THE FORECAST OF ELECTRICAL POWER DISTRIBUTION UNIT USING SUPPORT VECTOR REGRESSION OPTIMIZED WITH GENETIC ALGORITHM

Ronnachai Chuentawat[1*], Kittisak Kerdprasop[1], and Nittaya Kerdprasop[1]

Received: June 17, 2016; Revised date: October 05, 2016; Accepted date: October 06, 2016

Abstract

This research applied time series forecasting with a proposed algorithm: the genetic algorithm optimizing support vector egression (GASVR). The forecasting accuracy performance of the GASVR has been compared with the techniques of an artificial neural network and an autoregressive integrated moving average to forecast the power consumption of Bangkok's metropolitan area. Time series data in terms of the electrical power distribution nit for household electricity usage were obtained from the Metropolitan Electricity Authority of Thailand. The forecasting performance of each model is measured by the root mean square error (RMSE) and the mean absolute percentage error (MAPE) metrics. The experimental results of the RMSE and MAPE comparisons between the 3 models reveal that the GASVR model has the lowest RMSE and MAPE. Based on such results, we can conclude that the proposed GASVR algorithm, which is the support vector regression with parameter optimization by the genetic algorithm, is the most powerful model to forecast time series data in the specific domain of household power consumption.

Keywords: Support vector regression, genetic algorithm, artificial neuralnNetwork, ARIMA model, time series

Introduction

This research studies univariate time series data having a single observed variable that changes its value by time order. This data set is the electrical power distribution unit (EPDU) of the Metropolitan Electricity Authority of Thailand (MEA) covering household consumers in Bangkok and the provinces of Samutprakan and Nonthaburi. The time series data were collected from the MEA's monthly reports during the period from January 2010 to May 2015. The data are presented in an electrical power distribution unit metric in which 1 unit refers to 1,000 kilowatts per hour. These data in a monthly period scale are expected to contain a time series pattern regarding the demand for electricity. An accurate capturing of such a pattern is obviously beneficial to the MEA in that an accurate forecast provides precise information which is advantageous in resource planning, management of funding, and reducing

[1] *School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand. Tel. 0-4422-4422; Fax. 0-4422-4602; E-mail: c_ronnachai@hotmail.com*

* *Corresponding author*

the operational costs. Bunn and Farmer (1985) found and reported that 1% of electrical load forecasting errors raised the operational costs by 10 million pounds. Therefore, forecasting precision is a challenging problem, especially the demand for electricity supply.

The data used in this study are of limited size containing only 65 observed values. However, there has been much research in the past confirming that electricity consumption data have underlying patterns that can be captured with a low error rate regardless of their size. This research work to generate the forecasting model from a small data set can be examined as follows. Wang *et al*. (2012) used 31 data sets of the annual total electricity consumption of Beijing city between 1978 and 2008 to generate a forecasting model using support vector regression (SVR) optimized with differential evolution. Then, comparing the forecasting power with a back propagation neural etwork and regression technique, they found that their method yielded the best model showing the lowest mean absolute percentage error (MAPE) at 4.8%. Wang *et.al*. (2012) used 35 data sets of the monthly electricity consumption of northwest China from March 2007 to Jan 2010 for training a forecasting model using a seasonal autoregressive integrated moving average (ARIMA), particle swarm optimization, and hybrid method. The hybrid method was the best model with the MAPE value at 2.38%. Kaytez *et al*. (2015) used 40 data sets of the annual electricity consumption of Turkey during the period from 1970 to 2009 to generate a forecasting model using regression analysis, neural networks, and least squares support vector machines. The best model was the least squares support vector machines which achieved the lowest MAPE value at 1.004%. Therefore, we are confident that the data set containing 65 observed values used in this research is sufficient in its size for generating a reliable forecasting model, if we use the appropriate technique to generate the model.

The appropriate strategy for forecasting time series data is to apply several techniques to generate the forecasting models. After that the best model can be chosen by the measurement of the forecasting errors. Previous research studies that explored many possible techniques to find the best model for forecasting time series data in the specific domain of electricity consumption can be reviewed as follows.

Wang *et al*. (2012) tried to increase the forecasting accuracy of the model to predict the electrical load in China. They created a seasonal ARIMA model for the forecasting task. They also modified the residuals to increase the accuracy using the seasonal ARIMA, particle swarm optimization, and hybrid method. They found that the residual modification could increase the forecasting accuracy and the hybrid method was the best method for electricity forecasting.

Lee and Tong (2011) studied a time series that was the yearly data of electricity consumption in China. They presented a hybrid model of the ARIMA and genetic programming methods for forecasting. The model's accuracy was measured by 3 metrics: root mean square error (RMSE), mean absolute error, and mean absolute percentage error (MAPE). They reported that the hybrid model could generate a forecast more accurately than a single ARIMA or single genetic programming model.

Wang and Meng (2012) studied the forecasting of the energy consumption in China using the ARIMA, artificial neural network (ANN), and hybrid ARIMA-ANN models. They compared the accuracy using the RMSE, mean absolute error, and MAPE. The result of their study showed that a hybrid model provided better accuracy in forecasting than either the ARIMA or ANN models.

Ogcu *et al*. (2012) did an experiment on forecasting electricity consumption in Turkey. The data set was a time series that had a monthly time period. They used the ANN and SVR techniques to model the data set and compared the accuracy by means of the MAPE. The result was that the model using SVR had more accuracy than the model generated from the ANN.

Jirong *et al*. (2011) did an experiment to forecast housing prices in China. The data set was a time series that had a yearly time period between 1993 and 2002. They applied a genetic algorithm (GA) to optimize the SVR parameters and called their method G-SVR. After building the model, they compared the accuracy between their own G-SVR and a grey model. The result was that the G-SVR showed more accuracy than the grey model.

From the literature review, it can be concluded that existing methods to model a time series can be divided into 2 groups. The first group is traditional time series modeling such as regression analysis, exponential smoothing, or the Box and Jenkins method (Box and Jenkins, (1990). The second group is the machine learning method such as an ANN or SVR. In traditional time series modeling, the most popular model is the ARIMA model from the Box and Jenkins method. The popularity is due to its high precision (Lee and Tong, 2011; Wang *et al*., 2012 ; Wang and Meng, 2012). In the machine learning method, the SVR model has recently been used extensively because it shows high precision in predicting various data sets (Wang and Meng, 2012; Ogcu *et al*., 2012; Jirong *et al*., 2011; Fan *et al*., 2016). SVR was also found to be the suitable technique to model small sized time series data (*Fan et al*., 2016).

On applying the SVR technique to model a time series' data, the model's accuracy depends on the appropriate setting of the parameters for the SVR. Therefore, the objective of this research is modeling the EPDU data to forecast future values as accurately as possible. We propose a novel technique using a genetic algorithm to optimize the parameters of the SVR and the proposed technique is called the genetic algorithm optimizing support vector regression (GASVR). After presenting the GASVR technique, we experimentally compare its forecasting accuracy against 2 well-known techniques: the ANN and ARIMA. The comparison is based on the measurement of the RMSE and MAPE.

