2020

A novel Monte Carlo-based neural network model for electricity load forecasting

Binbin Yong
University of Wollongong, yongbb14@lzu.edu.cn

Zijian Xu

Jun Shen
University of Wollongong, jshen@uow.edu.au

Huaming Chen
University of Wollongong, hc007@uowmail.edu.au

Jianqing Wu
University of Wollongong, jw937@uowmail.edu.au

See next page for additional authors

A novel Monte Carlo-based neural network model for electricity load forecasting

Abstract

The ongoing rapid growth of electricity over the past few decades greatly promotes the necessity of accurate electricity load forecasting. However, despite a great number of studies, electricity load forecasting is still an enormous challenge for its complexity. Recently, the developments of machine learning technologies in different research areas have demonstrated its great advantages. General Vector Machine (GVM) is a new machine learning model, which has been proven very effective in time series prediction. In this article, we firstly review the basic concepts and implementation of GVM. Then we apply it in electricity load forecasting, which is based on the electricity load dataset of Queensland, Australia. A detailed comparison with traditional back-propagation neural network (BP) is presented in this paper. To improve the load forecasting accuracy, we specially propose to use the weights-fixed method, ReLu activation function, an efficient algorithm for reducing time and the influence of parameter matrix β to train the GVM model. Analysis of our approach on the historical Queensland electricity load dataset has demonstrated that GVM could achieve better forecasting results, which shows the strong potential of GVM for general electricity load forecasting.

Keywords

monte, novel, network, model, neural, load, electricity, forecasting, carlo-based

Disciplines

Engineering | Science and Technology Studies

Publication Details

Yong, B., Xu, Z., Shen, J., Chen, H., Wu, J., Li, F. & Zhou, Q. (2020). A novel Monte Carlo-based neural network model for electricity load forecasting. International Journal of Embedded Systems, 12 (4), 522-533.

Authors

Binbin Yong, Zijian Xu, Jun Shen, Huaming Chen, Jianqing Wu, Fucun Li, and Qingguo Zhou

A Novel Monte Carlo Based Neural Network Model for Electricity Load Forecasting

Binbin Yong, yongbb14@lzu.edu.cn
School of Information Science and Engineering Lanzhou University. P.R.China
South Tianshui Road 222. Lanzhou 730000

Binbin Yong received his master's degree in Computer Science and Technology from Lanzhou University in 2012, and received PhD in the School of Information Science and Engineering, Lanzhou University in 2017. He is researching in parallel computing of GPU, machine learning, deep learning and general vector machine.

Zijian Xu, xuzj15@lzu.edu.cn
School of Information Science and Engineering Lanzhou University. P.R.China
South Tianshui Road 222. Lanzhou 730000

Zijian Xu graduated from Sun Yat-sen University in 2014. He is studying for his master degree in Lanzhou University since 2015. His major is computer science, and his interests include machine learning, cloud computing and medical informatics.

Jun Shen, jshen@uow.edu.au
School of Computing and Information Technology
University of Wollongong NSW 2522 Australia

Jun Shen was awarded PhD in 2001 at Southeast University, China. He is Associate Professor at University of Wollongong in Wollongong, NSW, Australia. He has published more than 160 papers in journals and conferences in Computer Science and Information System area. His expertise is on cloud computing and big data. He has been Editor, PC Chair, Guest Editor, PC Member for numerous journals and conferences published by IEEE, ACM, Elsevier and Springer. He won Outstanding Leadership Award from IEEE Education Society.

Huaming Chen, hc007@uowmail.edu.au
School of Computing and Information Technology
University of Wollongong NSW 2522 Australia

Huaming Chen received the B.Eng. and M.Eng. degrees from Lanzhou University, China, in 2012 and 2015 respectively. He is currently a Ph.D. candidate in University of Wollongong. His main research interests are bioinformatics, machine learning, and neural network.

Jianqing Wu, jw937@uowmail.edu.au
School of Computing and Information Technology
University of Wollongong NSW 2522 Australia

Jianqing Wu is a PhD candidate at University of Wollongong, Australia. He received the BSc in computer science from Manchester Metropolitan University, UK, in 2014 and Mres in computer science from the University of Liverpool (Xi'an Jiaotong-Liverpool University, China) in 2016 respectively. His research interests include machine learning and data analytics for Intelligent Transport System.

Fucun Li, fl626@uowmail.edu.au
School of Computing and Information Technology
University of Wollongong NSW 2522 Australia
Fucun Li graduated from University of Wollongong with master degree of Advanced Information Technology studies in 2014. He is currently a Ph.D. candidate in University of Wollongong. His main research interests are how to use information system to improve competitiveness of iron companies and how to use machine learning to control quality and cost of iron products.

Qingguo Zhou, zhouqg@lzu.edu.cn
School of Information Science and Engineering Lanzhou University. P.R.China
South Tianshui Road 222. Lanzhou 730000
Qingguo Zhou received the BS and MS degrees in Physics from Lanzhou University in 1996 and 2001, respectively, and received PhD in Theoretical Physics from Lanzhou University in 2005. Now he is a professor of Lanzhou University and working in the School of Information Science and Engineering. He is also a Fellow of IET. He was a recipient of IBM Real-Time Innovation Award in 2007, a recipient of Google FacultyAward in 2011, and a recipient of Google Faculty Research Award in 2012. His research interests include safety-critical systems, embedded systems, and real-time systems.

Abstract

The ongoing rapid growth of electricity over the past few decades greatly promotes the necessity of accurate electricity load forecasting. However, despite a great number of studies, electricity load forecasting is still an enormous challenge for its complexity. Recently, the developments of machine learning technologies in different research areas have demonstrated its great advantages. General Vector Machine (GVM) is a new machine learning model, which has been proven very effective in time series prediction. In this article, we firstly review the basic concepts and implementation of GVM. Then we apply it in electricity load forecasting, which is based on the electricity load dataset of Queensland, Australia. A detailed comparison with traditional back-propagation neural network (BP) is presented in this paper. To improve the load forecasting accuracy, we specially propose to use the weights-fixed method, ReLu activation function, an efficient algorithm for reducing time and the influence of parameter matrix β to train the GVM model. Analysis of our approach on the historical Queensland electricity load dataset has demonstrated that GVM could achieve better forecasting results, which shows the strong potential of GVM for general electricity load forecasting.

Keywords: Electricity load forecasting; BP neural network; General Vector Machine

1. Introduction

As a basic energy in our daily life, electricity is becoming more and more important, with the improvement of people's living standard. Unfortunately, with the increase of electricity consumption, more and more electric energy is wasted due to the poor electricity load forecasting. Thus, an accurate electricity forecasting is very momentous, especially for satisfying people's requirements for higher usage of electricity. Hidden in the growing trend is a fluctuation caused by the alteration in demand from day to day and month to month (Kenny and Durbin, 1982; Dagum, 1978). In fact, in the past few decades, electrical energy requirement has dramatically enlarged. For example, in the 1980s, due to the development of the economy proceeding quicker than that of the electric-power industry, China had a scarcity of electric power (Murata et al., 2008). In addition, electricity load requirements depend on not only the economic factors but also the environment conditions.

