International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 8689-8695

Cloud Computing for Short-Term Load Forecasting Based on Machine Learning Technique

Tamer A. Farrag[1], Ehab. E. Elattar[2,3]

[1]*Computers Engineering Department, Misr Higher Institute of Engineering and Technology, Egypt.*
[2]*Department of Electrical Engineering, Faculty of Engineering, Taif University, KSA.*
[3]*Department of Electrical Engineering, Faculty of Engineering, Menofia University, Egypt.*

Orcid IDs : [1]0000-0003-1976-138X, [2]0000-0002-3966-2584

Abstract

Short-term electric load forecasting (STLF) plays the main role in making operational decisions in any electrical power system. The implementation of forecasting algorithms collides with the high computational power needed to perform the complex perdition processes on large datasets. In this paper, a cloud-based STLF algorithm is implemented. The performance analysis of the proposed system was compared against the implementation of the same algorithm on a local machine and against many other forecasting algorithms. The results show that the cloud-based implementation enhances the algorithm execution time.

Keywords: Load Forecasting; SVR; Cloud Computing; Azure ML

INTRODUCTION

Electric load forecasting has always been a vital part of an efficient power system planning and operation with the introduction of deregulation into power industry. The fundamental objective of the electric power industry is to maximize the efficiency of electricity generation and consumption and reduce energy prices [1].

Short-term electric load forecasting (STLF) is imperative for making operational decisions in electric power systems, such as unit commitment, reducing spinning reserve, economic dispatch, reliability, analysis, and maintenance scheduling [2]. STLF deals with load forecasting from one hour up to a few days. Recently, accurate STLF has gained more importance and greater challenges in deregulation of electricity markets.

STLF is still a challenge because of its high complexity. A number of models were proposed during the past few decades. These can be classified as either traditional or machine learning based techniques. The former include time series predictors such as autoregressive moving average exogenous variable (ARMAX) model[3]. These methods are based on a linear regression model and cannot always represent the nonlinear characteristics of complex loads[4]. Different machine learning techniques were used for STLF, such as artificial neural networks (ANNs) [5], radial basis function (RBF) [6], fuzzy-neural models [7] and support vector regression[8] .

The forecasting process depends on analyzing the historical data of electricity demands in a particular country or region, and many other factors can be considered such as weather forecasting and commercial plans. Therefore, the entire data history is needed to train the prediction model, but such an approach has the disadvantage that if new information is taken into consideration then all the parameters of the model may need to be retrained, and also a long parameter re-estimation stage is required. In addition, this huge amount of data and the expected complex forecasting process lead to the need for massive computational power. The researchers tried to find approximation methods to minimize this amount of data and to minimize the needed computational power. They try to exclude some factors, to apply sampling techniques and so on. Many approaches solved these backdraws. One of them employed the local predictors as in[9][10]. In this work, we will benefit from Cloud Computing (CC), the new revolution in the field of computing systems.

Economic constraints play the main role in any applied research. The appearing of CC paradigm solves many economic issues that face researchers and developers. Before CC, the supercomputer was the only appropriate choice to get a huge computational power which was a very expensive choice. There are many computing systems such as disturbed systems, grid computing, internet computing, quantum computing …etc., but CC is the most applicable commercial system. CC works according to three service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Nowadays, many leading companies provide CC services to public, and they attract a huge number of clients. The most important CC services providers list includes Microsoft Azure[11], Amazon Web Service (AWS)[12] and Google App Engine[13].

The rest of this paper is organized as follow: Section 0, the problem definition and proposed solution are explained. In Section 0, the design of experiments is presented. Experiments results and discussion about them is introduced in section 0. Finally, the paper is concluded in section 0.

SHORT TERM LOAD FORECASTING BASED ON SUPPORT VECTOR REGRESSION

Nowadays, there is no electric utility that can operate in an economical, secure and reliable mode without STLF. Forecasting tools with higher accuracy are essential to get lower operating costs. Overprediction of STLF wastes resources since more reserves are available than needed and, in turn, increases the operating cost. On the other hand, underprediction of STLF leads to a failure to provide the necessary reserves which also increases the operating cost [15].

