Artificial Neural Networks and Time Series Models for Electrical Load Analysis

Ronnachai Chuentawat*, Supaporn Bunrit, Chanintorn Ruangudomsakul, Nittaya Kerdprasop, and Kittisak Kerdprasop

Abstract—**We applied traditional time series analysis and artificial neural network (ANN) techniques to model and forecast the power consumption of Bangkok's metropolitan area. Time series data in terms of units of household electricity usage were obtained from the Metropolitan Electricity Authority of Thailand. The data had been collected monthly from January 2010 to May 2015. Forecasting models with different parameters are generated from both techniques using the training data, which are the series from January 2010 to December 2014. The remaining data from January 2015 to May 2015 are employed as the testing data. Forecasting performance of each model is measured by the rooted mean square error (RMSE) and the mean absolute percentage error (MAPE) metrics. The traditional time series forecasting models studied in this research are GLM, HoltWinters, and ARIMA. For ANN, we examine four models using 3 layers with different number of neurons ranging from 4 to 7: 3L-4N, 3L-5N, 3L-6N, and 3L-7N. The experimental results reveal that ARIMA is superior among the traditional time series models. For the intelligent based models, 3L-6N is the best of ANN models. Moreover, the MAPE metric of the 3L-6N model is less than the ARIMA model. As a result, we can conclude that ANN model is more powerful in forecasting power distribution units than the traditional time series models.**

Index Terms—**Artificial neural networks, forecasting, time series model, electrical load analysis**

I. Introduction

THIS research aims to perform time series analysis to forecast future data by using electricity supply information of Metropolitan Electricity Authority of Thailand. It is a monthly reported data from January 2010 to May 2015. The data are presented in unit metric in which 1 unit refers to 1000 kilowatts per hour. These data contain the time series pattern.

An accurate forecast on the demand of electricity provides the advantageous information in resource planning, management of funding, and reducing the operation cost. Bunn and Farmer [2] found and reported that 1% of forecasting error raised by 10 million units of the operation costs. Therefore, the precision of forecast is a challenging problem, especially on the demand of electricity supply.

Traditional time series forecasting technique generates the forecasting model from the training data set by determining the unrecognized parameters. In our study, simple linear regression analysis, triple exponential smoothing and autoregressive integrated moving average (ARIMA) of Box and Jenkins method are applied. The acquired forecasting models from the three methods are then compared the accuracy using the two metrics: rooted mean square error (RMSE) and mean absolute percentage error (MAPE). The lower RMSE and MAPE, the more accurate the model is.

Artificial neural network (ANN) is a machine learning technique that mimics the concept of neuron operation in human brain. It can be used to forecast time series data by setting the output neuron to provide the observed value at time *t* (y_t). Input is the observed value of lag time from 1 to *p* period prior to the time *t* (y_{t-1}, y_{t-2}, …, y_{t-p}). In this study, we used three-layer feed-forward back propagation neural network for time series analysis. We compared each model to find the most suitable one with the lowest RMSE and MAPE.

Our research is thus a comparative study of artificial neural networks and traditional time series analysis for forecasting time series data. Research objectives are:

1. Apply three traditional time series analysis methods for generating forecasting model using the R language. Such methods are generalized linear models (GLM), Holt-Winters (HoltWinters) and autoregressive integrated moving average (ARIMA).
2. Apply ANN models for time series analysis by generating forecasting models in Matlab.
3. Find the most suitable forecasting model by comparing RMSE and MAPE resulting from all traditional time series analysis methods and ANN models.

II. Literature Review

There are a number of related works in literatures about time series analysis using traditional time series methods and ANNs.

Maçaira, Souza and Oliveira [3] studied about how to define forecasting model and forecast yearly electricity consumption of residential units in Brazil until 2050 using

Manuscript December 26, 2015; revised January 18, 2016. This work has been supported in part by grants from Suranaree University of Technology through the funding of the Data Engineering Research Unit.

R. Chuentawat is a doctoral student with the School of Computer Engineering, Suranaree University of Technology, and also an assistant professor at Nakhon Ratchasima Rajabhat University, Nakhon Ratchasima 30000, Thailand (e-mail: c_ronnachai@hotmail.com).

S. Bunrit is a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: sbunrit@sut.ac.th).

C. Ruangudomsakul is a doctoral student with the School of Computer Engineering, Suranaree University of Technology, and also a lecturer at Sisaket Rajabhat University, Thailand (e-mail: chanintorn.r@sskru.ac.th).

N. Kerdprasop is an associate professor with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: nittaya@sut.ac.th).

K. Kerdprasop is an associate professor with the School of Computer Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand (e-mail: kerdpras@sut.ac.th).

Pegels exponential smoothing forecasting technique. They performed parameter adjustment for forecasting data along with the estimated value that is available in PDE (the ten year energy planning) and PNE (the nation energy planning). The result was that the Pegels exponential smoothing forecasting technique produced the nearest estimation of electrical consumption in both PDE and PNE.