Materials and Methods

This research adopts the EPDU data of the MEA with a particular sector of household consumers. The EPDU data set is a monthly time series collected from the monthly reports of the MEA from January 2010 to May 2015. In this research, we apply the GASVR to model the electricity consumption pattern of the population in the Bangkok metropolitan and surrounding areas. The forecasting efficiency of the GASVR model is compared with the ANN and ARIMA models. The GASVR technique is

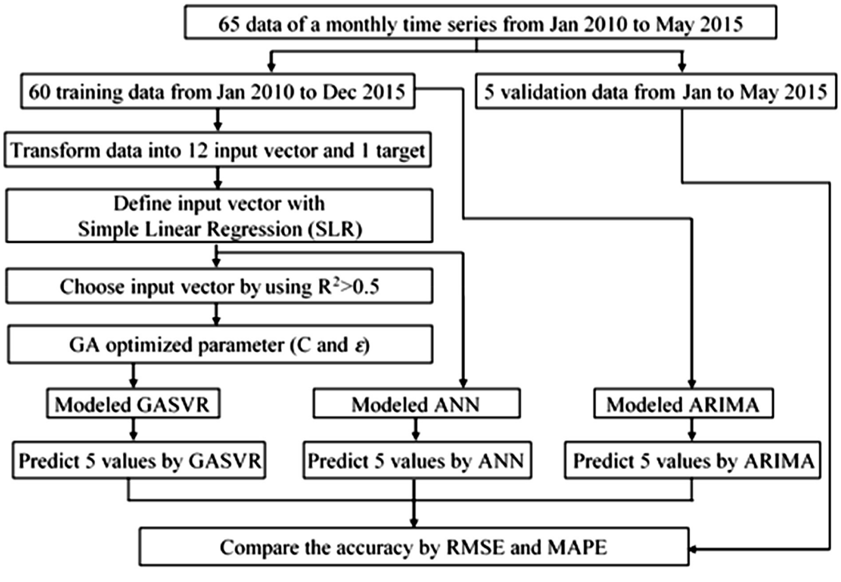

65 data of a monthly time series from Jan 2010 to May 2015
60 training data from Jan 2010 to Dec 2015
5 validation data from Jan to May 2015
Transform data into 12 input vector and 1 target
Define input vector with Simple Linear Regression (SLR)
Choose input vector by using R²>0.5
GA optimized parameter (C and ε)
Modeled GASVR
Modeled ANN
Modeled ARIMA
Predict 5 values by GASVR
Predict 5 values by ANN
Predict 5 values by ARIMA
Compare the accuracy by RMSE and MAPE

Figure 1. The conceptual framework of this research

explained in the following sub-section with its conceptual framework illustrated in Figure 1. The concepts of the ANN and ARIMA are also presented at the end of this section.

GASVR Model

Support vector regression (SVR) was introduced as a machine learning technique by Vapnik (1995). SVR is an extension of support vector machine (SVM), which is a classification method, to analyze the regression between the input vector and the output. SVR is thus used to forecast numeric values instead of the categorical classification, as traditionally has been done by the SVM. Therefore, SVR focuses on finding a linear relationship mapping the input vector X in n- dimensions ($X \in R^n$) to the output y ($y \in R$). Due to the fact that SVR has been modified from the SVM, the regression equation of SVR is therefore similar to a hyperplane equation of the SVM. The regression equation of SVR can be shown in Equation 1 (Bagheripour et al., 2016).

$$f(x) = w^T x + b \tag{1}$$

where, *w* is a slope and b is an offset of the regression line. We can define w and b by minimizing Equation 2 (Bagheripour *et al.*, 2016).

$$R = \frac{1}{2}\|w\|^2 + \frac{C}{l}\sum_{i=1}^{l}|y_i - f(x_i)|_\varepsilon \tag{2}$$

When we use SVR to predict the output from the input vector, we will define the epsilon tube by the loss function. In our research, we use an ε-insensitive loss function that can be shown in Equation 3 (Vapnik, 1995).

$$|y_i - f(x_i)|_\varepsilon = \begin{cases} 0 & if\ |y_i - f(x_i)|_\varepsilon \le \varepsilon \\ |y_i - f(x_i)|_\varepsilon - \varepsilon & Otherwise \end{cases} \tag{3}$$

To solve Equation 2 that has Equation 3 as a constraint, we can reformulate the problem to the dual problem by using Lagrange multipliers, as shown in Equations 4 and 5 (Bagheripour *et al.*, 2016).

$$Maximize\ L_p(\alpha_i, \alpha_i^*) = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i^T x_j \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\,y_i \tag{4}$$

$$Subject\ to \begin{cases} \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\ 0 \le \alpha_i \le C, \quad i = 1, \dots, l \\ 0 \le \alpha_i^* \le C, \quad i = 1, \dots, l \end{cases} \tag{5}$$

where $\alpha_i, \alpha_i^* > 0$ are the positive Lagrange multipliers, C is the cost of error and it is a positive parameter, ε is the width of the epsilon tube, and l is the number of the support vector. The support vector is an input vector that has $\alpha_i, \alpha_i^* >$ and when we calculate α_i, α_i^* from the training set, we can formulate the equation of the SVR to predict the output from the input vector as shown in Equation 6 (Bagheripour *et al.*, 2016).

$$f(x) = w_0^T x + b = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\,x_i^T x + b \tag{6}$$

The weight vector of the regression hyperplane (w_0) is given by Equation 7 (Bagheripour *et al.*, 2016).

$$w_0 = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\,x_i \tag{7}$$

Equation 6 is the format of linear regression. For the non-linear case, we can map the input vector in high dimensional feature space by using a kernel function. The mapping can be the multiplication of the vectors x_i and x_j, and the popular kernel functions can be listed as follows (Sajan *et al.*, 2015).

(1) Linear kernel: $k(x_i, x) = x_i^T x$
(2) Polynomial kernel: $k(x_i, x) = (1 + x_i.x_j)^d$
(3) Gaussian (RBF) kernel:
$$k(x_i, x) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$$

Therefore, Equation 6 can be reformulated in the format of non-linear regression by using the kernel function as shown in Equation 8 (Bagheripour *et al.*, 2016).

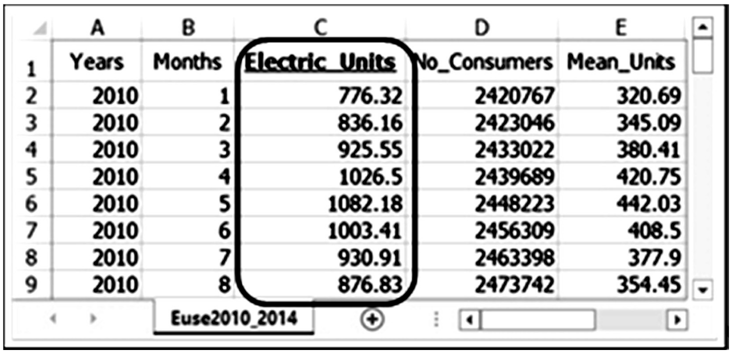
$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\,k(x_i, x) + b \tag{8}$$

This research created the model of SVR with R language and used the “svm()” function available in the “e1071” package. This package was introduced by Meyer *et al*. (2015) and is publicly available in the CRAN repository. To generate the SVR model, we need to define the kernel function. From a series of experiments, we found that the linear kernel gave the best result, so we used the linear kernel in our research. For the parameters C and ε, their optimal values were to be searched for with the genetic algorithm (GA), and the final model would be called the GASVR model. From the conceptual framework, as shown in Figure 1, we can describe the steps to generate the GASVR model as follows.