Recently, the power grid collapse frequently happens in the electric power systems. Once the power grid collapsed, people's daily life, basic activity, and public utility system will be greatly influenced. For example, a serious power grid collapse event happened in India; the power grid in the northern region collapsed on July 30, 2012, the basic life of 350 million people in nine Indian northern regions was affected. The worse thing is that they suffered a more serious large range power outage the following day; the basic services, public transport system, and daily life of more than 600 million people were greatly affected, which inspires people to pay more attention to assisting avoiding unexpected power grid collapse effectively. To achieve this, analysis and forecast electricity load in one region can provide basic information to grasp the demand trends and correlative changes (Taylor et al., 2006).

On the other hand, electricity cannot be stored extensively, which means electricity demand must be satisfied instantaneously and producers need to accurately anticipate future demands to avoid overproduction (Vilar et al., 2012). Electricity supply plan requires efficient management of existing power systems which takes the responsibility for hourly scheduling of the electricity generators. It means that the power systems need to guarantee the balance of power production and demand. Therefore, we need to minimize the error between forecasting load and real power generation to reduce resource consumption. Finally, there must be a production surplus, with which local failures do not dramatically affect the whole system. Hence, to optimize the electricity systems, we need to develop a scheduling algorithm for the hourly generation and transmission of electricity (Wang et al., 2012; Soares and Souza, 2006). To sum up, accurate short-term and mid-term electricity demand forecasting become significant for electricity consumers and producers.

From the perspective of technical approaches, in the prediction sub-discipline, the most frequently used method for training artificial neural networks (ANN) is back-propagation (BP) algorithm (Hagan et al., 2002), which is used to adjust the weight matrices of the ANN by gradient descent algorithm. However, because of being sensitive to the initial values and easily falling into local minimum, BP based ANN often suffers prediction accuracy problem for electricity forecasting. Meanwhile, the random algorithms used for training ANN are studied. Wang et al. (2016) explored the performance in both diluted and full-connected ANN based on Monte Carlo (MC) algorithm. Zhao (2016) used the idea of MC algorithm to adjust the weights of ANN, which is designed as the so-called General Vector Machine (GVM) model. Results show that, in many cases, GVM performs better than BP, especially for forecasting issues with small dataset (Wang et al., 2016; Zhou et al., 2016). Chen et al. (2015) had successfully applied GVM to detect colon cancer and got good results. Hence, in this paper, we propose to apply GVM in load forecasting for small electricity demand dataset.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 presents the GVM model and its architecture. Section 4 discusses the performance optimization strategy of GVM model. In Section 5, we give the details of our experiments and results. Finally, Section 6 concludes this paper.

2. Related Work

The literatures present various electricity load forecasting approaches over the last two decades (Wang et al., 2016). In general, the traditional electricity load forecasting is based on time series forecasting, which mainly applies the historical electricity load data to forecast the load in the future. The classic forecasting methods include trends extrapolation, regression models and so on (Bianco et al., 2009; Goia et al., 2010; Wu et al., 2013; Magnano et al., 2007; Wang et al., 2016; Agrawal et al.,2017). Time series forecasting techniques include autoregressive moving average (ARMA) (Wu et al., 2013; Wang and Schulz, 2006), grey-based prediction models (Bahrami et al., 2014), experts and experience forecasting etc. Recently, some machine learning technologies are used to forecast the electricity load, which include self-organizing maps (Che et al., 2012), support vector machine(SVM) (Kavousi-Fard et al., 2014; Xiong et al., 2014; Ju and Hong, 2013; Shayeghi and Ghasemi, 2013), particle swarm optimization(PSO) (Shayeghi and Ghasemi, 2013), ANN (Ju and Hong, 2013), fuzzy logic (Shayeghi and Ghasemi, 2013; Gu et al., 2016) and genetic algorithm (GA) (Kucukali and Baris, 2010).

Bates et al. (1969) proposed a combined method, which tries to sum up the advantages of various methods. Dickinson (1975) proved that the mean absolute error (MAE) of the combined method is lower than a single method, which means that the combined method could perform better than the single method. Afterwards, various combination methods are used to forecast electricity load. For example, Hernández et al. (2014) presented a new method, which combines self-organizing maps, means algorithm and multilayer perceptron. Che and Wang (2014) proposed a kernel-based support vector regression (SVR) combination model to forecast the electricity load of Australia and California. In order to get more accurate results of electricity load forecasting, Geng et al. (2015) combined seasonal SVR model and chaotic cloud simulated annealing algorithm. Che (2013) applied the adaptive PSO algorithm to select the parameters in SVR model. Zhang et al. (2012) used the chaotic genetic algorithm (GA), simulated annealing algorithm (SA), and the SVR to forecast the cyclic electricity demand in Northeast China, in which the SVR was applied to deal with the trend of cyclic electricity load. Liu et al. (2014) introduced a combination model that mainly concentrates on the parameters optimization, and the model was tested with electricity load data in micro-grid.

Some other studies are as follows: Hamzacebi and Es (2014) used the direct optimized grey modeling (1, 1) and iterative optimized grey modeling (1, 1) to forecast electricity load of Turkey. And the results showed that the direct forecasting approach performs better than the iterative forecasting approach. Lu et.al (2015) proposed a new on-line network training method namely distributed hyper-spherical ARTMAP (dHS-ARTMAP) to forecast the electricity load. The forecasting results of dHS-ARTMAP show the effectiveness compared with other methods, and it is a promising alternative to be put into practice. Yan and Chowdhury (2014) used multiple SVMs to forecast the mid-term electricity market clearing price. In his experiments, the multiple SVMs model outperforms the forecasting model with a single SVM.

Although a number of combined methods have shown good performance in electricity load forecasting, there are still some limitations. Hence, we will introduce the newly proposed GVM method for electricity forecasting in the next sections.

3. General Vector Machine Model

Recently, based on statistical method and MC algorithm, Zhao (2016) proposed GVM model to deal with prediction for small dataset problems. In this section, we will introduce the basic structure and implementation of GVM.

As shown in Fig. 1, GVM is based on 3-layer ANN, which includes input layer, hidden layer and output layer. The weight matrix connected between the input layer and the hidden layer is denoted as ***W1***, and the weight matrix between the hidden layer and the output layer is denoted as ***W2***. The bias matrix of hidden layer is denoted as ***B***. In our implementation, the biases of hidden nodes are organized as part of ***W1***. β is parameter of each hidden node, which is used to control the stability performance of GVM model.

The process of computing the output nodes is the so-called forward propagation, which mainly includes two steps (Faruk, 2010).