Keka and Hamiti [4] did experiment to find mathematical model representing relationship between electrical energy and time by using linear regression techniques. They used electrical supply data collected from the electrical substation every 15 minutes and set the time interval by using day, week and month. Result from the experiment showed a linear mathematical model that can represent the relationship between electrical energy and time.

Chujai, Kerdprasop and Kerdprasop [5] studied time-series analysis of household electrical consumption using data from UCI Machine Learning Repository. They generated forecast models from ARIMA and ARMA using 4 kinds of time interval (day, week, month, and quarter) and experimented with the R language. They performed model tolerant analysis by measuring the AIC (Akaike Information Criterion) and RMSE. The result showed that the ARIMA model is suitable for forecasting data in the month and quarter interval while the ARMA model is suitable for forecasting data in day and week interval.

Wang and Ming [6] studied about forecasting energy consumption in China using ARIMA, ANN and hybrid ARIMA-ANN model. They compared the accuracy using RMSE, MAE (Mean Absolute Error), and MAPE. The result of the study showed that a hybrid model provided better accuracy in forecasting than either the ARIMA, or ANNs.

Firata, Turanb and Yurdusev [7] studied about forecasting water supply consumption in Turkey using several ANN techniques including generalized regression neural networks (GRNN), cascade correlation neural network (CCNN), and feed forward neural networks (FFNN). They defined 6 network structures and then performed statistical testing by measuring the average absolute relative error (AARE), normalized root mean square error (NRMSE), and threshold statistic (Ts). The result showed that the values of AAER and NRMSE of the M5 models are better than those of the other models and the test statistics of the M5 CCNN model is slightly better than those of GRNN and FFNN models.

III. Methodology

A. Traditional Time Series Models

In traditional time series forecasting technique, we apply general linear regression, triple exponential smoothing, and Box and Jenkins methods. We then forecast data for the next 5 intervals. After that, we compare forecasted data against real data in a test set. The RMSE and MAPE metrics are used as statistical measure instruments to find the suitable forecasting model. The 3 traditional forecasting techniques (GLM, HoltWinters, and ARIMA) are described as follows:

1. *GLM* model is the forecasting model that are generated by simple linear regression analysis. GLM linear regression [8] is as shown in equation 1.

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{1}$$

where X is a time-interval variable in a monthly unit,
Y is the amount of monthly power distribution unit,
ε is a residual term.

2. *HoltWinters* model is deriving from triple exponential smoothing. It is a smoothing technique that can be used to generate forecast models from data with trend and seasonal [9]. HoltWinters model can be described as in equations 2-4. To forecast the future event, equation 5 which is composed of the level, trend, and seasonal parts is applied.

$$\textbf{Level :}\ L_t = \alpha(Y_t - S_{t-s}) + (1-\alpha)(L_{t-1} + b_{t-1}) \tag{2}$$

$$\textbf{Trend :}\ b_t = \beta(L_t - L_{t-1}) + (1-\beta)b_{t-1} \tag{3}$$

$$\textbf{Seasonal :}\ S_t = \gamma(Y_{t-1} - L_t) + (1-\gamma)S_{t-s} \tag{4}$$

$$\textbf{Forecast :}\ F_{t+m} = L_t + b_{tm} + S_{t-s+m} \tag{5}$$

where α is constant for level smoothing
β is constant for trend smoothing
γ is constant for seasonal smoothing
L_t is estimated level of time series at time t
Y_t is observed value at time t
b_t is estimated slope of time series at time t
S_t is seasonal factor
s is seasonal length, equal to 12 months (s = 12)
m is forecast intervals
F_{t+m} is forecasted value m intervals

3. *ARIMA* model is a model derived from Box and Jenkins method. General term of ARIMA [10] can be presented with backward shift operator (*B*) as in equation 6.

$$\theta_P(B)\Theta_P(B^S)(1-B)^d(1-B^S)^D Y_t = w_q(B)W_Q(B^S)\alpha_t \tag{6}$$

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

$$w_q(B) = 1 - w_1 B - w_2 B^2 - \cdots - w_q B^p$$

$$\theta_p(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_p B^p$$

$$\Theta_P(B^S) = 1 - \Theta_1 B^S - \Theta_2 B^{2S} - \cdots - \Theta_P B^{PS}$$

$$W_Q(B^S) = 1 - W_1 B^S - W_2 B^{2S} - \cdots - W_Q B^{QS}$$

In order to generate an ARIMA model, we have to analyze time series data for defining suitable parameters of ARIMA(*p*, *d*, *q*)x(*P*, *D*, *Q*)$_S$. That means selecting the suitable *p*, *d*, *q* from trend and *P*,*D*,*Q* from seasonal. In R language, we can use function auto.arima() to assign suitable ARIMA(*p*, *d*, *q*)x(P, D, Q)$_S$. In this research, we use (*p*, *d*, *q*) = (1, 0, 0) and (*P*, *D*, *Q*)$_S$ = (1, 0, 0)$_{12}$, where 1 season = 12 months (S=12). Therefore, a suitable ARIMA(*p*, *d*, *q*)x(*P*, *D*, *Q*)$_S$ is ARIMA(1, 0, 0)x(1, 0, 0)$_{12}$.

