples. It is worth mentioning that using the second approach for price forecasting significantly decreased the training time.

Training times in the first approach were 100, 35 and 20 seconds for ANN, ACO-ANN and SVM respectively. However, in the second approach training time decreased to 5 and 10 second for SVM and ACO-SVM implementations.

### A. T-Test Comparison

In order to perform statistical comparisons according to the t-test procedure, 3300 samples were randomly selected from the sample space. According to the student's t-square test, $k$-fold cross validation with $k = 11$ was performed on three models. The accuracy results of cross validation on the test data sets for three models are shown in Table 2.

TABLE II. Error percentage of each method in each iteration

| Iterations | SVM | ACO-ANN | ANN |
|---|---|---|---|
| 1 | 14.08 | 10.29 | 9.16 |
| 2 | 15.30 | 15.00 | 10.48 |
| 3 | 15.45 | 11.13 | 9.40 |
| 4 | 12.72 | 10.09 | 8.93 |
| 5 | 11.63 | 7.39 | 7.65 |
| 6 | 14.99 | 8.51 | 8.48 |
| 7 | 10.63 | 9.62 | 9.78 |
| 8 | 12.03 | 6.95 | 3.38 |
| 9 | 13.99 | 8.17 | 8.40 |
| 10 | 15.72 | 6.52 | 6.83 |
| 11 | 12.72 | 10.43 | 9.97 |
| Average | 12.43 | 9.4 | 8.4 |

With comparing SVM with ACO-ANN based on the $k$-fold cross validation, the true difference of expected error between SVM and ACO-ANN for this implementation and this specific data set is significant at a level of p less than 0.01. In other words, the ACO-ANN has less error than SVM with confidence of 99%.

The error rates for the ACO-ANN and ANN are almost the same. Therefore ant colony optimization was helpful in choosing the best subset of attributes among all attribute candidates for electricity price forecasting without significantly affecting the error rate.

Results show that the specific implementation of SVM that is performed in our study has a larger error rate among other implemented models. However, it has the shortest training time on the electricity price forecasting.

## V. Conclusions

In this paper, three models for electricity price forecasting were introduced and implemented using MATLAB. These three models were trained and tested on the data set that contains one year hourly electricity prices. Based on the results of the cross validation, we can claim that SVM had the largest error rate. However, SVM has the smallest training time among these three algorithms. The difference in accuracy between ACO-ANN and ANN is very small. But ACO-ANN uses less complicated network according to the small number of attributes selected by ACO. Consequently, the training time is reduced. Our results showed that the combination of ANN with ACO outperformed the ANN without the ACO. The two SVM and ACO-ANN algorithms are useful and practical in real applications.

## VI. Acknowledgement

The authors would like to thank Hossein Rabiee for helpfull discussions [15 , 16].

## References

[1] S. K. Aggarwal, L. M. Saini, and A. Kumar. "Electricity price forecasting in deregulated markets: A review and evaluation," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 1. pp. 13-22, 2009.

[2] L. M. Saini, S. K. Aggarwal, A. Kumar, "Parameter optimization using genetic algorithm for support vector machine-based price-forecasting model in national electricity market," *Generation, Transmission & Distribution, IET*, vol. 4, no. 1, pp. 36-49, January 2010.

[3] T. Soares, F. Fernandes, H. Morais, P. Faria, Z. Vale, "ANN-based LMP forecasting in a distribution network with large penetration of DG," *Transmission and Distribution Conference and Exposition (T&D)*, vol. 1, no. 8, pp. 7-10, May 2012.

[4] D. Singhal, and K. S. Swarup, "Electricity price forecasting using artificial neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 33. no. 3. pp. 550-555, 2011.

[5] M. Negnevitsky, P. Mandal, and A. K. Srivastava, "Machine learning applications for load, price and wind power prediction in power systems," *15th International Conference on Intelligent System Applications to Power Systems, ISAP'09*, 2009.

[6] S. Fan, J. R. Liao, K. Kaneko, and L. Chen, "An integrated machine learning model for day-ahead electricity price forecasting," *Power Systems Conference and Exposition, PSCE '06*, 2006.

[7] A. T. Lora, J. M. R. Santos, A. G. Expósito, J. L. M. Ramos, and J. C. R. Santos, "Electricity market price forecasting based on weighted nearest neighbors techniques," *IEEE Trans. Power Systems*, vol. 22, no. 3, pp. 1294-1301, Aug 2007.

[8] N. M. Pindoriya, S. N. Singh, and S. K. Singh, "An adaptive wavelet neural network-based energy price forecasting in electricity markets," *IEEE Trans. Power Systems*, vol. 23, no. 3, pp. 1423-1432, Aug 2008.

[9] J. H. Zhao, Z. Y. Dong, Z. Xu, and K. P. Wong, "A statistical approach for interval forecasting of the electricity price," *IEEE Trans. Power Systems*, vol. 23, no. 2, pp. 267-276, May 2008.

[10] D. Niu, Y. Wang, and D. D. Wu, "Power load forecasting using support vector machine and ant colony optimization," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2531-2539, 2010.

[11] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong, and H. W. Ngan, "Electricity price forecasting with extreme learning machine and bootstrapping," *IEEE Trans. Power Systems*, vol. 27, no. 4, pp. 2055-2062, Nov 2012.

[12] http://www.nyiso.com/public/markets_operations/market_data/pricing_data/index.jsp

[13] E. Alpaydin, "Introduction to Machine Learning," MIT press, 2004.

[14] M. Dorigo, M. Vittorio, and A. Colorni. "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1 pp. 29-41, 1996.

[15] M. Rostami, R. Faez, and H. R. Golgir, "Magnetization of bilayer graphene with interplay between monovacancy in each layer," *Journal of Applied Physics*, vol. 114, no. 8, p. 084313, 2013.

[16] H. R. Golgir, R. Faez, M. Pazoki, H. Karamitaheri, and R. Sarvari, "Investigation of quantum conductance in semiconductor single-wall carbon nanotubes: Effect of strain and impurity," *Journal of Applied Physics*, vol. 110, no. 6, p. 064320, 2011.