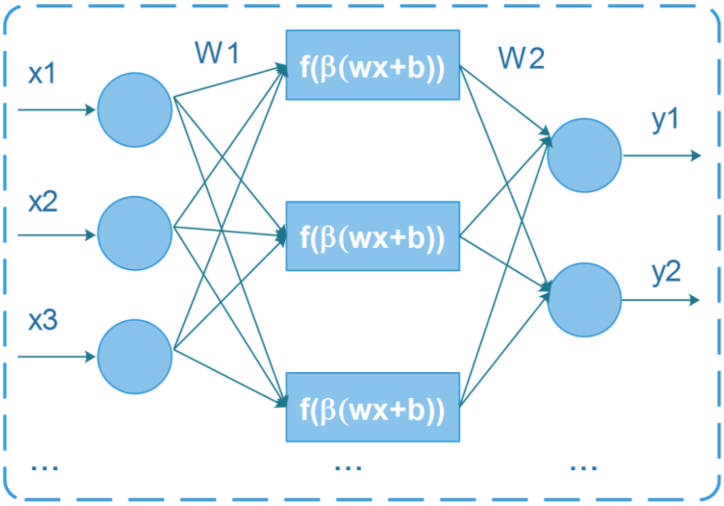


Fig. 1 The architecture of GVM based on 3-layer ANN

Firstly, the values of the hidden nodes are calculated by Eq. (1) and Eq. (2). M is the number of input nodes and N is the number of hidden nodes.

$$a_i = \sum_{j=1}^{M} W1_{ij} \cdot x_j + b_i, \quad i = 1, \dots, N \tag{1}$$

$$h_i = f(\beta_i a_i), \quad i = 1,2, \dots, N \tag{2}$$

In Eq. (1), b_i is the bias of the i_{th} hidden node and a_i is the weighted sum of the i_{th} hidden node. h_i is the output of the i_{th} hidden node. β_i is an important parameter whose range could influence the performance of the network seriously, f is the activation function of the hidden nodes which is defined by the following Eq. (3):

$$f(x) = \frac{1}{1+\exp(-x)} \tag{3}$$

In general, there are several different non-linear functions, which are suitable for the selection of activation function. In fact, we will test the influences of different activation functions in our experiment part.

At last, the outputs of GVM are simply calculated by Eq. (4). L is the number of output nodes.

$$y_l = \sum_{i=1}^{N} W2_{li} \cdot a_i , \quad l = 1,2, \ldots , L \tag{4}$$

In Eq. (4), y_l (l=1, 2, …, L) is the final outputs of the GVM. In reality, we can add activation functions to the output nodes. However, the line function could achieve a good result in our prediction model. Hence, we simply use the weighted sum of the hidden nodes as the outputs of the output layer.

Before training the GVM model, we randomly initialize the weights of matrix $\boldsymbol{W1}$ as float numbers between -1 and +1. The weights of matrix $\boldsymbol{W2}$ are randomly set as -1 or +1 and we will not change matrix $\boldsymbol{W2}$ after initialization. The biases of hidden nodes are randomly initialized between -1 and +1. After initialization, we can train our GVM model with MC algorithm.

Generally, to train a GVM model, we need to fix the weight matrix $\boldsymbol{W2}$ and adjust the weights and biases in matrix $\boldsymbol{W1}$. The specific procedure of training GVM is stated as follows, and the flowchart is presented in Fig. 2.

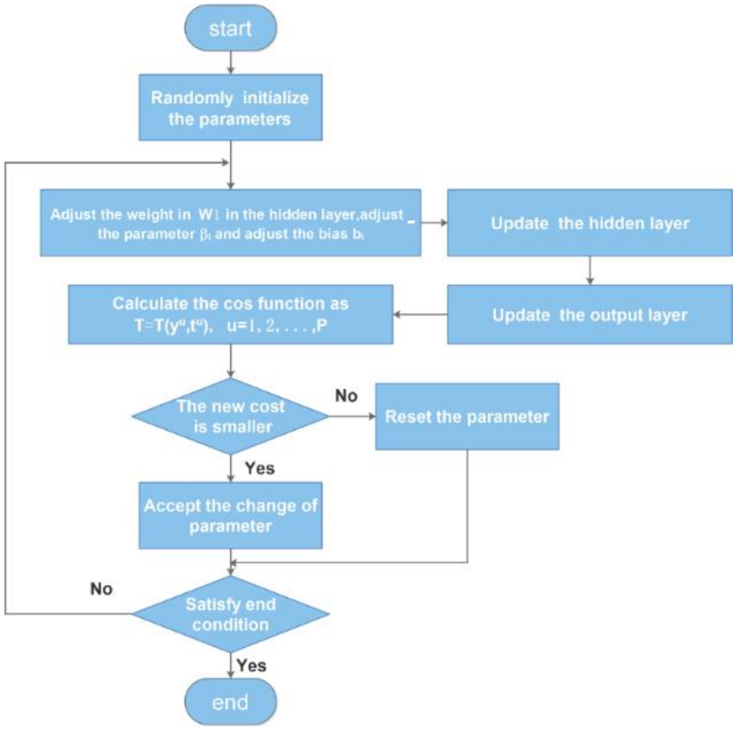


Fig. 2 The flowchart of training GVM

(1) In order to know when the training process should end, an overall cost is defined as the function: $T = T(y^u, t^u)$, in which u is the dimension of output nodes, while t^u represents the actual output of GVM under the input x^u.

(2) The weights and biases in matrix $\boldsymbol{W1}$ are all set to be random values between -1 and +1. The weights of matrix $\boldsymbol{W2}$ are randomly set to be -1 or +1. The parameter matrix β is initialized to values in a range, which is usually between 0 and 1.

(3) After initialization, we repeat the following procedure to adjust the weights in $\boldsymbol{W1}$ and parameters in matrix $\boldsymbol{\beta}$ to reduce the overall cost: Firstly, we randomly select a weight or bias in matrix $\boldsymbol{W1}$, or parameter in matrix $\boldsymbol{\beta}$ and change it to a new value within its intervals. Then we calculate the new overall cost of the training samples. If the cost gets smaller, we accept the change and record the

new minimum cost. Otherwise, we reset the changed weight or parameter to its original value. In reality, we do not need to recalculate all the nodes in the hidden layer. The changes happen in the nodes whose connected weights are changed. Hence, we only update the output values of the hidden nodes connected to the changed weights. By this way, the overall cost is calculated more efficiently.

(4) We repeat step (3) until the overall cost is less than a fixed value T_0, or the training time t is larger than a fixed time interval t_0. T_0 and t_0 are artificially set before training.

When training GVM by MC algorithm, we change the weight parameters in a small deviation ε. For a weight with discrete states (Rosen-Zvi and Kanter, 2001; Zhao, 2004), the parameter ε is used to change the weights jumping from the present interval to a neighboring interval randomly. After repeating the above step (1), (2), (3), (4) several iterations, the overall cost of the trained GVM model will obviously converge to a small value, which indicates that the training process finishes.

4. Optimization of GVM

In this section, we will employ the performance optimization methods in training GVM for electricity load forecasting, which are divided into four subsections including weights-fixed method, mixed activation functions, the efficient algorithm for overall cost and the influence of parameter matrix $\boldsymbol{\beta}$.

4.1 Weights-fixed method

As mentioned in Sect. 3, we fix the weight matrix connected between the hidden layer and output layer, which is the so-called $\boldsymbol{W2}$ matrix because of the fact that we can reduce the computation of training GVM by fixing $\boldsymbol{W2}$ matrix. To verify this, we also conducted an experiment to test the training times with the weights-fixed method and weights-unfixed method.