After obtaining 3 forecasting models from the 3 methods mentioned above, statistical measuring in terms of RMSE and MAPE is compared for model accuracy. A model with the lowest RMSE and MAPE is the most suitable one. Summary of steps in applying traditional time series analysis to find a suitable forecasting model is shown in figure 1.

Step 1: Collect data from secondary source. Data are obtained from the monthly report of the Metropolitan Electricity Authority from January 2010 to May 2015. The data are the household electrical usage of residents in the Metropolitan Electricity Authority area. We then order the data to build up time series.

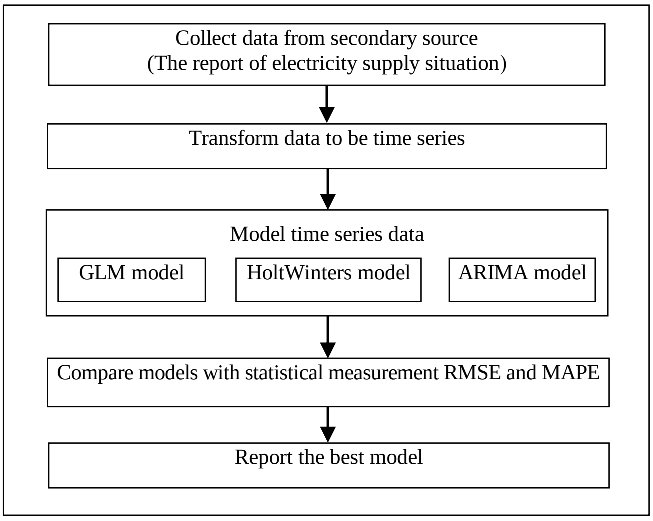
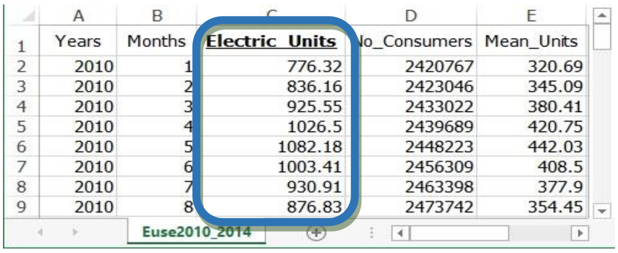

Collect data from secondary source
(The report of electricity supply situation)
Transform data to be time series
Model time series data
GLM model
HoltWinters model
ARIMA model
Compare models with statistical measurement RMSE and MAPE
Report the best model

Fig. 1. Steps in selecting the most appropriated model for forecasting time series data



	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

Fig. 2. Time series data are created from secondary data

```
myDFlr <- read.csv(file="d:/R/EuseLRmodel.csv"
              , header=TRUE, sep=",")
glm.D9 <- glm(myDFlr[,4] ~ myDFlr[,3])
```

(a) create GLM (Generalized linear models)

```
library("forecast", lib.loc="~/R/win-library/3.2")
myDF_HW <- read.csv(file="d:/R/Euse2010_2014.csv",
                   header=TRUE, sep=",")
mytsHW <- ts(myDF_HW[,3], start=c(2010,1),
                  end=c(2014,12), frequency=12)
HWfit <- HoltWinters(mytsHW)
```

(b) create HoltWinters (Holt-Winters)

```
library("forecast", lib.loc="~/R/win-library/3.2")
myDFarima <- read.csv(file="d:/R/ Euse2010_2014.csv",
                   header=TRUE, sep=",")
mytsARIMA <- ts(myDFarima[,3], start=c(2010,1),
                  end=c(2014,12), frequency=12)
auto.arima(mytsARIMA)
ARIMAfit <- arima(mytsARIMA, order=c(1,0,0),
                 list(order=c(1,0,0), period=12))
```

(c) create ARIMA (Autoregressive Integrated Moving Average)

Fig. 3. R commands for generating traditional forecasting model

Step 2: Transform data to be time series. The data from the monthly report of the Metropolitan Electricity Authority is a monthly interval from January 2010 to May 2015 as shown in figure 2. We divide the data into 2 sets. The first is a training set consisting of data from January 2010 to December 2014. It is used for generating the forecasting model. The test set is data from January 2015 to May 2015. Test data are used for validating the model and evaluating by RMSE and MAPE.

Step 3: Model time series data. This step generates 3 different forecasting models from the training set. We build 3 forecasting models from the same training set with commands in R language that can be summarized in figure 3.

Step 4: Compare models with statistical measurements RMSE and MAPE. The computation of RMSE (rooted mean square error) and MAPE (mean absolute percentage error) [11] is shown in equations 7 and 8, respectively.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \qquad (7)$$

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left| 100\,\frac{y_t-\hat{y}_t}{y_t}\right| \qquad (8)$$

where y_t is observed value at time t,
$\hat{y}_t$ is forecasted value at time t, and
n is the number of forecasting period.

Step 5: Report the best model. The best forecasting model is the one with the lowest RMSE and MAPE values.