1. Collect the data from the monthly reports of the MEA. The data were obtained from January 2010 to May 2015. The data are the household electrical usage of residents in the MEA area. We then order the data to build up a time series as shown in Figure 2.

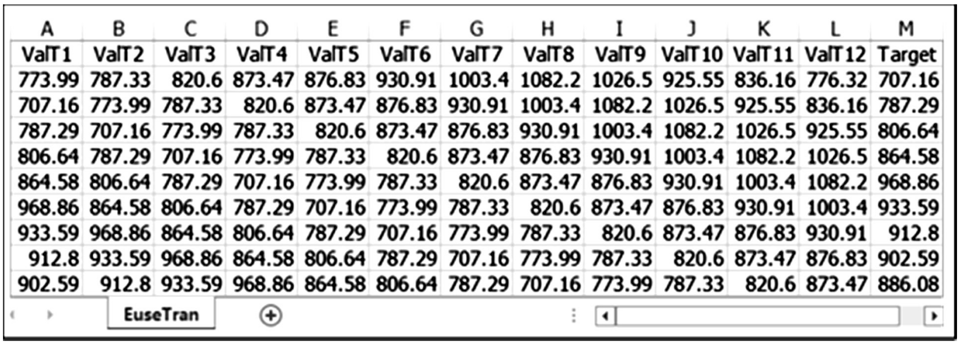
2. Divide the data into 2 data sets. The first data set is the training set to create the forecasting model. Our training set is the data from January 2010 to December 2014. The second data set is the validating set to measure the accuracy of the forecasting model. The validating set is the data from January to May 2015.

3. For the model creation of the GASVR and ANN algorithms, the data set that is used to build the model must be in the form of a pair between the input vectors and corresponding targets $(D = \{(x_i, y_i)\})_{i=1}^{n})$. Therefore, the data in Figure 2 would be transformed to the data as shown in Figure 3 before the data can be used to build the model. Each column of the data set in Figure 3 is the lag time ranging from the



	A	B	C	D	E
1	Years	Months	Electric_Units	No_Consumers	Mean_Units
2	2010	1	776.32	2420767	320.69
3	2010	2	836.16	2423046	345.09
4	2010	3	925.55	2433022	380.41
5	2010	4	1026.5	2439689	420.75
6	2010	5	1082.18	2448223	442.03
7	2010	6	1003.41	2456309	408.5
8	2010	7	930.91	2463398	377.9
9	2010	8	876.83	2473742	354.45

Figure 2. The time series data from the monthly reports



A	B	C	D	E	F	G	H	I	J	K	L	M
ValT1	ValT2	ValT3	ValT4	ValT5	ValT6	ValT7	ValT8	ValT9	ValT10	ValT11	ValT12	Target
773.99	787.33	820.6	873.47	876.83	930.91	1003.4	1082.2	1026.5	925.55	836.16	776.32	707.16
707.16	773.99	787.33	820.6	873.47	876.83	930.91	1003.4	1082.2	1026.5	925.55	836.16	787.29
787.29	707.16	773.99	787.33	820.6	873.47	876.83	930.91	1003.4	1082.2	1026.5	925.55	806.64
806.64	787.29	707.16	773.99	787.33	820.6	873.47	876.83	930.91	1003.4	1082.2	1026.5	864.58
864.58	806.64	787.29	707.16	773.99	787.33	820.6	873.47	876.83	930.91	1003.4	1082.2	968.86
968.86	864.58	806.64	787.29	707.16	773.99	787.33	820.6	873.47	876.83	930.91	1003.4	933.59
933.59	968.86	864.58	806.64	787.29	707.16	773.99	787.33	820.6	873.47	876.83	930.91	912.8
912.8	933.59	968.86	864.58	806.64	787.29	707.16	773.99	787.33	820.6	873.47	876.83	902.59
902.59	912.8	933.59	968.86	864.58	806.64	787.29	707.16	773.99	787.33	820.6	873.47	886.08

Figure 3. The data set for modeling with the GASVR

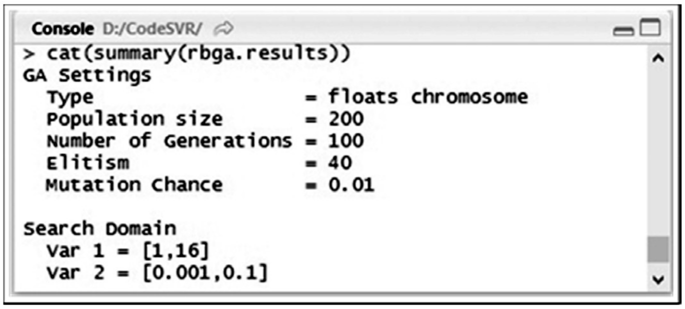
1 to 12 time periods (y_{t-1}, y_{t-2},…, y_{t-12}). The last column is a target, which is the observed value at time t. The first value of the target is the observed value that has the time index equal to 13 (t=13). After transforming the data, we will get 53 rows from 65 observed values that have the time period from January 2010 to May 2015.

4. Use the simple linear regression to define the input vectors of the GASVR by using the data set as shown in the Figure 3. We created 12 models of simple linear regression and each model used the observed values at the lag times 1 to 12 (y_{t-1},y_{t-2},…,y_{t-12}) as independent variables. All 12 models have the same observed value at time t (yt) as a dependent variable.

5. Select the input vectors of the GASVR by considering the value of R^2 (coefficient of determination). It must be greater than or equal to 0.5 ($R^2 \geq 0.5$).

6. Define the optimal C and ε parameters by applying the GA. In the R language, it has the “rgba()” function in the “genalg” package that was introduced by Willighagen and Ballings (2015) and which is available in the CRAN repository of R language for finding the optimal solution by the GA. The process of the GA can be described as follows.