In our experiments, the minimum value of overall cost is set as 0.01. Then, we compare the training times between fixed and unfixed methods for the different dimension of hidden nodes. The results are shown in Table 1, in which we denote the GVM model with N hidden nodes as N-GVM

Table 1. The average and variance of training times of weights-fixed method and weights-unfixed method

Hidden dimension	Average training time(s)			Variance		
	100-GVM	500-GVM	1000-GVM	100-GVM	500-GVM	1000-GVM
Fixed	41.345	**40.880**	43.004	88.323	**67.352**	208.945
Unfixed	57.756	**54.294**	74.837	150.793	**102.454**	259.366

From Table 1, we can conclude that weights-unfixed method takes more time in training compared to the weights-fixed method. While the 500-GVM achieves the best performance compared to 100-GVM and 1000-GVM. Moreover, the training time of weights-unfixed method is nearly 1.5 times of the weights-fixed method. Hence, we adopt the weights-fixed method in our paper.

4.2 Mixed activation functions

Generally, GVM is absolutely free to select its activation functions. Hence, herein we also researched the influences of activation functions. In fact, we could apply a different type of activation functions in GVM to achieve a better performance. In our test, we use different activation functions to construct the GVM model for electricity demand forecasting. The popularly used activation functions in constructing ANN include *sigmoid* function, *tanh* function, *gauss* function and Rectified Linear Units (*ReLu*) etc. These functions are given in the following Eq. (5) – Eq. (8):

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{5}$$

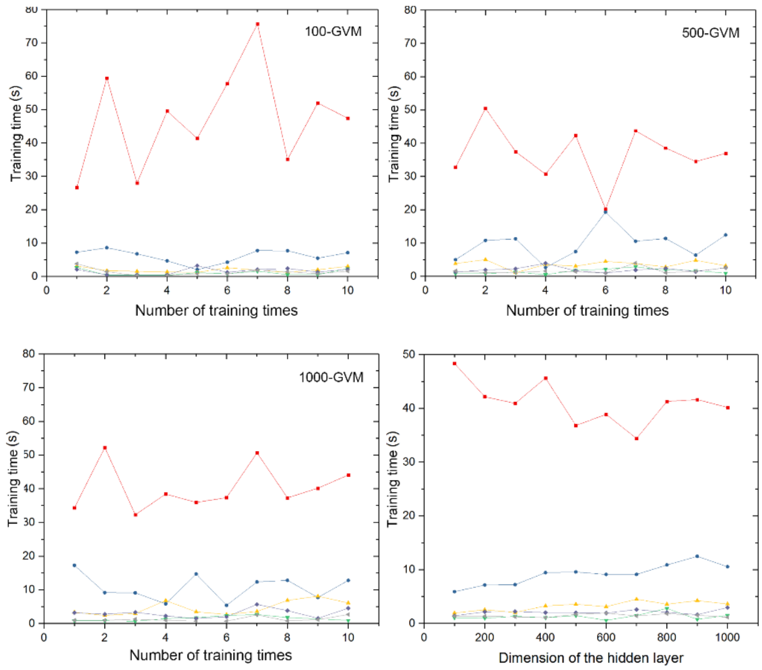
$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{6}$$

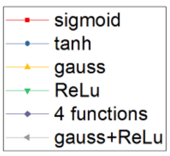
$$gauss(x) = e^{-x^2} \tag{7}$$

$$ReLu(x) = \begin{cases} x, & x > 0 \\ 0, & x \le 0 \end{cases} \tag{8}$$

These activation functions are randomly selected by the hidden nodes of our GVM model. In reality, we found that using *gauss* function and *ReLu* function as activation functions of hidden nodes could achieve better forecasting performance.

Hence, we also test the combination of the two activation functions. In our training, the minimum value of overall cost is set as 0.01. Then, we train the GVM models and compare the training time of the GVM models with different activation functions. The results are shown in Fig. 3, from which we can get the following conclusions. If only a single type activation function is used, the GVM with *ReLu* activation function shows the best forecasting performance. Moreover, along with the increase of the dimension of hidden layer, training time of GVM with *ReLu* activation function is not much affected. It is noted that *ReLu* activation function is widely used in constructing deep ANN. And Krizhevsky et al. (2012) found that *ReLu* activation function could achieve a better efficiency than *sigmoid* activation function. Furthermore, *ReLu* activation function has advantages over traditional *sigmoid* and *tanh* activation functions in training 3-layer ANN. We can also see from Fig. 3 that time-consuming of training GVM with *sigmoid* function is larger than the time with other activation functions. Also, the training time of GVM with *tanh* function increases with the increase of the dimension of hidden layer. Meanwhile, we also found that *gauss* activation function showed a better performance than *sigmoid* and *tanh*, but no better than *ReLu* function. The combination of multi-activation functions also shows a good result in training time, but no better than using a single *ReLu* activation function. Overall, the *ReLu* activation function is the best choice of GVM for electricity load in this paper. Therefore, *ReLu* is selected as the activation function of GVM to accelerate the training process of electricity load forecasting model.





Active function	Average			Variance		
	100 dimension	500 dimension	1000 dimension	100 dimension	500 dimension	100 dimension
sigmoid	47.307	36.771	40.282	207.587	60.356	40.568
tanh	6.158	9.708	10.712	3.663	19.51	13.587
gauss	1.968	3.566	4.628	0.457	1.193	3.965
ReLu	1.071	1.456	1.443	0.668	0.48	0.431
4 functions	1.59	2.031	3.054	0.803	0.625	1.629
gauss and ReLu	1.304	1.692	1.256	0.95	0.782	0.493

Fig. 3 Comparisons s of training performance with different activation functions

4.3 An efficient algorithm to change only one weight in each iteration

In GVM, only one weight is changed in each iteration. Hence, the output values of hidden nodes can be saved to accelerate the training process. With these saved outputs, we only need to recalculate the outputs of the node connected to the changed weight in each iteration, which makes the algorithm more efficient. Conventionally, GVM algorithm recalculates all the outputs of all nodes to get a new cost of GVM. We tested the performances of the conventional algorithm (namely normal algorithm) and our new efficient algorithm, as illustrated in Fig. 4.

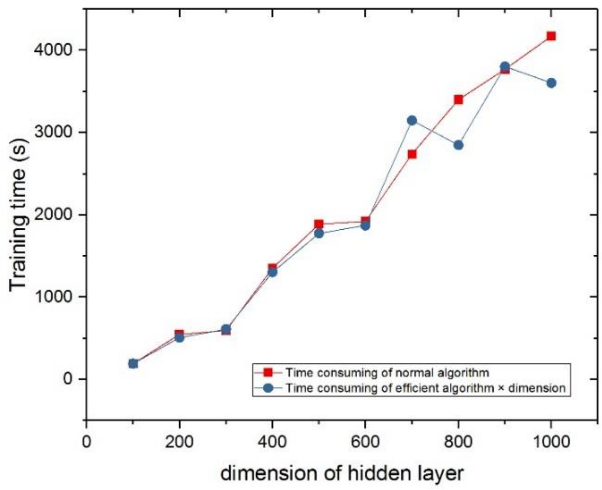


Fig. 4 Comparisons s of training performance between normal algorithm and efficient algorithm

From Fig. 4, we can see that the training time of normal algorithm nearly coincides with the training time of efficient algorithm multiplied by N (the dimension of the hidden layer of GVM), which means that the efficient algorithm could improve the performance by N times compared to the normal algorithm. Therefore, the efficient algorithm performs better than the normal algorithm.