B. *Artificial Neural Network Model*

Artificial neural networks (ANNs) are the simulated networks of neurons in the human brain. We can apply ANNs to forecast time series data. ANNs perform learning from existing data by analyzing the correlation between observed values at current time with previous observed values. After getting an ANN model from the training set, we can use such model to forecast the value of new observation in the test set.

In this research, we use ANNs model calls three-layer feed-forward back propagation neural networks which has 1 hidden layer. Output from a model is a forecasting value at current time (y_t). Input to a model is the previous observed values at 1 to p time intervals (y_{t-1}, y_{t-2}, …, y_{t-p}) and be represented as a vector. The correlation between input and output [6] can be shown as in equation 9.

$$y_t = \omega_0 + \sum_{j=1}^{q}\omega_j\, g\left(\omega_{0j} + \sum_{i=1}^{p}\omega_{i,j}\, y_{t-i}\right) + e_t \qquad (9)$$

where ω_j $(j=1,\ldots,q)$ *and* $\omega_{i,j}$ $(i=0,\ldots,p;\ j=1,\ldots,q)$ are model parameters which are called weights ,
p is the number of neurons in the input layer,
q is the number of neurons in the hidden layer and have sigmoid function [6] as a transfer function presented in equation 10.

$$sig(x) = \frac{1}{(1+\exp(-x))} \qquad (10)$$

From ANNs model shown in equation 9, we can transform to nonlinear function to represent the relationship [6] between previous observed values (y_{t-1}, y_{t-2}, …, y_{t-p}) and forecasting value (y_t) as in equation 11.

$$y_t = f(y_{t-1},\ldots,y_{t-p},\omega) + e_t \qquad (11)$$

where ω is a vector of all 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 nonlinear autoregressive model.

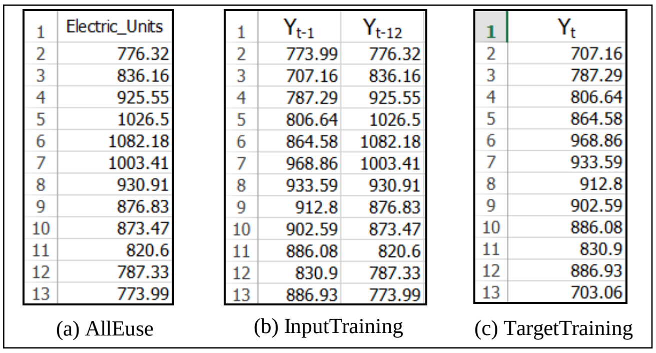
In the research on Wang and Meng [6], they suggested that ARIMA and ANNs have something similar. From our ARIMA model, it has p = 1 and P = 1 which means observation at time t (y_t) relates to the previous observation with a lag of one interval (y_{t-1}) and relates to previous observation with 1 season lag or 12 intervals (y_{t-12}). So, we can use y_{t-1} and y_{t-12} as the input to ANNs model. Therefore, input layer of ANNs model has 2 neurons. In order to investigate the complexity of an ANN causing the over-fitting problem, we experiment with 4 ANNs models defined as follows:

Model 1: Three layers, two input neurons, one hidden neuron and one output neuron (3L-4N).

Model 2: Three layers, two input neurons, two hidden neurons and one output neuron (3L-5N).

Model 3: Three layers, two input neurons, three hidden neurons and one output neuron (3L-6N).

Model 4: Three layers, two input neurons, four hidden neurons and one output neuron (3L-7N).



1	Electric_Units
2	776.32
3	836.16
4	925.55
5	1026.5
6	1082.18
7	1003.41
8	930.91
9	876.83
10	873.47
11	820.6
12	787.33
13	773.99

(a) AllEuse

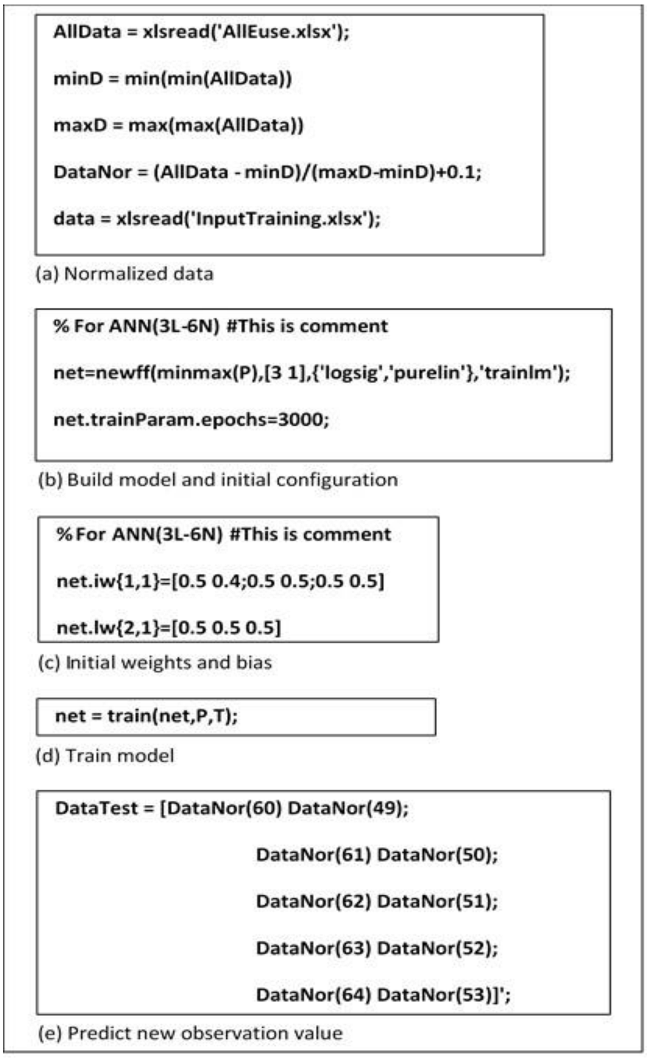
1	Y_{t-1}	Y_{t-12}
2	773.99	776.32
3	707.16	836.16
4	787.29	925.55
5	806.64	1026.5
6	864.58	1082.18
7	968.86	1003.41
8	933.59	930.91
9	912.8	876.83
10	902.59	873.47
11	886.08	820.6
12	830.9	787.33
13	886.93	773.99