In general, the electrical load is composed of different consumption units. Various factors affect the changes of this electrical load. These factors can be classified as weather, calendar, economical, and random factors.

STLF can be considered as a multivariable prediction problem. It can be solved as a function of regression problem. The next day load is the output of regression model, while the historical load data and its influencing factors are the input data of regression model. History database supplies the training data. The final goal of this regression problem is to find a mapping function from historical load data and its influencing factors to the forecasted load with good generalization ability. The historical load data is divided into two different data sets. The first one is the training data set which is used to train the regression model, while the other one is the testing data set which is used to evaluate the trained regression model [16].

Support vector regression (SVR)[17] [18] which proposed based on statistical learning theory (SLT), has been investigated as a promising approach to power load forecasting [19]. Its advantages mainly come from the employing of structural risk minimization principle as an alternative of empirical risk minimization principle Consequently, it can obtain an optimal global solution by solving a quadratic problem.

There are two main features in the execution of SVR. They are quadratic programming and kernel functions.A quadratic programming problem will be solved with linear constraints to get the SVR's parameters. The flexibility of kernel functions lets the technique to search a broad range of the solution space [18]. The primary objective of SVR is to map the data x into a high dimensional feature space via a nonlinear mapping, and perform a linear regression in that feature space as [17] [18]:

$$f(x) = \langle w, x \rangle + b \tag{1}$$

Where, $\langle ., .\rangle$ represents the dot product, w contains the coefficients that have to be estimated from the data and b is a real constant. Using Vapink's ε–insensitive loss function [18], the overall optimization is formulated as:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2} w^T w + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)$$
$$\text{subject to} \begin{cases} y_i - \left(w^T\phi\left(x_i\right) + b\right) \le \varepsilon + \xi_i^* \\ \left(w^T\phi\left(x_i\right) + b\right) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, \qquad i = 1,\ldots,N \end{cases} \tag{2}$$

where x_i is mapped to higher dimensional space by the function ϕ, ε is a real constant, ξ_i and ξ_i^* are slack variables subject to ε-insensitive zone and the regularization parameter (C) determines the trade-off between the flatness of f and training errors.

Introducing Lagrange multipliers α_i and α_i^* with $\alpha_i\alpha_i^* = 0$ and $\alpha_i, \alpha_i^* = 0$ for $i=1,\ldots, N$, and according to the Karush-Kuhn-Tucker optimality conditions [18], the SVR training procedure amounts to solving the convex quadratic problem:

$$\min_{\alpha,\alpha^*} \frac{1}{2}\sum_{i,j=1}^{N}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)K\left(x_i, x_j\right) + \varepsilon\sum_{i=1}^{N}\left(\alpha_i + \alpha_i^*\right) - \sum_{i=1}^{N} y_i\left(\alpha_i - \alpha_i^*\right)$$
$$\text{subject to} \begin{cases} 0 \le \alpha_i, \alpha_i^* \le C \\ \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right) = 0, i = 1,\ldots,N \end{cases} \tag{3}$$

The output is a unique global optimized result that has the form:

$$\hat{y} = \hat{f}\left(x\right) = \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right)Q\left(x_i, x\right) + b \tag{4}$$

Where, $Q(x_i,x)$ is a kernel function. By employing kernels in SVR, all necessary computations can be calculated directly in the input space. Various kernels exist such as linear, hyperbolic tangent, Gaussian Radial basis function, polynomial, etc. [18]. Here, we used the commonly used RBF kernel as follows:

$$Q(x_i, x) = e^{-\gamma\|x_i - x\|^2} \tag{5}$$

The SVR's parameters (C, **γ**, and ε) play a vital role in the performance of SVR. So, choosing the proper values of these parameters leads to minimize the forecasting error as shown next in this paper. Kernelized SVRs require the computation of a distance function between each point in the dataset, which is the dominating cost of:

$$O(n_{features} * n_{observations}^2) \tag{6}$$

Solving the STLF problem based on SVR can be summarized in the following steps:

- **First step:** The historical data is loaded and divided into a training set and validation set.
- **Second step:** The parameters of SVR are determined accurately.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 8689-8695

- **Third step:** The SVR is trained using the defined parameters to get the support vectors and corresponding coefficients.
- **Fourth step:** Then the forecasted load can be obtained using (4).