4.4 The influence of parameter matrix β

Finally, we discuss the influence of the parameter matrix ***β***, which affects the stability of training results. Because we randomly initialize the weight matrices of the network, and the weights are also randomly updated, the experimental results will change in a range. In fact, this is the design risk of GVM. In other words, with a large interval of β, we get a bigger design risk. Therefore, the parameter β reflects the stability and the repeatability of GVM. In our experiment, we fix the other hyper parameters but the range of weights in β. That is to say, we only change the value of β to test the influences. For each β, we repeat the experiment 40 times to get an average result. By this way, we can test the stability of GVM. The experimental results are shown in Fig. 5.

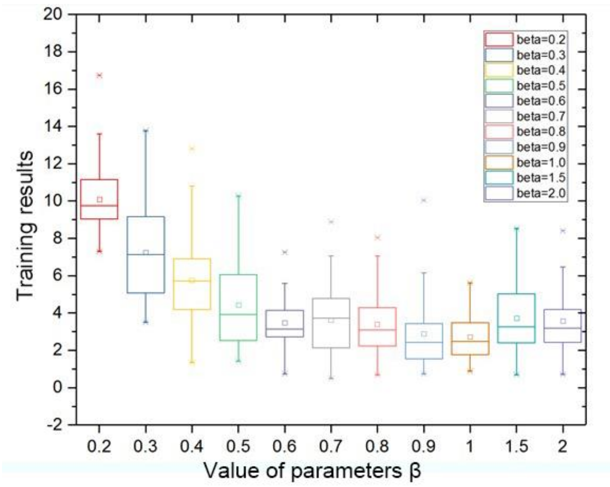


Fig. 5 Stability of training timings of different parameters β

From Fig. 5, we can see that: when the value of β is smaller than 0.4, the training time is long and the result is unstable. However, when the value of beta ranges from 0.6 to 1.0, the training time is almost the same. In these results, the training time is optimal when β is fixed as 0.6. Normally, the value of β is not fixed for different applications, where β in general is kept in the range between 0 and 1. Hence, in order to get the best result, we need to test different β values. In this paper, we empirically set beta as 0.6, by which we can achieve stable and best experimental results.

Above all, we specially proposed to use the weights-fixed method, ReLu activation function and the efficient algorithm to improve the performance of GVM model for electricity load forecasting. And we also optimized the parameter β to improve the stability of GVM model.

5. EXPERIMENT RESULTS

5.1 Electricity data

There is a simple fact that the electricity load data has internal rules with it. For example, the people's life, industrial and commercial activities have some kind of similarity on Mondays in a month statistically, which leads to the situation that electricity load data from all Mondays in a month are similar. By using the similarity, we divide the electricity load data and use the data to train our GVM model.

In this paper, the electricity load data from May 2, 2011, to July 3, 2011, in Queensland is selected as the dataset. As mentioned above, we divide the electricity load data into seven groups, which include Monday group, Tuesday group, Wednesday group, Thursday group, Friday group, Saturday group and Sunday group, which are shown in Fig. 6.

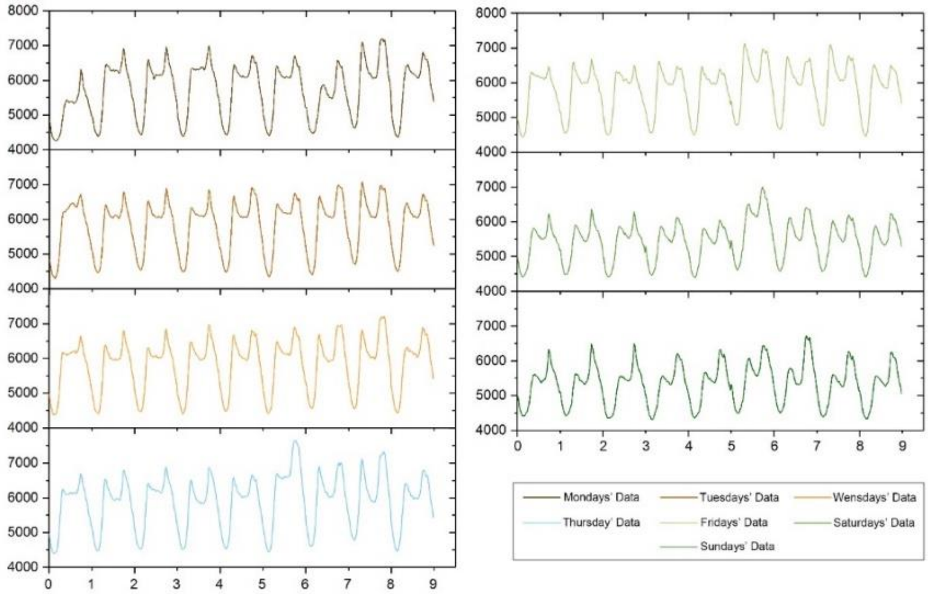


Fig. 6 The original electricity demand data of 7 groups.

The electricity load data in Queensland was collected every half an hour, so there are 48 electricity data in one day. We divide the electricity from May 2, 2011, to July 3, 2011, into seven groups (data of 63 days), and each group has electricity data of 9 days. For example, Monday group includes the data of May 2, May 9, May 16, May 23, May 30, June 6, June 13, June 20 and June 27. Among these 9 days' electricity data, the first eight days' data is used to as the input vector of GVM model, and the electricity data of last day is seen as the data to be predicted. Specifically, when training the GVM model, the first seven days' electricity data is used as the input vector, and the data of the eighth day is used as the output vector. When testing the model, the data from the second day to the eighth day is used as the input vector, and the data of the ninth day is used as the output to be predicted. Similarly, the data used in other six groups is preprocessed in the same way.

5.2 Forecasting evaluation methods

In this paper, different forecasting performance metrics are evaluated to determine the accuracy of the prediction models, including smallest mean error (AE), root mean square error (RMSE), MAE and mean absolute percentage error (MAPE), which are given in the following Eq. (9) – Eq. (12).

$$\mathrm{AE} = \frac{1}{n}\sum_{i=1}^{n}(P_i - A_i) \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(P_i - A_i)^2}{n}} \tag{10}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|(P_i - A_i)| \quad (11)$$

$$\mathrm{MAPE} = \frac{\sum_{i=1}^{n}\left|\frac{(P_i - A_i)}{A_i}\right|}{n} \times 100\% \quad (12)$$

Where P_i and A_i are the predicted and actual values, respectively, and *n* is the number of forecasting samples. Lower values of these measures indicate better forecasting results. According to Lin and Hsu (2002), if MAPE is less than 10%, the model is successful for electricity load forecasting.