(b) InputTraining

1	Y_t
2	707.16
3	787.29
4	806.64
5	864.58
6	968.86
7	933.59
8	912.8
9	902.59
10	886.08
11	830.9
12	886.93
13	703.06

(c) TargetTraining

Fig. 4. Examples of data file for generating ANNs model



```
AllData = xlsread('AllEuse.xlsx');

minD = min(min(AllData))

maxD = max(max(AllData))

DataNor = (AllData - minD)/(maxD-minD)+0.1;

data = xlsread('InputTraining.xlsx');
```

(a) Normalized data

```
% For ANN(3L-6N) #This is comment

net=newff(minmax(P),[3 1],{'logsig','purelin'},'trainlm');

net.trainParam.epochs=3000;
```

(b) Build model and initial configuration

```
% For ANN(3L-6N) #This is comment

net.iw{1,1}=[0.5 0.4;0.5 0.5;0.5 0.5]

net.lw{2,1}=[0.5 0.5 0.5]
```

(c) Initial weights and bias

```
net = train(net,P,T);
```

(d) Train model

```
DataTest = [DataNor(60) DataNor(49);

                        DataNor(61) DataNor(50);

                        DataNor(62) DataNor(51);

                        DataNor(63) DataNor(52);

                        DataNor(64) DataNor(53)]';
```

(e) Predict new observation value

Fig. 5. Command sets in Matlab for generating ANNs model

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

Step 1. Create 3 data files with Excel (figure 4) for generating models. The first file is *AllEuse*; it contains a power distribution unit data from January 2010 to May 2015. The second file is *InputTraining*; it contains a training set as input data consisting of y_{t-1} and y_{t-12}. The third file is *TargetTraining*; it contains an output data of the training data set y_t .

Step 2. Perform data normalization in *InputTraining* and *TargetTraining* to adjust the data values into the interval of 0 and 1 using a command set in figure 5(a).

Step 3. Create ANNs by using newff() function and initial configuration using a command set in figure 5(b).

Step 4. Define initial weights and bias as random constants using a command set in figure 5(c).

Step 5. Perform training model by using normalized data (*P* and *T*) with a command set in figure 5(d).

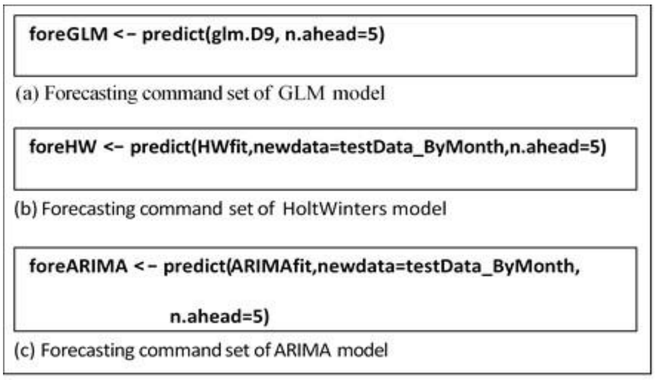
Step 6. Forecast 5 intervals of power distribution units from January 2015 to May 2015. Then calculate RMSE and MAPE by comparing forecasted data against the real data using a command set in figure 5(e).

IV. Experimental Evaluation

We aims at forecasting power distribution units of household electrical usage of residents in the Metropolitan Electricity Authority area. We apply 2 different techniques: traditional time series forecasting technique by using R language and time series forecasting with ANNs by using Matlab. We, therefore, branch the results by experiments and their comparison into 3 sections.