EXPERIMENTS

The main purpose of this research is to study the impact of applying cloud paradigm in the field of electrical load forecasting. The experiments are designed to test two points. The first point is the accuracy of implementing load forecasting techniques using Azure ML. The second and the most important point is to test the improvement in the execution time.

Dataset

The dataset is used to verify the applying of cloud-based tools is that provided by European Network on Intelligent Technologies for Smart Adaptive Systems (EUNITE) competition [20]. The organizers of the competition had provided the half hourly electricity load demand from January 1997 to December 1998, average daily temperature from 1995 to 1998, and holiday's information from 1997 to 1999. The results of more than 25 competitors are available. The goal of the competition is to forecast the daily maximum load in January 1999. The real values of loads in January 1999 are available to compute the efficiency of the forecasting algorithm.

Implementation

The implementation of the designed experiments has two choices, local implementation, and cloud-based implementation. For the local implementation, a desktop computer is used, and the proposed algorithm is implemented using MATLAB. The desktop computer has the following specifications: Processor – Microsoft Windows 10, Intel Core i7 2.7GHz, RAM 8GB. For the cloud-based implementation, Azure ML is used.

Performance Metrics

In this research, two main performance metrics is considered. The first and the most important one is the execution time ($T_{Execution}$) . The second one is the prediction accuracy. All the experiments will be evaluated the prediction accuracy using four metrics: mean absolute percentage error (MAPE), the magnitude of maximum error (MAX), mean-absolute error (MAE) and normalized mean-square error (NMSE). These values are defined by the following equations:

$$\mathrm{MAPE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}\frac{|\mathrm{A_i} - \mathrm{P_i}|}{\mathrm{A_i}} * 100 \quad (7)$$

$$\mathrm{MAX} = \max(|\mathrm{A_i} - \mathrm{P_i}|) \quad (8)$$

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}|\mathrm{A_i} - \mathrm{P_i}| \quad (9)$$

$$\Delta^2 = \frac{1}{\mathrm{N}-1}\sum_{\mathrm{i=1}}^{\mathrm{N}}(\mathrm{A_i} - \overline{\mathrm{A}})^2 \quad (10)$$

$$\mathrm{NMSE} = \frac{1}{\Delta^2 \mathrm{N}}\sum_{\mathrm{i=1}}^{\mathrm{N}}(\mathrm{A_i} - \mathrm{P_i})^2 \quad (11)$$

where $\mathrm{A_i}$, $\mathrm{P_i}$, $\overline{\mathrm{A}}$ and N is the actual value, the predicted value, the mean of the actual values and the testing dataset size respectively.

RESULTS AND DISCUSSION

As stated before, there are a dataset was used in this research: EUNITE. For the used dataset, there were a number of experiments scenarios were examined to compare and evaluate the proposed cloud-based SVR algorithm for STLF against the other techniques and against local implementation using MATLAB and the personal PC.

Table 1: Experiments scenarios of EUNITE Dataset.

Experiment code	Dataset	Training set		Testing set	
		criteria	size	criteria	size
EUNITE1	EUNITE	• Daily Load only • All months in 1997 and 1998.	723	January 1999	31
EUNITE2	EUNITE	• Daily Load + Temp • All months in 1997 and 1998.	723	January 1999	31
EUNITE3	EUNITE	• Daily Load only • Winter months (10, 11 , 12 ,1, 2, 3) in 1997 and 1998.	357	January 1999	31
EUNITE4	EUNITE	• Daily Load + Temp. • Winter months (10, 11,12,1,2,3) in 1997 and 1998.	357	January 1999	31

Four different experiment scenarios were implemented. The target of all of these scenarios was to train the machine learning model using the training set and to use this trained model to predict the maximum daily load expected in 31 days in January 1999. The difference among the proposed experiments scenarios was the criteria of choosing the training set.

Table 1 summarizes the metadata of the proposed experiments scenarios.

Table 2: Results of EUNITE experiments.