6.1 *Define the chromosome representation*. This step defines the pattern of the genes in a chromosome. This research uses



```
> cat(summary(rbga.results))
GA Settings
  Type                  = floats chromosome
  Population size       = 200
  Number of Generations = 100
  Elitism               = 40
  Mutation Chance       = 0.01

Search Domain
  Var 1 = [1,16]
  Var 2 = [0.001,0.1]
```

Figure 4. The settings of the genetic algorithm

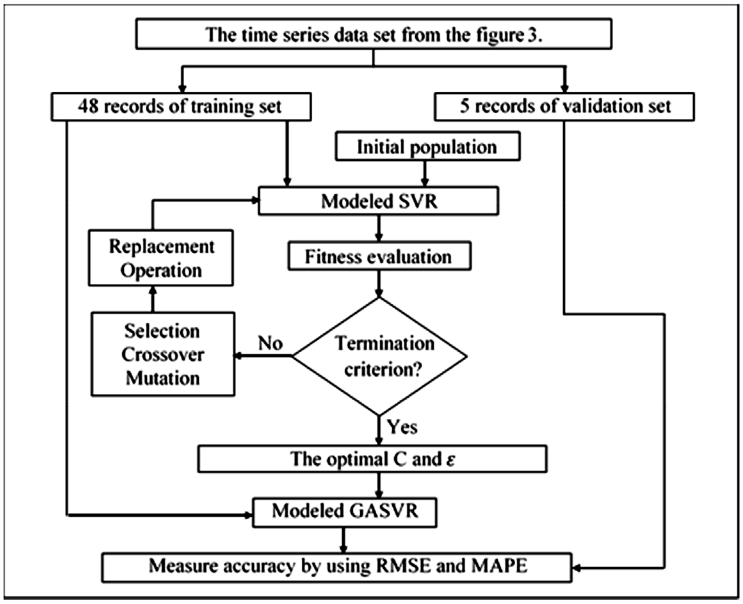

The time series data set from the figure 3.
48 records of training set
5 records of validation set
Initial population
Modeled SVR
Replacement Operation
Fitness evaluation
Selection Crossover Mutation
No
Termination criterion?
Yes
The optimal C and ε
Modeled GASVR
Measure accuracy by using RMSE and MAPE

Figure 5. The operational process of the GASVR

a GA to find the optimal C and ε, so that each chromosome consists of 2 genes that represent C and ε. The type of chromosome encoding is a float chromosome and the setting of the GA can be shown as in Figure 4. We randomly set the initial population with 200 chromosomes.

6.2 *Evaluate the fitness value*. The evaluation of the fitness in each chromosome is used to choose the offspring that will be the next generation. In our research, we define an evaluation function to be the RMSE, defined as shown in Equation 9 (Bergmeir and Benítez, 2012).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \qquad (9)$$

6.3 *Perform the selection operation*. This step is the selection of the offspring that will be the next generation. This research uses the roulette wheel method for the selection operation.

6.4 *Perform the crossover operation*. This step is to build the offspring. This research uses 2 parent chromosomes for the single-point crossover operation and sets the rate of crossover to be 0.8.

6.5 *Perform the mutation operation*. This step is to mutate the offspring by using only a single parent chromosome. The mutation is used to avoid the problem of local optimum and the rate of mutation is set to be 0.01.

6.6 *Perform the replacement operation*. This step is the replacement over the existing population by using the new set of the population that has the fitness value better than the old population set. We replace the parents with the offspring at the rate of 0.8 (replace 160 parent chromosomes from 200).

The whole process of the GA to find the optimal C and ε parameters is shown in Figure 5.

7. After obtaining the optimal C and ε parameters from the GA, apply the 2 parameters to create the model through the SVR algorithm, and the final product is the GASVR model.

8. Test the accuracy of the GASVR model by predicting the 5 observed values in January to May 2015. The forecasting values are compared against the actual values of the validation set to compute forecasting errors.

9. Measure the forecasting accuracy by using the RMSE in Equation 9 and the MAPE as shown in Equation 10 (Bergmeir and Benítez, 2012). The RMSE and MAPE are also used to compare with the ANN and ARIMA in later steps of the experimentation.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n} | \; 100\,\frac{y_t - \hat{y}_t}{y_t} \qquad (10)$$

ANN Model

An artificial neural network (ANN) is the simulated network of neurons in the human brain. We can apply an ANN to forecast time series data. The ANN performs learning from the existing data by analyzing the correlation between the observed values at the current time with previously observed values. After getting an ANN model from the training set, we can use such a model to forecast the values of new observations and can measure the accuracy by comparing with the actual values in the validation set.

In this research, we use an ANN model

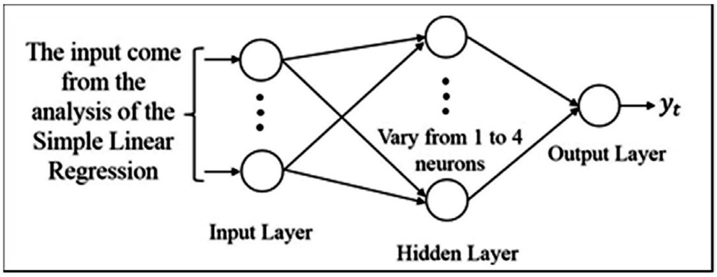

The input come from the analysis of the Simple Linear Regression
y_t
Vary from 1 to 4 neurons
Output Layer
Input Layer
Hidden Layer

Figure 6. The structure of the ANN model for our experiments

from the network architecture called three-layer feed-forward back-propagation neural networks with 1 hidden layer. The output from a model is a forecasting value at the current time (y_t). The input to a model is the previously observed values at 1 to p time intervals ($y_{t-1}, y_{t-2}, \ldots, y_{t-p}$) and can be represented as a vector. The correlation between the input and output can be shown as in Equation 11 (Wang and Meng, 2012).

$$y_t = \omega_0 + \sum_{j=1}^{q} \omega_j \, g\left(\omega_{0j} + \sum_{i=1}^{p} \omega_{i,j} \, y_{t-i}\right) + e_t \qquad (11)$$

where ω_j (j=1,…,q) and $\omega_{i,j}$ (i=0,…, p; j=1,…, q) are the model parameters, which are called weights, p is the number of neurons in the input layer, and q is the number of neurons in the hidden layer. Each neuron uses a sigmoid function (presented in Equation 12) as a transfer function (Wang and Meng, 2012).

$$sig(x) = \frac{1}{(1+exp(-x))} \qquad (12)$$

From the ANN model shown in Equation 11, we can transform to a non-linear function to represent the relationship between previously observed values ($y_{t-1}, y_{t-2}, \ldots, y_{t-p}$) and the forecasting value (yt), as in Equation 13 (Wang and Meng, 2012).

$$y_t = f(y_{t-1}, \ldots, y_{t-p}, \omega) + e_t \qquad (13)$$

where ω is a vector of all the parameters, and f() is a function used for determining the network structure and the connecting of weights. As a result, the neural network model is equivalent to and can be expressed in terms of a non-linear autoregressive model. The output layer will have 1 neuron because it will be used to predict 1 future value in a single time period. In our research, the input of the ANN will be analyzed by simple linear regression.

This research defines the optimal ANN model by testing with 4 ANN models. Each model has a different number of neurons in a hidden layer varying from 1 to 4 neurons. The structure of the final ANN model used for the experiment in this section can be shown as in Figure 6.