5.3 Experiment results and analysis

In this section, we will firstly give the results comparisons between traditional BP and GVM model, then the comparisons between different GVM models are given. In the first part, the experiment environments of two models are set the same. For convenience, we call the BP based neural network as BP and denote the GVM model with *N* hidden nodes as *N*-GVM. In the second part, we will change the number of hidden nodes *N* to compare different GVMs.

5.3.1 BP vs 100-GVM

In the experiments of comparing BP and GVM, we also fix the dimension of input vector, transfer function and the dimension of output vector. The forecasting results of BP and 100-GVM in each day are illustrated in Fig. 7.

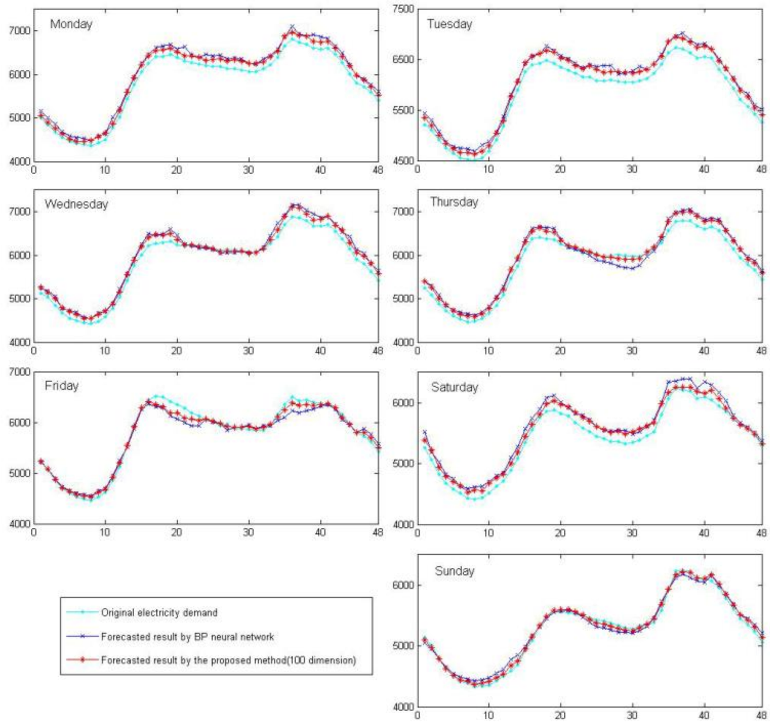


Fig. 7 Forecasting results of BP and 100-GVM

From Fig. 7, we can observe that the curves of GVM model are more consistent with the original data curves compared with the curves of BP. In order to compare these two methods more clearly, the metrics of AE, RMSE, MAE and MAPE are calculated.

From the bar figures of the four metrics of BP and GVM shown in Fig. 8, we can get the following intuitive conclusion: the AE, MAE, RMSE and MAPE of BP are both larger than 100-GVM. That is to say, 100-GVM performs better than BP for electricity load forecasting.

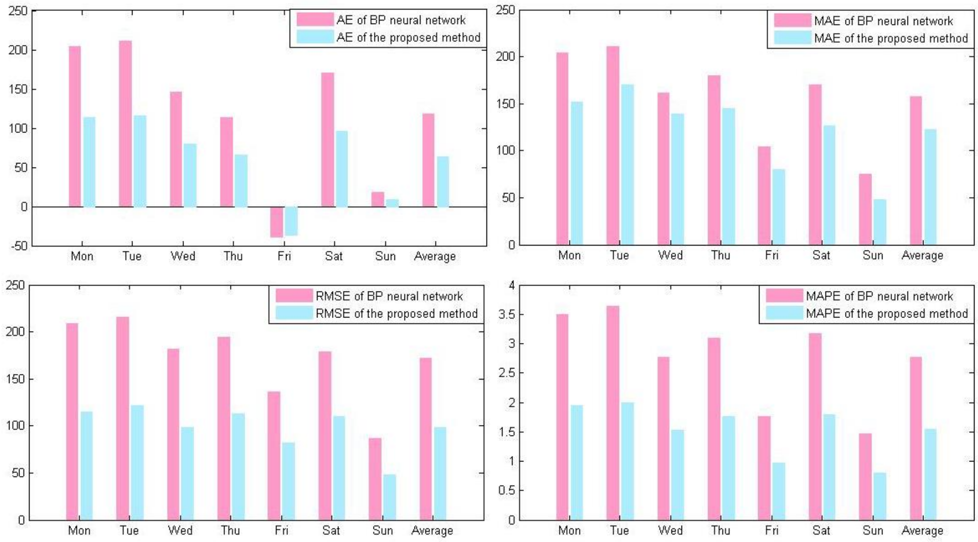


Fig. 8 Bar figures of BP and proposed GVM

5.3.2 100-GVM, 500-GVM vs 1000-GVM

In this section, we test the GVM with different dimensions of hidden nodes. Herein, we will see that 100-GVM has the worst forecasting results compared with 500-GVM and 1000-GVM. However, compared to BP, it reduced AE by 46.26%, MAE by 22.33%, RMSE by 42.79% and MAPE by 44.52%.

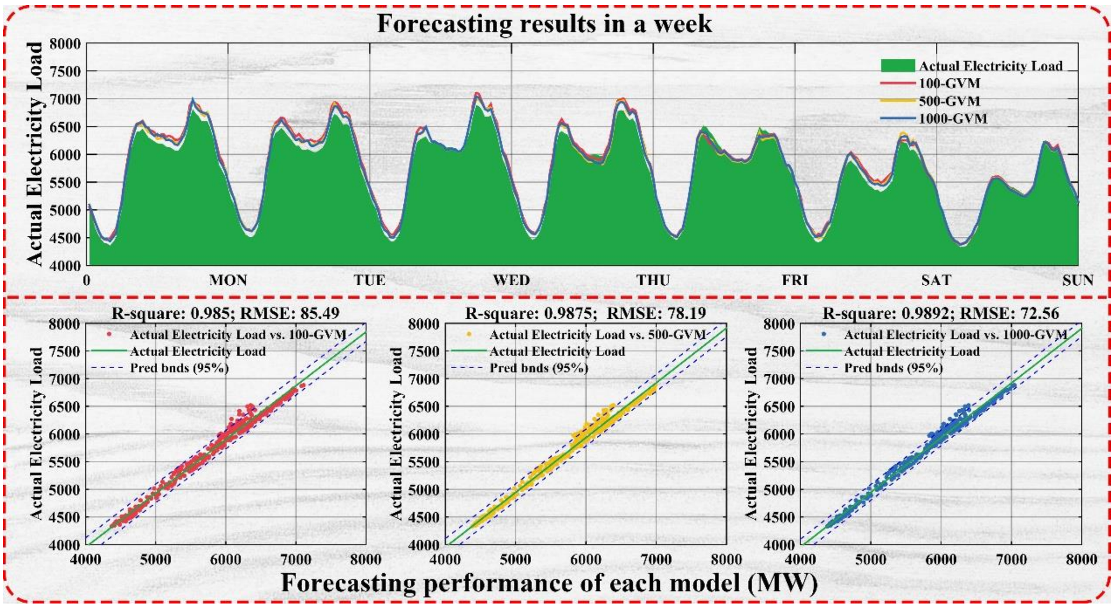


Fig. 9. Forecasting results of GVM in each day

In order to verify the effectiveness of GVM model used in this paper, we conduct the following three experiments. We fix the dimension of the input vector, transfer function, parameter matrix β and the dimension of the output vector. Meanwhile, we change only the dimension of hidden nodes. We try different dimensions of 100, 500 and 1000, respectively. The forecasting results of 100-GVM, 500-GVM and 1000-GVM in each day are illustrated in Fig. 9.