Experiment code	$T_{Execution}$		MAPE %	MAE	MAX (MW)	NMSE	Parameters	
	Local	Cloud					γ	C
EUNITE1	677 Secs	70 Secs	2.21	16.45	42.8	0.346	0.5	1
EUNITE2	725 Secs	72 Secs	2.41	17.65	60.6	0.421	0.25	2
EUNITE3	101 Secs	67 Secs	2.08	15.55	36.35	0.295	0.25	2
EUNITE4	120 Secs	56 Secs	2.04	15.2	40.4	0.297	0.125	2

Table 2, Figure 1 and Figure 2 show the results of executing the EUNITE experiments for STLF. The results show that the execution time in the case of using the cloud was much less that using the local machine. This is observed easily in case of experiment "EUNITE2" because of the relatively big number of training data (723 * 4 (dimension of the time series) * 2(load +temp) = 5784 value) refer to equation (6). In EUNITE2, the improvement in time execution was more than 10 times in the cloud case vs. the local machine case. It is worth to state that the second run of the experiments on the cloud using Azure Ml studio was faster than the first run, for example, in EUNITE2 the execution time equals 9 seconds instead of 72 seconds. In addition, the results show that the best prediction accuracy was achieved in the case of "EUNITE4" where MAPE=2.04%. Figure 3 shows the values of MAPE and MAX for competitors in EUNITE competition compared to the result of "EUNITE4" experiment. As shown, this is the second best result when compared to the EUNITE competition results.