A. *Results of Traditional Time Series Models*

We generate forecasting models by using 3 traditional time series forecasting models, then use all 3 models to forecast power distribution unit in the next 5 intervals from January 2015 to May 2015. After that, we take the predicted values to calculate RMSE and MAPE by comparing with actual values using the R command set in figure 6. The result of experiments is shown in table 1.



```
foreGLM <- predict(glm.D9, n.ahead=5)
```

(a) Forecasting command set of GLM model

```
foreHW <- predict(HWfit,newdata=testData_ByMonth,n.ahead=5)
```

(b) Forecasting command set of HoltWinters model

```
foreARIMA <- predict(ARIMAfit,newdata=testData_ByMonth,

                  n.ahead=5)
```

(c) Forecasting command set of ARIMA model

Fig. 6. Forecasting command set in R language for traditional time series model

TABLE I
PREDICTED VALUE AND ERROR FOR TRADITIONAL TIME SERIES MODELS

Value	Jan	Feb	Mar	Apr	May	RMSE	MAPE
Actual Value	745.82	853.60	1017.89	1100.93	1216.77	-	-
GLM	986.13	988.18	990.24	992.3	994.36	166.07	15.78
HoltWinters	841.78	928.33	1000.99	1066.63	1086.62	81.48	7.42
ARIMA	799.24	859.33	969.4	1065.12	1117.63	57.18	4.80

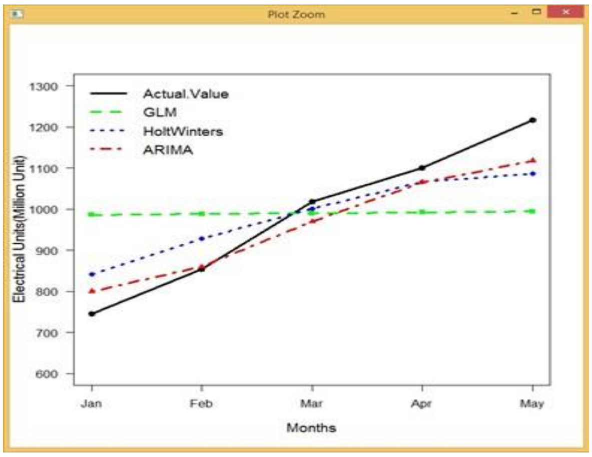
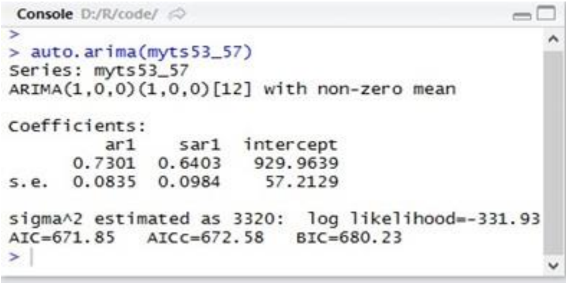


Fig. 7. Comparison graph between 5 intervals of actual value and predicted value



```
>
> auto.arima(myts53_57)
Series: myts53_57
ARIMA(1,0,0)(1,0,0)[12] with non-zero mean

Coefficients:
         ar1     sar1  intercept
      0.7301   0.6403   929.9639
s.e.  0.0835   0.0984    57.2129

sigma^2 estimated as 3320:  log likelihood=-331.93
AIC=671.85   AICc=672.58   BIC=680.23
> |
```

Fig. 8. Function auto.arima() for defining format of ARIMA model

From table 1, the lowest RMSE and MAPE are obtained from ARIMA model. RMSE is 57.18 and MAPE is 4.80. Figure 7 presents a linear plot of actual against predicted values that are obtained from all 3 models. A graph of ARIMA shows the smallest difference between the actual and predicted values. This result is obtained from using auto.arima() for defining ARIMA model as shown in figure 8.

The forecasting ARIMA model consists of autoregressive(AR) in which p = 1 and seasonal autoregressive(SAR) with P = 1. When estimating the coefficient of AR$_1$ ($\hat{\theta}_1$) and SAR$_1$ ($\hat{\Theta}_1$), we get the values of 0.7301 and 0.6403, respectively, and the parameters $d = D = 0$ and $q = Q = 0$. The forecasting equations are presented as follows:

$$(1-\hat{\theta}_1 B)(1-\hat{\Theta}_1 B^{12})Y_t = \theta_0, \qquad \text{while } \theta_0 \text{ is constant}$$

$$(1-\hat{\theta}_1 B-\hat{\Theta}_1 B^{12}+\hat{\theta}_1 B\hat{\Theta}_1 B^{12})Y_t = \theta_0$$

$$Y_t-\hat{\theta}_1 Y_{t-1}-\hat{\Theta}_1 Y_{t-12}+\hat{\theta}_1\hat{\Theta}_1 Y_{t-13} = \theta_0$$

$$Y_t = \theta_0+\hat{\theta}_1 Y_{t-1}+\hat{\Theta}_1 Y_{t-12}-\hat{\theta}_1\hat{\Theta}_1 Y_{t-13}$$

$$Y_t = \theta_0+0.73Y_{t-1}+0.64Y_{t-12}-0.47Y_{t-13}$$

From existing observed values of time series, θ_0 = 92.42 and the forecasting equation of ARIMA can be present as equation 12.

$$Y_t = 92.42+0.73Y_{t-1}+0.64Y_{t-12}-0.47Y_{t-13} \qquad (12)$$

B. *Results of Artificial Neural Networks Model*

We define 4 different ANN models: 3L-4N (one neuron in hidden layer), 3L-5N (two neurons in hidden layer), 3L-6N (three neurons in hidden layer), and 3L-7N (four neurons in hidden layer).