From Fig. 9, we can conclude that there are little differences between these GVM models. The forecasting curves of each day are almost coincident with the original data. In fact, the forecasting curves of 500-GVM and 1000-GVM are particularly good. In order to see the results clearly, the metrics of these three GVM models are also listed in the following Table 2 and Table 3.

Table 2. The AE and MAE of GVM with different dimensions of hidden nodes

Date	AE of			MAE of		
	100-GVM	500-GVM	1000-GVM	100-GVM	500-GVM	1000-GVM
Monday	145.854	113.065	**111.701**	145.854	113.065	**111.701**
Tuesday	165.247	**115.876**	119.924	165.247	**115.876**	119.924
Wednesday	114.989	79.853	**78.972**	122.614	**88.237**	88.254
Thursday	116.521	65.265	**56.296**	130.246	**101.618**	105.281
Friday	**-13.777**	-37.085	-21.500	60.281	**57.225**	63.486
Saturday	114.666	**96.063**	106.085	114.708	**96.063**	106.085
Sunday	20.228	**9.019**	11.639	**40.881**	41.158	41.398
Whole week	94.818	**63.151**	66.160	111.404	**87.606**	90.876

The AE and MAE in each day have little differences in Table 2. For example, on Monday, the AE and MAE of the three models have progressively declined, which means that the 1000-GVM performs better than 500-GVM and 100-GVM. Hence, the accuracy order of these models for Monday is: 1000-GVM, 500-GVM and 100-GVM. Take the results on Tuesday, Saturday, and Sunday, it can be found that 500-GVM performs better than 1000-GVM and 100-GVM. Hence, the accuracy order of forecasting these 3 days is: 500-GVM, 1000-GVM and 100-GVM. However, in general, if considering the average values of the whole week, there are little differences between AE and MAE metrics of the three GVM models.

Table 3. The RMSE and MAPE of GVM with different dimensions of hidden nodes

Date	RMSE of			MAPE of (%)		
	100-GVM	500-GVM	1000-GVM	100-GVM	500-GVM	1000-GVM
Monday	151.771	115.762	**114.601**	2.486	1.939	**1.924**
Tuesday	169.689	**117.856**	121.954	2.802	**1.995**	2.066
Wednesday	138.745	99.158	**98.108**	2.138	**1.520**	1.530
Thursday	144.476	**109.616**	112.570	2.243	**1.752**	1.815
Friday	79.818	**74.400**	82.252	1.031	**0.971**	1.059
Saturday	126.226	**100.657**	110.129	2.163	**1.788**	1.955
Sunday	47.587	**47.173**	47.848	**0.781**	0.801	0.805
Whole week	122.616	**94.946**	98.209	1.949	**1.538**	1.593

The above analysis can also be applied to the results in Table 3. Though the RMSE and MAPE in each day have some differences, the accuracy order of Monday is: 1000-GVM, 500-GVM and 100-GVM, and the accuracy order of the Tuesday, Saturday and Sunday is: 500-GVM, 1000-GVM and 100-GVM. Similarly, if average values of the whole week are considered, there is not much difference for RMSE and MAPE metrics between these three models. For example, the MAPEs of three models are 1.949, 1.538 and 1.593 respectively. According to Lin and Hsu (2002), an accepted MAPE is less than 10% for a successful prediction model. Therefore, the three GVM models are successfully applied to the electricity load forecasting.

Because of the similarity of the average MAPE of the three models, we can conclude that the dimension of hidden nodes has little impact on the electricity load forecasting, while BP algorithm always tries to reduce the number of hidden nodes to reduce the risk of over-fitting. Instead, GVM is able to control the over-fitting problem by setting the structure of the network. Meanwhile, GVM uses more hidden nodes to forecast the dataset better.

6. CONCLUSION

In this paper, we aim to apply a novel GVM model to short-term electricity load forecasting with the actual data collected from Queensland, Australia. By combining MC algorithm, GVM is more suitable for time series prediction for small dataset. We also discuss the influences of the weights-fixed method, mixed activation functions, the efficient algorithm for training our model and the influences of the β matrix. The comparisons with BP-ANN are also given in the article. Results show that using GVM model in electricity load forecasting is effective and efficient. Although this paper significantly extended the work that we reported in (Yong et al., 2017), especially by providing a discussion on the performance of the GVM, the combination of GVM model with other model is still not discussed, which needs to be undertaken in the future. For example, applications of fuzzy theories to incorporate linguistic values in ANN are widely used in time series prediction, such as the long-term travel time prediction model with a fuzzy neural network (Li et al., 2017). The fluctuations of electricity supplies and demands could be affected by many indeterministic factors such as extreme weather and human errors or even terrorist attacks. In this case, more work on new model for electricity load forecasting needs to be researched.

REFERENCES

Kenny, P. B. and Durbin, J. (1982). Local trend estimation and seasonal adjustment of economic and social time series. Journal of the Royal Statistical Society, 145(145), 1-41.

Dagum, E. B. (1978). Modelling, forecasting and seasonally adjusting economic time series with the x-11 ARIMA method. Journal of the Royal Statistical Society, 27(3/4), 203-216.

Murata, A., Kondou, Y., Mu, H. and Zhou, W. (2008). Electricity demand in the chinese urban household-sector. Applied Energy, 85(12), 1113-1125.

Taylor, J. W., Menezes, L. M. D. and Mcsharry, P. E. (2006). A comparison of univariate methods for forecasting electricity demand up to a day ahead. International Journal of Forecasting, 22(1), 1-16.

Vilar, J. M., Cao, R. and Aneiros, G. (2012). Forecasting next-day electricity demand and price using nonparametric functional methods. International Journal of Electrical Power & Energy Systems, 39(1), 48-55.

Wang, C. H., Grozev, G. and Seo, S. (2012). Decomposition and statistical analysis for regional electricity demand forecasting. *Energy,41*(41), 313–325.

Soares, L. J., Souza, L. R. (2006) Forecasting electricity demand using generalized long memory. Int J Forecast 2006; 22:17–28.

Hagan, M. T., Demuth, H. B. and Beale, M. H. (2002). Neural network design. China Machine Press.

Wang, L., Shen, J., Zhou, Q., Shang, Z., Chen, H., & Zhao, H. (2016). An evaluation of the dynamics of diluted neural network. International Journal of Computational Intelligence Systems, 9(1-6), 1191-1199.