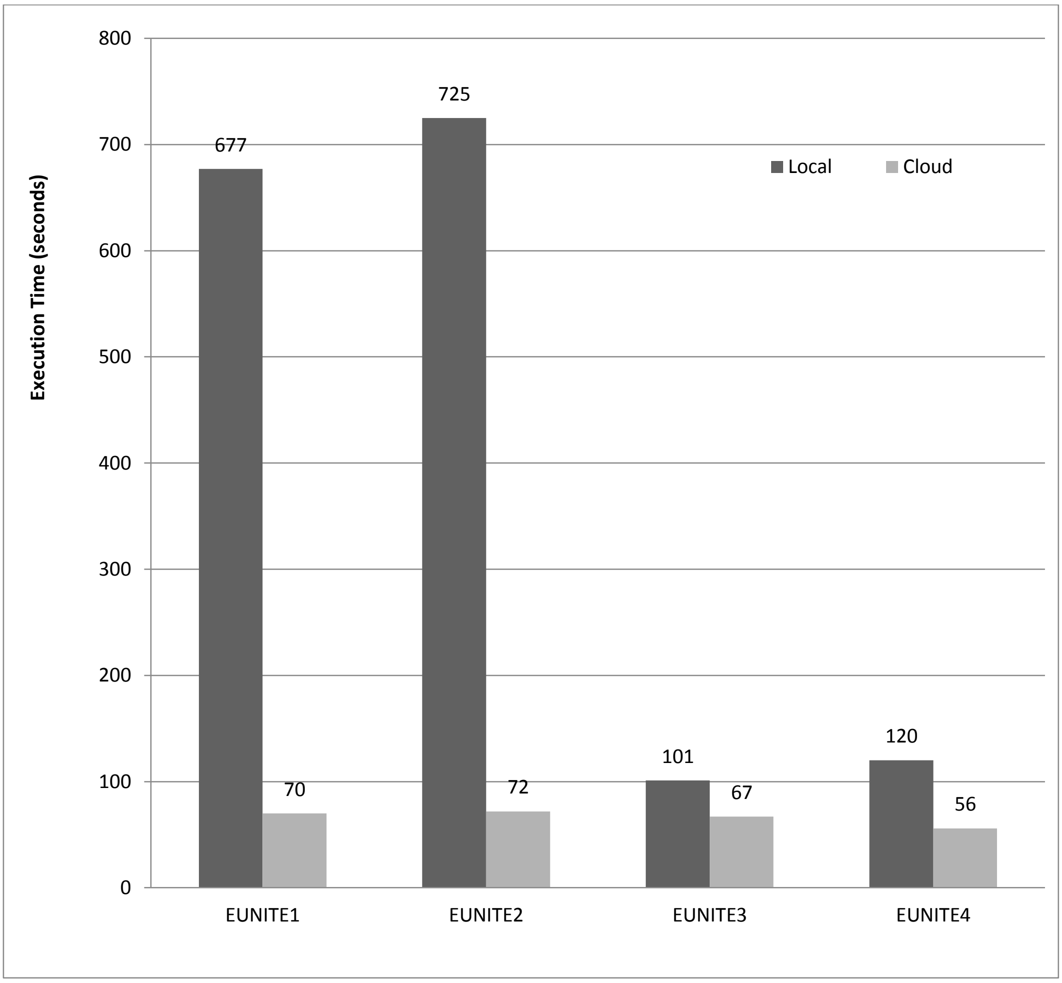

Figure 1: Execution time of EUNITE experiments in the cloud vs. local machine.

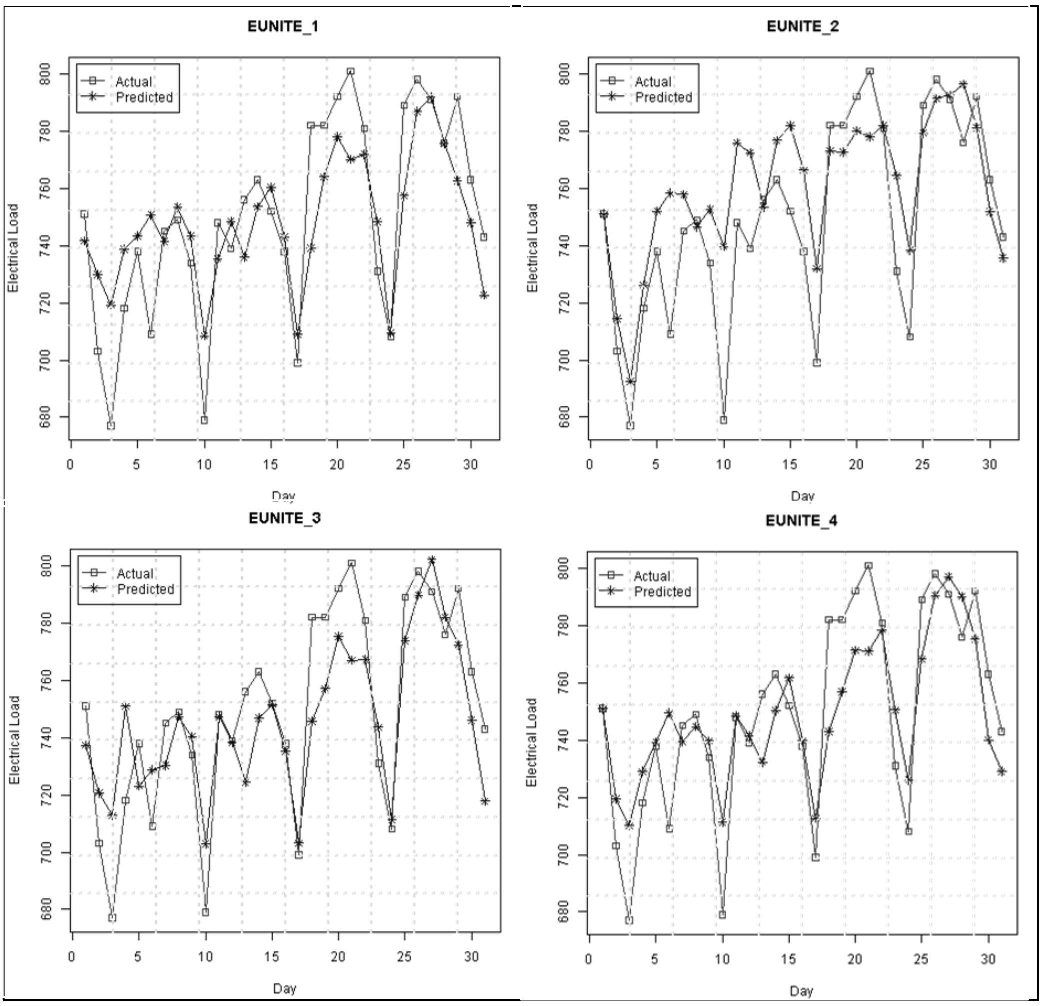

Figure 2: Predicted daily electrical load vs. the actual reads of Jan. 1999 for EUNITE experiments.