From our experiment, we discover that we should not set the goal an error parameter (net.trainParam.goal) too low because it may cause over-fitting. Thus, we set goal error equal to 0.0001. In addition, when we set the initial weight and bias of each model appropriately, RMSE and MAPE of each model are decreased. As a result, the lowest RMSE and MAPE of each model have been observed after experimentally set the most appropriate parameter values. These appropriate initial weights and bias of each model can be summarized in the table 2.

While using initial weight and bias of each model in table 2 to perform model training, then predict next 5 interval of power distribution unit from January 2015 to May 2015. After that, calculate RMSE and MAPE comparing with actual value. We summarize the result in table 3.

From table 3, we can see that 3L-6N model resulted in the lowest MAPE and table 4 shows that 3L-6N has the highest correlation coefficient. That means the 3L-6N model is highly related to the target. Figure 9 presents a regression graph between the output and the target of a 3L-6N model. When we increase 1 number of neuron, from 3L-6N to 3L-7N, the RMSE and MAPE increase. Therefore, the 3L-6N model is the most appropriate model for predicting this time series. When we use command set in Matlab to show the weight values of each layer, we can define the structure of ANNs as shown in figure 10.

TABLE II
INITIAL WEIGHTS AND BIAS OF EACH MODEL

Model	net.iw{1,1}	net.lw{2,1}	net.b{1}	net.b{2}
3L-4N	[0.5 0.5]	[0.5]	[0.5]'	[0.5]'
3L-5N	[0.5 0.5;0.1 0.1]	[0.4 0.4]	[0.5 0.5]'	[0.5]'
3L-6N	[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]'
3L-7N	[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]'

TABLE III
ACTUAL VERSUS PREDICTED VALUES OF ANNS MODELS

	Jan	Feb	Mar	Apr	May	RMSE	MAPE
Actual Value	745.82	853.60	1017.89	1100.93	1216.77	-	-
3L-4N	786.94	789.21	911.85	1096.59	1190.53	59.67	5.21
3L-5N	785.26	809.44	905.99	1123.58	1193.65	58.47	5.08
3L-6N	753.33	816.63	890.15	1148.87	1207.41	63.46	4.58
3L-7N	759.61	760.58	913.55	1089.92	1184.74	64.66	5.33

TABLE IV
CORRELATION COEFFICIENT AND EQUATION OF ANNS

Model	Correlation coefficient (R)	Relative equation of output and target
3L-4N	0.84323	Output=0.71*Target+0.16
3L-5N	0.87184	Output=0.76*Target+0.14
3L-6N	0.88868	Output=0.79*Target+0.12
3L-7N	0.84950	Output=0.72*Target+0.16

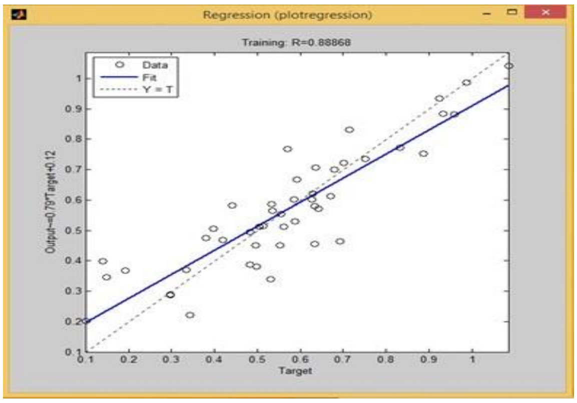


Fig. 9. Regression graph of 3L-6N model

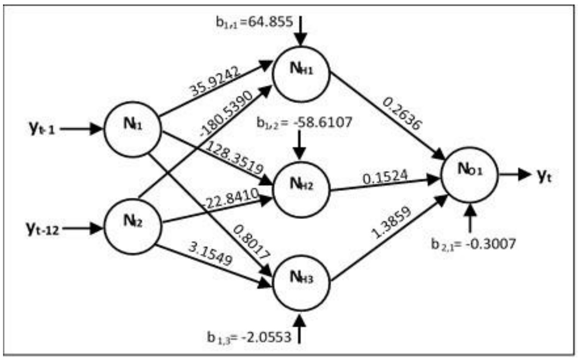


Fig. 10. Artificial neural networks for 3L-6N model

C. Comparison Between ARIMA and 3L-6N ANN Models

From the literature review [5],[6], the authors suggest that ARIMA is the accurate model to forecast data with moving by trend and season. We thus compare ARIMA with tripple exponential smoothing technique using HoltWinters model and simple linear regression using GLM model. From our experiment, we discover that the ARIMA model gives the lowest error (RMSE = 57.18 and MAPE = 4.80).

To perform time series analysis with ANNs technique, we have found that the 3L-6N model gives the lowest MAPE (=4.58) and also the highest correlation coefficient. That means the output of a 3L-6N model relates the most to the target.