Zhao, H., (2016) “General vector machine,” arXiv preprint, arXiv:1602.03950, 2016.

Zhou, Q., Chen, H., Zhao, H., Zhang, G., Yong, J. & Shen, J. (2016). A local field correlated and Monte Carlo based shallow neural network model for non-linear time series prediction. Scalable Information Systems, 2016, 3(8):e5

Chen, H., Zhao, H., Shen, J., Zhou, R. & Zhou, Q., (2015) Supervised Machine Learning Model for High Dimensional Gene Data in Colon Cancer Detection. *IEEE BigData Congress*, pp.134-141

Bianco, V., Manca, O. and Nardini, S. (2009). Electricity consumption forecasting in italy using linear regression models. Energy, 34(9), 1413-1421.

Goia, A., May, C. and Fusai, G. (2010). Functional clustering and linear regression for peak load forecasting. International Journal of Forecasting, 26(4), 700-711.

Wu, J., Wang, J., Lu, H., Dong, Y. and Lu, X. (2013). Short term load forecasting technique based on the seasonal exponential adjustment method and the regression model. Energy Conversion & Management, 70(70), 1–9.

Magnano, L. and Boland, J. W. (2007). Generation of synthetic sequences of electricity demand: application in south australia. Energy, 32(11), 2230-2243.

Wang, T. D., Wu, X., & Fyfe, C. (2016). Factors important for good visualisation of time series. Inderscience Publishers.

Agrawal, K. P., Garg, S., Sharma, S., Patel, P., & Bhatnagar, A. (2017). Fusion of statistical and machine learning approaches for time series prediction using earth observation data. International Journal of Computational Science & Engineering, 14(3), 255-266.

Wang, H. and Schulz, N. N. (2006). Using AMR data for load estimation for distribution system analysis. Electric Power Systems Research, 76(5), 336-342.

Bahrami, S., Hooshmand, R. A. and Parastegari, M. (2014). Short term electric load forecasting by wavelet transform and grey model improved by PSO (particle swarm optimization) algorithm. Energy, 72(7), 434-442.

Che, J., Wang, J. and Wang, G. (2012). An adaptive fuzzy combination model based on self-organizing map and support vector regression for electric load forecasting. Fuel & Energy Abstracts, 37(1), 657-664.

Kavousi-Fard, A., Samet, H. and Marzbani, F. (2014). A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting. Expert Systems with Applications, 41(13), 6047-6056.

Xiong, T., Bao, Y. and Hu, Z. (2014). Interval forecasting of electricity demand: a novel bivariate EMD-based support vector regression modeling framework. International Journal of Electrical Power & Energy Systems, 63: 353-362.

Ju, F. Y. and Hong, W. C. (2013). Application of seasonal SVR with chaotic gravitational search algorithm in electricity forecasting. Applied Mathematical Modelling, 37(23), 9643-9651.

Shayeghi, H. and Ghasemi, A. (2013). Day-ahead electricity prices forecasting by a modified CGSA technique and hybrid wt in LSSVM based scheme. Energy Conversion & Management, 74(11), 482-491.

Gu, L., Liu, X., & Guo, H. (2017). Fuzzy time series forecasting based on information granule and neural network. International Journal of Computational Science & Engineering, 15(1/2), 146.

Kucukali, S. and Baris, K. (2010). Turkey's short-term gross annual electricity demand forecast by fuzzy logic approach. Energy Policy, 38(5), 2438-2445.

Bates, J. M. and Granger, C. W. J. (1969). The combination of forecasts. Journal of the Operational Research Society, 20(4), 451-468.

Dickinson, J. P. (1975). Some comments on the combination of forecasts. Journal of the Operational Research Society, 26(1), 205-210.

Hernández, L., Baladrón, C., Aguiar, J. M., Carro, B., Sánchez-Esguevillas, A. and Lloret, J. (2014). Artificial neural networks for short-term load forecasting in microgrids environment. Energy, 75(C), 252-264.

Che, J. X. and Wang, J. Z. (2014). Short-term load forecasting using a kernel-based support vector regression combination model. Applied Energy, 132(11), 602-609.

Geng, J., Huang, M. L., Li, M. W. and Hong, W. C. (2015). Hybridization of seasonal chaotic cloud simulated annealing algorithm in a SVR-based load forecasting model. Neurocomputing, 151, 1362-1373.

Che, J. X. (2013). Support vector regression based on optimal training subset and adaptive particle swarm optimization algorithm. Applied Soft Computing, 13(8), 3473-3481.

Zhang, W. Y., Hong, W. C., Dong, Y., Tsai, G., Sung, J. T. and Fan, G. F. (2012). Application of svr with chaotic gasa algorithm in cyclic electric load forecasting. Energy, 45(1), 850–858.

Liu, N., Tang, Q., Zhang, J., Fan, W. and Liu, J. (2014). A hybrid forecasting model with parameter optimization for short-term load forecasting of micro-grids. Applied Energy, 129, 336-345.

Hamzacebi, C. and Es, H. A. (2014). Forecasting the annual electricity consumption of turkey using an optimized grey model. Energy, 70(3), 165-171.

Lu, X., Wang, J., Cai, Y. and Zhao, J. (2015). Distributed hs-artmap and its forecasting model for electricity load. Applied Soft Computing, 32(C), 13-22.

Yan, X. and Chowdhury, N. A. (2014). Mid-term electricity market clearing price forecasting: a multiple svm approach. International Journal of Electrical Power & Energy Systems, 58(58), 206–214.

Faruk, D. Ö. (2010). A hybrid neural network and arima model for water quality time series prediction. Engineering Applications of Artificial Intelligence, 23(4), 586-594.

Rosen-Zvi, M. and Kanter, I. (2001). Training a perceptron in a discrete weight space. Physical Review E Statistical Nonlinear & Soft Matter Physics, 64(4 Pt 2), 046109.

Zhao, H. (2004). Designing asymmetric neural networks with associative memory. Physical Review E Statistical Nonlinear & Soft Matter Physics,70(6 Pt 2), 066137.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. International Conference on Neural Information Processing Systems (Vol.25, pp.1097-1105). Curran Associates Inc.

Lin, C. and Hsu, P. (2002). Forecast of non-alcoholic beverage sales in taiwan using the grey theory. Asia Pacific Journal of Marketing and Logistics, 14(4), 3-12.

Tseng, F. M., Yu, H. C. and Tzeng, G. H. (2001). Applied hybrid grey model to forecast seasonal time series. Technological Forecasting & Social Change, 67(2–3), 291-302.

Yong, B., Xu, Z., Shen, J., Chen, H., Tian, Y., & Zhou, Q. (2017). Neural network model with Monte Carlo algorithm for electricity demand forecasting in Queensland. Australasian Computer Science Week Multiconference (pp.47). ACM.

Li, R., Rose, G., Chen, H. and Shen, J. (2017). Effective long-term travel time prediction with fuzzy rules for tollway. Neural Computing and Applications, online first, 1-13. http://dx.doi.org/10.1007/s00521-017-2899.