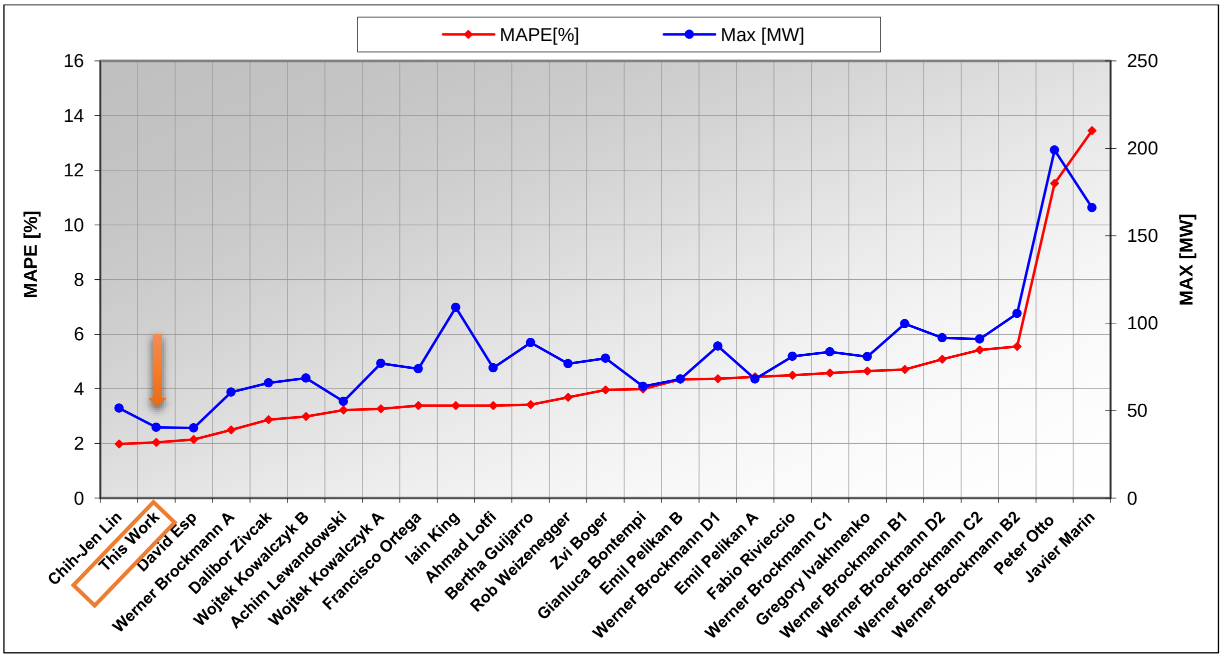

Figure 3: Results of this work compared to results of the 2001 EUNITE competition indicating MAPE and MAX metrics [20].

CONCLUSION

In this paper, A cloud base implementation for STLF using SVR is presented. The time of execution and the prediction accuracy are the performance measures. The forecasting algorithm introduced in [9] is implemented locally and on the cloud.

For the used dataset, the prediction accuracy for the two approaches was identical. But, the enhancement in the execution time is remarkable in the case of cloud implementation. Traditionally, SVR is not recommended for using with the large datasets because of its high computational cost as shown in equation (6). So, this was a good exercise for the cloud implementation.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Tail University for its support to carry out this work. It funded this project with a fund number 1-437-5282.

REFERENCES

[1] Askarpour, M., and V. Zeinadini. "Security-Constrained unit commitment reaction to load and price forecasting errors." 6th International Conference on the European Energy Market, 2009, pp. 1–7., doi:10.1109/eem.2009.5207177.

[2] Kyriakides, Elias, and Marios Polycarpou. "Short Term Electric Load Forecasting: A Tutorial." Studies in Computational Intelligence Trends in Neural Computation, vol. 35, pp. 391–418., doi:10.1007/978-3-540-36122-0_16.