When plot 5 intervals from January 2015 to May 2015 of actual observation values compared to data predicted by ARIMA and 3L-6N models (table 5 and figure 11), it can be concluded that ARIMA's graph is more similar to the actual observed values than 3L-6N's graph but a 3L-6N model can adjust irregular error in March to normal error rapidly. As a result, MAPE of 3L-6N is lower than ARIMA but RMSE is higher than ARIMA.

TABLE V
ACTUAL VS PREDICTED VALUES OF ARIMA AND 3L-6N ANN MODELS

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
3L-6N	753.33	816.63	890.15	1148.87	1207.41	63.46	4.58

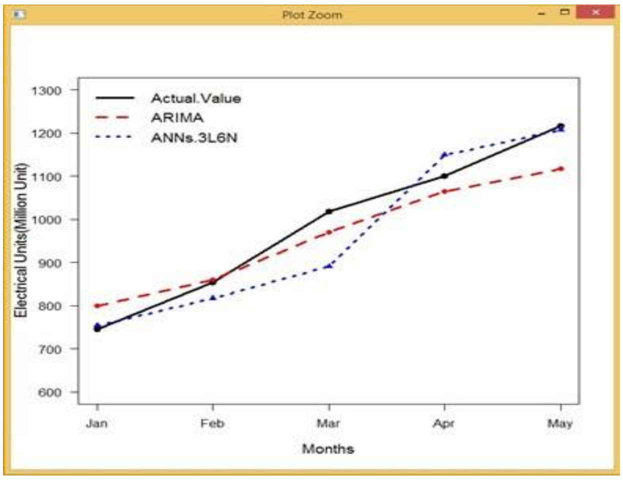


Fig. 11. Comparison graph between ARIMA and ANNs 3L-6N

V. CONCLUSIONS

We study time series analysis for forecasting power distribution unit using monthly data reported by the Metropolitan Electricity Authority from January 2010 to May 2015. We use data from January 2010 to December 2014 as a training set and data form January 2015 to May 2015 as the test set. In our experiment for model comparison, we select 3 traditional time series models (GLM, HoltWinters, ARIMA) and 4 ANNs models (3L-4N, 3L-5N, 3L-6N, 3L-7N).

In traditional time series analysis, ARIMA model is the most accurate model (RMSE = 57.18 and MAPE = 4.80). In ANNs technique, we discover that 3L-6N model (three-layer feed-forward back propagation neural network) is the most appropriate model (MAPE = 4.58, correlation coefficient = 0.88868). When compare ARIMA against the 3L-6N model, it turns out that ARIMA's graph is more similar to the actual observed values than the 3L-6N's graph, but MAPE of 3L-6N is lower than ARIMA's MAPE. As a result, predicted values from 3L-6N model are more accurate than the ARIMA model.

In conclusion, the ANNs model of the 3L-6N structure can be applied to time series forecasting at high precision. The caution is that the high number of hidden layer's neurons may cause over-fitting.

REFERENCES

[1] Metropolitan Electricity Authority. (2015). Report of electrical distribution [Online]. Available : http://www.mea.or.th/download/index.php#

[2] D.W. Bunn and E.D. Farmer, *Comparative Models for Electrical Load Forecasting*. New York: John Wiley and Sons, 1985.

[3] P. M. Maçaira, R. C. Souza and F. L. C. Oliveira, "Modelling and Forecasting the Residential Electricity Consumption in Brazil with Pegels Exponential Smoothing Techniques", *Procedia Computer Science*, vol.55, pp. 328-335, 2015.

[4] I. Keka and M. Hamiti, "Load profile analyses using R language*" in Proceedings of the ITI 2013 35th Int. Conf. on International Technology Interfaces*, Cavtat, Croatia, 2013, pp. 245-250.

[5] P. Chujai, N. Kerdprasop and K. Kerdprasop, "Time series analysis of household electric consumption with ARIMA and ARMA models" *in Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, China, 2013, pp. 295-300.

[6] X. Wang and M. Meng, "A Hybrid Neural Network and ARIMA Model for Energy Consumption Forecasting" *JOURNAL OF COMPUTERS*, vol.7, No.5, pp. 1184-1190, 2012.

[7] M. Firata, M.E. Turanb and M.A. Yurdusev, "Comparative analysis of neural network techniques for predicting water consumption time series" *Journal of Hydrology*, vol.384, pp. 46-51, 2010.

[8] I. Boldina and P.G. Beninger, "Strengthening statistical usage in marine ecology: Linear regression" *Journal of Experimental Marine Biology and Ecology*, vol. 474, pp. 81-91, 2016.

[9] V. Prema, and K.U. Rao, "Development of statistical time series models for solar power prediction" *Renewable Energy*, vol. 83, pp. 100-109, 2015.

[10] Y. Wang, J. Wang, G. Zhao and Y. Dong, "Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China" *Energy Policy*, vol.48, pp. 284-294, 2012.

[11] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation" *Information Sciences*, vol.191, pp. 192-213, 2012.