We then implement our ANN structures using Matlab software and the steps of implementation can be described as follows:

1. Use the data in Figure 3 to train the 4 ANN models.
2. The inputs of all the ANN models come from the simple linear regression analysis.
3. Normalize the data set to be in the range of 0 to 1 to avoid the data overflow because the process of the ANN is a time-consuming iterative process.
4. Define the structure of the ANN by using the "newff()" function.
5. Fix the initial weight and bias instead of randomizing them. The reason for having specific values is that there will be a rapid convergence to a good solution.
6. Train the 4 ANN models by using the training data in Figure 3.
7. Predict the 5 EPDU values from January to May 2015 from all the ANN models

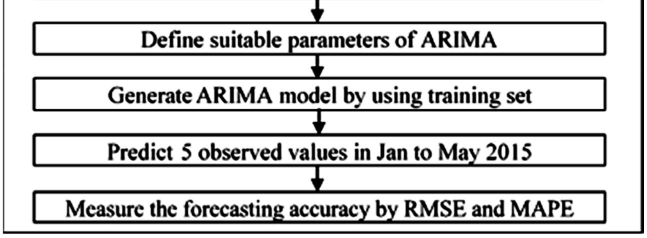

Collected data from the monthly report
Transform data to be time series
Define suitable parameters of ARIMA
Generate ARIMA model by using training set
Predict 5 observed values in Jan to May 2015
Measure the forecasting accuracy by RMSE and MAPE

Figure 7. The steps to generate the ARIMA model

for accuracy measurement with the RMSE and MAPE metrics.

8. Choose the optimal ANN model that has the minimum RMSE and MAPE. Later on we will use the optimal ANN to compare its forecasting performance with the GASVR and ARIMA.

ARIMA Model

The ARIMA model is a model derived from the Box and Jenkins method (Box and Jenkins, 1990). The ARIMA model can predict the data through the conformation of autoregressive (AR) and moving average (MA); the compound of these 2 conformations is thus called ARMA. The ARMA is used for the stationary process of a time series if that series is non-stationary. The transformation to a stationary one is a necessary process before generating the model, called the ARIMA model. The general term of the ARIMA (Wang *et al*., 2012) can be presented with a backward shift operator (B) in the following equation.

$$\theta_P(B)\Theta_P(B^S)(1-B)^d(1-B^S)^D Y_t = w_q(B)W_Q(B^S)a_t \qquad (14)$$

where $BY_t = Y_{t-1}$ and $B^kY_t = Y_{t-k}$

$$w_q(B) = 1 - w_1B - w_2B^2 - \cdots - w_qB^p$$
$$\theta_p(B) = 1 - \theta_1B - \theta_2B^2 - \cdots - \theta_pB^p$$
$$\Theta_P(B^S) = 1 - \Theta_1B^S - \Theta_2B^{2S} - \cdots - \Theta_PB^{PS}$$
$$W_Q(B^S) = 1 - W_1B^S - W_2B^{2S} - \cdots - W_QB^{QS}$$

In order to generate an ARIMA model with R language, we used the "arima()" function in the "forecast" package that was introduced by Hyndman (2015) and which is available in the CRAN repository to create the ARIMA model. We have to analyze the time series data for defining the suitable parameters of the ARIMA(p,d,q)x(P,D,Q)S. That means selecting the suitable p, d, q from trend and P, D, Q from the seasonal. In the "forecast" package, it has the "auto.arima()" function to assign a suitable ARIMA(p,d,q)x(P,D,Q)S. From our conceptual framework, we design the time series forecasting with an ARIMA model as shown in Figure 7. Each step can be described as follows.

1. Collect the 65 data instances from the monthly reports of the MEA.

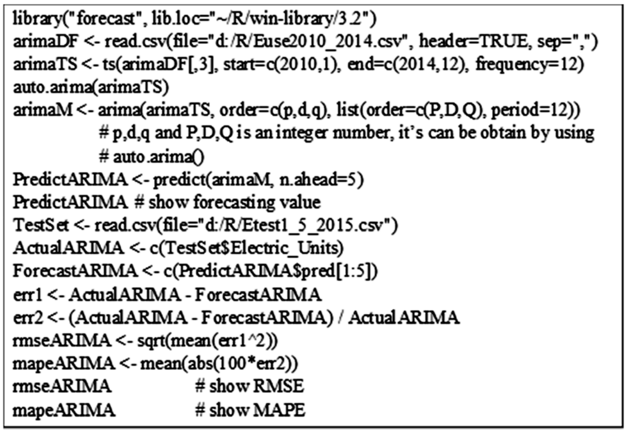
2. Transfer data to be a time series by using the "ts()" function and split it into 2 parts. The first part is the training data: January 2010 to December 2014. The second part is the validation data: January to May 2015.

3. Define the suitable parameters of the ARIMA (the parameters p, d, q and P, D, Q) by using the "auto.arima()" function.

4. Generate the ARIMA model by using the "arima()" function.

5. Predict the 5 observed values from January to May 2015 by using the "predict()" function.

6. Measure the forecasting accuracy with the RMSE and MAPE metrics.



```
library("forecast", lib.loc="~/R/win-library/3.2")
arimaDF <- read.csv(file="d:/R/Euse2010_2014.csv", header=TRUE, sep=",")
arimaTS <- ts(arimaDF[,3], start=c(2010,1), end=c(2014,12), frequency=12)
auto.arima(arimaTS)
arimaM <- arima(arimaTS, order=c(p,d,q), list(order=c(P,D,Q), period=12))
            # p,d,q and P,D,Q is an integer number, it's can be obtain by using
            # auto.arima()
PredictARIMA <- predict(arimaM, n.ahead=5)
PredictARIMA # show forecasting value
TestSet <- read.csv(file="d:/R/Etest1_5_2015.csv")
ActualARIMA <- c(TestSet$Electric_Units)
ForecastARIMA <- c(PredictARIMA$pred[1:5])
err1 <- ActualARIMA - ForecastARIMA
err2 <- (ActualARIMA - ForecastARIMA) / Actual ARIMA
rmseARIMA <- sqrt(mean(err1^2))
mapeARIMA <- mean(abs(100*err2))
rmseARIMA            # show RMSE
mapeARIMA            # show MAPE
```

Figure 8. The command set in R language to generate the ARIMA model

The command set in R according to the steps presented in Figure 7 can be shown as in Figure 8.

Results and Discussion

Due to this research having 3 different models, we divided the experimental results into 4 parts. The first 3 parts are the results obtained from the 3 models (GASVR, ANN, and ARIMA). The last part of the results section is the comparison between the 3 models.

Results of GASVR Model

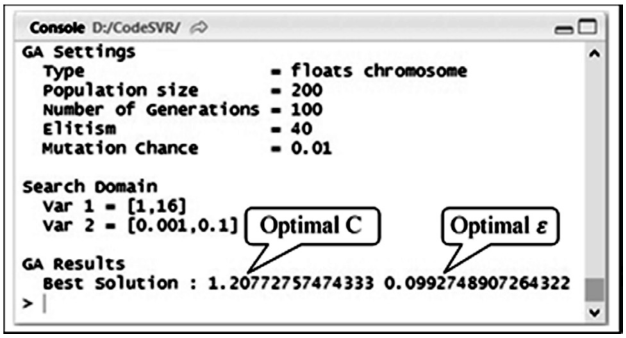
The input vectors of the GASVR model were defined by the simple linear regression analysis of the 1 to 12 lag time observed values and use the $R^2 \geq 0.5$ to determine the input vectors. The result of the input vector determination can be shown in Figure 9.