[3] Huang, C.-M., et al. "A Particle Swarm Optimization to Identifying the ARMAX Model for Short-Term Load Forecasting." IEEE Transactions on Power Systems, vol. 20, no. 2, 2005, pp. 1126–1133., doi:10.1109/tpwrs.2005.846106.

[4] Senjyu, T., et al. "One-Hour-Ahead load forecasting using neural network." IEEE Transactions on Power Systems, vol. 17, no. 1, 2002, pp. 113–118., doi:10.1109/59.982201.

[5] Hippert, H.s., et al. "Neural networks for short-Term load forecasting: a review and evaluation." IEEE Transactions on Power Systems, vol. 16, no. 1, 2001, pp. 44–55., doi:10.1109/59.910780.

[6] Sheng, Siqing, and Cong Wang. "Integrating Radial Basis Function Neural Network with Fuzzy Control for Load Forecasting in Power System." 2005 IEEE/PES Transmission &Amp; Distribution Conference &Amp; Exposition: Asia and Pacific, pp. 1–5., doi:10.1109/tdc.2005.1547038.

[7] Ling, S.h., et al. "Short-Term electric load forecasting based on a neural fuzzy network." IEEE Transactions on Industrial Electronics, vol. 50, no. 6, 2003, pp. 1305–1316., doi:10.1109/tie.2003.819572.

[8] Zhang, Ming-Guang. "Short-Term load forecasting based on support vector machines regression." 2005 International Conference on Machine Learning and Cybernetics, 2005, doi:10.1109/icmlc.2005.1527695.

[9] El-Attar, E. E., et al. "Forecasting electric daily peak load based on local prediction." 2009 IEEE Power & Energy Society General Meeting, 2009, pp. 1–6., doi:10.1109/pes.2009.5275587.

[10] Elattar, E. E., et al. "Integrating KPCA and locally weighted support vector regression for short-Term load forecasting." Melecon 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference, 2010, pp. 438–447., doi:10.1109/melcon.2010.5476265.

[11] "Microsoft Azure: Cloud Computing Platform & Services." [Online]. Available: azure.microsoft.com/en-us/. Accessed 5 Apr. 2017.

[12] "Amazon Web Services (AWS) - Cloud Computing Services." [Online]. Available: aws.amazon.com/. Accessed 5 Apr. 2017.

[13] "Google App Engine" [Online]. Available: cloud.google.com/appengine/docs/. Accessed 5 Apr. 2017.

[14] Buyya, Rajkumar, et al. Mastering Cloud Computing: Foundations and Applications Programming. 1st ed., Elsevier, Inc., 2013.

[15] E. Elattar and T. Farrag, "Security Constrained Unit Commitment Based Load and Price Forecasting Using Evolutionary Optimized LSVR." International Journal of Scientific & Engineering Research, V. 5, no. 7, pp. 723-732, 2014.

[16] Sun, Changyin, and Dengcai Gong. "Support Vector Machines with PSO Algorithm for Short-Term Load Forecasting." 2006 IEEE International Conference on Networking, Sensing and Control, pp. 676–680., doi:10.1109/icnsc.2006.1673227.

[17] Vapnik, Vladimir Naumovich. Statistical learning theory. New York, Wiley & Sons, 1998.

[18] Smola, Alex J., and Bernhard Schölkopf. "A tutorial on support vector regression." Statistics and Computing, vol. 14, no. 3, 2004, pp. 199–222., doi:10.1023/b:stco.0000035301.49549.88.

[19] Kavousi-Fard, Abdollah, et al. "A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting." Expert Systems with Applications, vol. 41, no. 13, 2014, pp. 6047–6056., doi:10.1016/j.eswa.2014.03.053.

[20] Chen, B.-J., et al. "Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001." IEEE Transactions on Power Systems, vol. 19, no. 4, 2004, pp. 1821–1830., doi:10.1109/tpwrs.2004.835679.

[21] Takens, Floris. "Detecting strange attractors in turbulence." Lecture Notes in Mathematics Dynamical Systems and Turbulence, Warwick 1980, 1981, pp. 366–381., doi:10.1007/bfb0091924.