From Figure 9, the observed values at the 1 and 12 lag times have R^2 greater than 0.5, so we use y_{t-1} and y_{t-12} to be the input of the GASVR model. Later on, we use the GA to find the optimal C and ε parameters that can be shown in Figure 10.

When we know the optimal C and ε parameters, we can thus use them to generate the optimal GASVR model by using the command set in Figure 11. We then predict the 5 observed values and measure the RMSE and MAPE error values that can be shown in Table 1.

Results of ANN Models

For the experiment with the ANN model, we use the structure of the ANN, as shown in

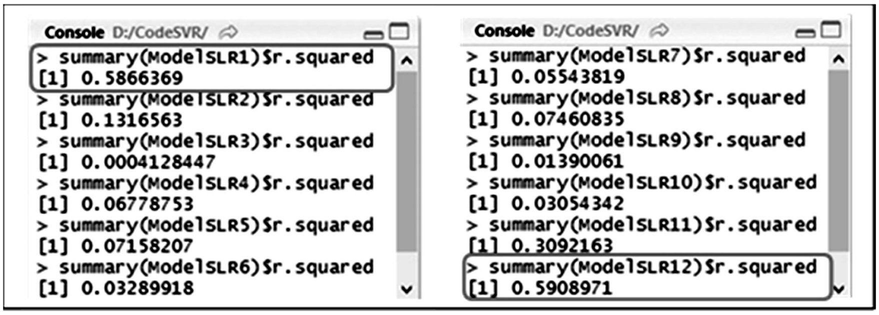


```
GA Settings
  Type                  = floats chromosome
  Population size       = 200
  Number of Generations = 100
  Elitism               = 40
  Mutation Chance       = 0.01

Search Domain
  Var 1 = [1,16]
  Var 2 = [0.001,0.1]

GA Results
  Best Solution : 1.20772757474333 0.0992748907264322
> |
```

Figure 9. R^2 of the 1 to 12 lag time observed values



```
> summary(ModelSLR1)$r.squared
[1] 0.5866369
> summary(ModelSLR2)$r.squared
[1] 0.1316563
> summary(ModelSLR3)$r.squared
[1] 0.0004128447
> summary(ModelSLR4)$r.squared
[1] 0.06778753
> summary(ModelSLR5)$r.squared
[1] 0.07158207
> summary(ModelSLR6)$r.squared
[1] 0.03289918
```

```
> summary(ModelSLR7)$r.squared
[1] 0.05543819
> summary(ModelSLR8)$r.squared
[1] 0.07460835
> summary(ModelSLR9)$r.squared
[1] 0.01390061
> summary(ModelSLR10)$r.squared
[1] 0.03054342
> summary(ModelSLR11)$r.squared
[1] 0.3092163
> summary(ModelSLR12)$r.squared
[1] 0.5908971
```

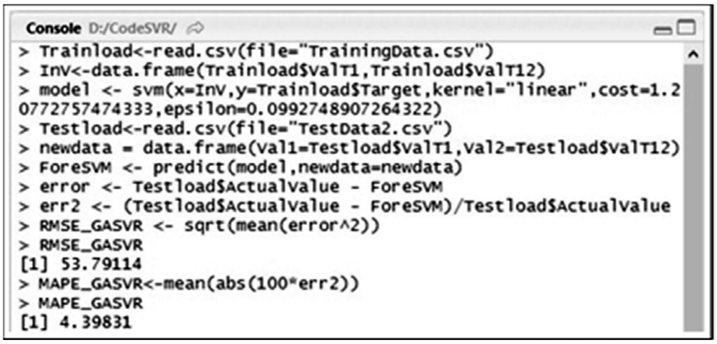
Figure 10. The optimal C and ε parameters

Table 1. Predicted values and error of the GASVR model

Value	Jan 2015	Feb 2015	Mar 2015	Apr 2015	May 2015	RMSE	MAPE
Actual Value	745.82	853.60	1017.89	1100.93	1216.77	-	-
GASVR	758.43	776.75	931.54	1097.62	1186.19	53.79	4.40

Figure 6, and use y_{t-1} and y_{t-12} as an input, which is the same as the GASVR. To avoid over-fitting, we design the ANN structure to contain a few neurons (from 1 to 4) in the hidden layer. The 4 ANN models are named ANN1 to ANN4. In addition, we found that by fixing the values of the initial weight and bias, the ANN shows a better performance than by randomizing them. From our experiment, the initial weight and bias of each ANN model that minimizes the RMSE and MAPE can be shown as in Table 2.

When we use the initial weight and bias for training the ANN models and then measure their RMSE and MAPE values with the validation data set, the results are as shown in Table 3. The ANN3 model has a minimum



```
> Trainload<-read.csv(file="TrainingData.csv")
> InV<-data.frame(Trainload$ValT1,Trainload$ValT12)
> model <- svm(x=InV,y=Trainload$Target,kernel="linear",cost=1.2
0772757474333,epsilon=0.0992748907264322)
> Testload<-read.csv(file="TestData2.csv")
> newdata = data.frame(Val1=Testload$ValT1,Val2=Testload$ValT12)
> ForeSVM <- predict(model,newdata=newdata)
> error <- Testload$ActualValue - ForeSVM
> err2 <- (Testload$ActualValue - ForeSVM)/Testload$ActualValue
> RMSE_GASVR <- sqrt(mean(error^2))
> RMSE_GASVR
[1] 53.79114
> MAPE_GASVR<-mean(abs(100*err2))
> MAPE_GASVR
[1] 4.39831
```

Figure 11. The command set in R language to generate the optimal GASVR model

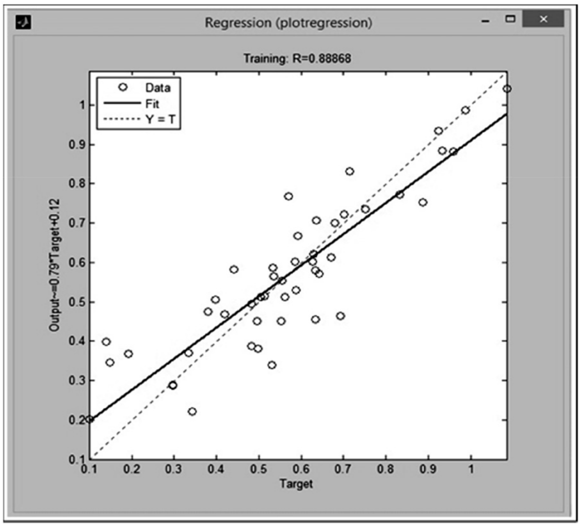


Figure 12. Regression graph of ANN3

Table 2. The initial weight and bias of each ANN model

Model	net.iw{1,1}	net.lw{2,1}	net.b{1}	net.b{2}
ANN1	[0.5 0.5]	[0.5]	[0.5]'	[0.5]'
ANN2	[0.5 0.5;0.1 0.1]	[0.4 0.4]	[0.5 0.5]'	[0.5]'
ANN3	[0.5 0.4;0.5 0.5;0.5 0.5]	[0.5 0.5 0.5]	[0.5 0.5 0.5]'	[0.5]'
ANN4	[0.5 0.4;0.5 0.5;0.5 0.5;0.5 0.5]	[0.5 0.5 0.5 0.5]	[0.5 0.5 0.5 0.5]'	[0.5]'

MAPE value and when we measure the correlation of each model, we found that the ANN3 model has a maximum correlation coefficient equal to 0.88868. Its regression graph can be shown as in Figure 12. Therefore, we can conclude that the ANN3 model is the optimal ANN model for this research and it has a structure as shown in Figure 13.

Results of ARIMA Model

After exploring with the auto.arima() function to find the suitable parameters for the ARIMA(p, d, q)x(P, D, Q)S, we found that the order of the autoregressive (AR) should be p = 1, the order of the seasonal autoregressive (SAR) should be P = 1, the order of the difference should be d = D = 0, and the order of the moving average should be q = Q = 0. When we estimate the coefficient of AR1 (θ_1) and SAR1 ($\widehat{\Theta}_1$) by the auto.arima() function, we get the values of 0.7301 and 0.6403, respectively. Therefore, the forecasting equations can be derived as follows:

$$(1-\hat{\theta}_1B)(1-\widehat{\Theta}_1B^{12})Y_t=\theta_0$$

while θ_0 is constant

$$(1-\hat{\theta}_1B-\widehat{\Theta}_1B^{12}+\hat{\theta}_1B\widehat{\Theta}_1B^{12})Y_t=\theta_0$$
$$Y_t-\hat{\theta}_1Y_{t-1}-\widehat{\Theta}_1Y_{t-12}+\hat{\theta}_1\widehat{\Theta}_1Y_{t-13}=\theta_0$$
$$Y_t=\theta_0+\hat{\theta}_1Y_{t-1}+\widehat{\Theta}_1Y_{t-12}-\hat{\theta}_1\widehat{\Theta}_1Y_{t-13}$$
$$Y_t=\theta_0+0.73Y_{t-1}+0.64Y_{t-12}-0.47Y_{t-13}$$

From the existing observed values of the time series, $\theta_0 = 92.42$ and the the forecasting computation of the ARIMA can be presented as in Equation 15.

$$Y_t=92.42+0.73Y_{t-1}+0.64Y_{t-12}-0.47Y_{t-13} \tag{15}$$

We use the ARIMA model to predict the 5 observed values for the measurement of the RMSE and MAPE. The results of the experiment are shown in Table 4.

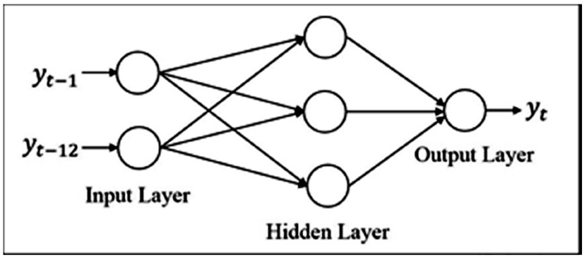


Figure 13. The structure of ANN3

Table 3. Predicted values and errors for the ANN models

Value	Jan	Feb	Mar	Apr	May	RMSE	MAPE
Actual Value	745.82	853.60	1017.89	1100.93	1216.77	-	-
ANN1	786.94	789.21	911.85	1096.59	1190.53	59.67	5.21
ANN2	785.26	809.44	905.99	1123.58	1193.65	58.47	5.08
ANN3	753.33	816.63	890.15	1148.87	1207.41	63.46	4.58
ANN4	759.61	760.58	913.55	1089.92	1184.74	64.66	5.33

Table 4. Predicted values and errors of the ARIMA model

Value	Jan	Feb	Mar	Apr	May	RMSE	MAPE
Actual Value	745.82	853.60	1017.89	1100.93	1216.77	-	-
ARIMA	799.24	859.33	969.4	1065.12	1117.63	57.18	4.80

The comparison of the 3 models: GASVR, ANN, and ARIMA

From our experiment, we found that the input vector selection for the GASVR and ANN models by means of simple regression analysis resulted in the input vectors of the observed values at the 1 and 12 lag time (y_{t-1} and y_{t-12}). For building the ARIMA model, the conformation of the ARIMA model is (p, d, q) x(P, D, Q) = (1, 0, 0)x(1, 0, 0). This means that the output value at time (yt) depends on 2 inputs that are the observed values at the 1 and 12 lag time (y_{t-1} and y_{t-12}). Therefore, the input of the 3 different models generated from 3 different methods in our experiment has the same set of input vectors.

By predicting the electricity usage during the period from January to May 2015, we found that the GASVR model yielded the lowest RMSE and MAPE (RMSE = 53.79 and MAPE = 4.40). This accuracy performance is about 8.33% better than the ARIMA model. Therefore, we can conclude that the GASVR model is the most suitable model for power consumption forecasting. The comparative results of the 3 models are summarized in Table 5, and also are graphically shown in Figure 14. From Figure 14, we found that the forecasting trends of the GASVR and ARIMA models are similar and align to the actual values, whereas the forecasting graph of the ANN3 model is worse than the other 2 models. But the graph of the ANN3 model can rapidly adjust the irregular error in March to a normal error.

From the accuracy comparison results of the GASVR, ANN, and ARIMA models, the GASVR has the lowest RMSE and MAPE. This may be due to the fact that the power

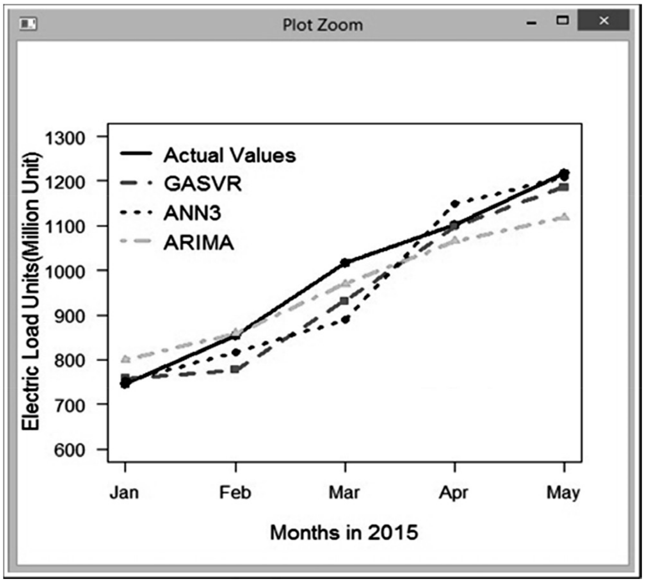


Figure 14. The forecasting graph of the 3 models

Table 5. Predicted values and errors of the 3 models

Value	Jan	Feb	Mar	Apr	May	RMSE	MAPE	Increasing
Actual Value	745.82	853.60	1017.89	1100.93	1216.77	-	-	-
GASVR	758.43	776.75	931.54	1097.62	1186.19	53.79	4.40	+8.33%
ANN3	753.33	816.63	890.15	1148.87	1207.41	63.46	4.58	+4.58%
ARIMA	799.24	859.33	969.4	1065.12	1117.63	57.18	4.80	0%

consumption time series consists of complex linear and non-linear patterns, which are difficult to forecast correctly. The ARIMA model is good at capturing linear patterns, but it cannot easily capture the non-linear patterns (Pai and Lin, 2005; Wang and Meng, 2012). Conversely, the machine learning techniques (ANN and GASVR) can capture non-linear patterns quite well, but they may not easily capture the linear patterns (Wang and Meng, 2012; Zhang *et al.,* 2016). Neither the ARIMA nor machine learning technique alone is adequate in modeling and predicting time series data that consist of linear and non-linear patterns. Therefore, we integrate both techniques by determining the inputs of the machine learning techniques using the lag time observed values that have R^2 greater than 0.5, which can represent linear patterns. We then generate the forecasting model by using the machine learning technique that can efficiently capture the non-linear patterns.

With this underlying assumption, the ANN and GASVR models should capture both linear and non-linear patterns. The correctness of this assumption has been experimentally confirmed through the lower MAPE values in the ANN and GASVR models, as compared with the ARIMA model. When we compare the ANN with the GASVR, the GASVR has a lower MAPE than the ANN because the GASVR is more suitable to the small data size than the ANN. This observation is in accordance with the result observed by other researchers (Zhang *et al*., 2016).

Conclusions

We studied the time series analysis for forecasting the electrical power distribution units (EPDU) using monthly data reported by the Metropolitan Electricity Authority from January 2010 to May 2015. The specific part of the data used in this research is the EPDU of household consumers. We generated 3 different models to predict the future EPDU by using 3 different techniques: genetic algorithm optimized support vector regression (GASVR), artificial neural network (ANN), and autoregressive integrated moving average (ARIMA). For the modeling of the GASVR and ANN, the data set must be in the form of a pair between the input vectors and targets $(D = \{(x_i, y_i)\})_{i=1}^{n}$). The input vectors are the observed values at the lag time 1 to 12 (y_{t-1}, y_{t-2}, …, y_{t-12}) and the targets are the observed values at time t (y_t). We defined the input vectors of the GASVR and ANN models by using simple linear regression analysis and the result reveals that the observed values at the 1 and 12 lag time (y_{t-1} and y_{t-12}) are the appropriate inputs for the GASVR and ANN models. For the ANN model, the result reveals that the ANN3 model that has 3 neurons in the hidden layer is the most suitable model. For the ARIMA model, the suitable conformation is ARIMA(1,0,0)x(1,0,0) 12 that means the output yt depends on the 2 inputs at the 1 and 12 lag time (y_{t-1} and y_{t-12}). These lagging periods are similar to the inputs of the GASVR and ANN models. Finally, we compared the performance of the 3 models using the same validation data set; we found that the GASVR model has the lowest RMSE and MAPE values because the GASVR model can capture both linear and non-linear patterns of the time series; this is the reason for its better performance when being compared with the ARIMA model. The GASVR model is also suitable to a small data set when compared with the ANN model that shows a lower performance than the GASVR model using the same training and testing data sets. Therefore, we can conclude that the idea proposed in this paper to extend the SVR technique by using the GA to optimize the parameters C and ε prior to the application of SVR yields the best method for modelling the power consumption time series.

Acknowledgements

This work is the part of learning in the doctoral program of computer engineering in the School of Computer Engineering, Suranaree University of Technology, Thailand. We are grateful to the Metropolitan Electricity Authority of Thailand for publicizing the data that had been used in this study.

References

Bagheripour, P., Gholami, A., Asoodeh, M., and Asadi, M.V. (2015). Support vector regression based determination of shear wave velocity. J. Petrol. Sci. Eng., 125:95-99.

Bergmeir, C. and Benítez, J.M. (2012). On the use of cross-validation for time series predictor evaluation. Inform. Sciences, 191:192-213.

Box, G.E.P. and Jenkins, G. (1990). Time Series Analysis, Forecasting and Control. Holden-Day, San Francisco, CA, USA, 500p.

Bunn, D.W. and Farmer, E.D. (1985). Comparative Models for Electrical Load Forecasting. John Wiley and Sons, New York, NY, USA, 232p.

Fan, G.F., Peng, L.L., Hong, W.C., and Fan, S. (2016). Electric load forecasting by the SVR model with differential empirical mode decomposition and autoregression. Neurocomputing, 173:958-970.

Hyndman, R.J. (2015). Forecasting Functions for Time Series and Linear Models. Melbourne, Australia: Monash University. Available from: http://github.com/robjhyndman/forecast. Accessed date: Sep 3, 2015.

Jirong, G., Mingcang, Z., and Liuguangyan, J. (2011). Housing price forecasting based on genetic algorithm and support vector machine. Expert Syst. Appl., 38:3,383-3,386.

Kaytez, F., Taplamacioglu, M.C., Cam, E., and Hardalac, F. (2015). Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. Electrical Power and Energy Systems, 67:431-438.

Lee, Y.S. and Tong, L.I. (2011). Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. Knowledge-Based Systems, 24:66-72.

Metropolitan Electricity Authority. (2015). Reports of electrical distribution. Bangkok, Thailand. Available from: http://www.mea.or.th/download/index.php#. Accessed date: Aug 12, 2015.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.C., and Lin, C.C. (2015). Misc Functions of the Department of Statistics, Probability Theory Group. Vienna, Austria: Vienna University of Technology. Available from: http://cran.r-project.org/web/packages/e1071/index.html. Accessed date: Dec 5, 2015.

Ogcu, G., Demirel, O.F., and Zaim, S. (2012). Forecasting electricity consumption with neural networks and support vector regression. Procedia-Social and Behavioral Sciences, 58:1576-1585.

Pai, P.F. and Lin, C.S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. Omega, 33:497-505.

Sajan, K.S., Kumar, V., and Tyagi, B. (2015). Genetic algorithm based support vector machine for on-line voltage stability monitoring. Electrical Power and Energy Systems, 73:200-208.

Wang, J., Li, L., Niu, D., and Tan, Z. (2012a). An annual load forecasting model based on support vector regression with differential evolution algorithm. Appl. Energ., 94:65-70.

Wang, X. and Meng, M. (2012). A Hybrid Neural Network and ARIMA Model for Energy Consumption Forecasting. Journal of Computers, 7(5):1,184-1,190.

Wang, Y., Wang, J., Zhao, G., and Dong, Y. (2012b). Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China. Energ. Policy, 48:284-294.

Willighagen, E. and Ballings, M. (2015). R Based Genetic Algorithm. CRAN Repository. Available from: http://github.com/egonw/genalg. Accessed date: Dec 5, 2015.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. 2nd ed. Springer-Verlag, New York, Ny, USA, 314p.

Zhang, F., Deb, C., Lee, S.E., , Yang, J., and Shah, K.W. (2016). Time series forecasting for building energy consumption usingweighted Support Vector Regression with differential evolution optimization technique. Energ. Buildings, 126:94-103.