Peak Load Forecasting for Vietnam National Power System to 2030

Nguyen Hoang Minh Vu[1], Vo Viet Cuong[1*], Phan Thi Thanh Binh[2]
[1] HCMC University of Technology and Education, No. 1 Vo Van Ngan Street, HCMC, Vietnam
[2] HCMC University of Technology, No. 268 Ly Thuong Kiet Street, District 10, HCMC, Vietnam
Received: August 10, 2017; Accepted: November 03, 2017

Abstract

Gross domestic product (GDP) growth rate, electric power consumption, and maximum load power demand (P_{max}) have a closed but complicated and unexplicit correlation. Using the feed-forward back propagation (FFBP) method, a modified model of neural network, this paper will introduce a new long-term prediction approach for the maximum load power of Vietnam. Results from simulation indicate a considerable correlation of three parameters regarding electric power consumption demand, GDP growth rate, and maximum load power demand; the mean error of suggested model is about 1.92%. This is a reasonable range of mean error for a long-term prediction where the correlation of variables is not explicit. According to the basic scenario of National Economic Forecasting Model to 2030, the Vietnam's GDP annual growth rate is about 7% per year, and the corresponding electric power demands (GWh) are forceated in the previous paper, P_{max} in 2020, 2025 and 2030 are forecasted here at 40,332 MW, 60,835 MW, and 87,558 MW, respectively. Those results are 3.4 - 4.2% lower than forecasted values of the Revised National Power Development Plan VII (hereinafter referred to as PDP 7 rev) for the period of 2011-2020 with the vision to 2030.

Keywords: Forecast, Peak load, Neural network, Feed-forward back propagation - FFBP, Vietnam.

1. Introduction

P_{max} (also called as peak load) is referred to the maximum power demand in a specific prediction time duration. In general, to forecast a P_{max} value requires a similar principle and high accuracy-requirements to load power demand prediction. However, forecasting load power demand requires a long duration of prediction (i.e. a week, a month, or a year…) to meet the load regulation plans of National Electricity Authority, while P_{max} forecasting only needs to identify the maximum point of load consumption demand as a significant input of both regulation plans and electricity reserve margin calculations.

P_{max} forecasting in particular, or power load prediction in general is one of the most concerns of power planning due to its direct impacts on generation, regulation, reserve margin, and energy security plannings. Thus, forecasting approaches for power load demand have been launched into many textbooks of speciality. Besides, numerous studies have been carried out for researching on different forecasting techniques where the national economic context is concerned as compulsory factor.

Additionally, the Electricity Regulatory and Authority of Vietnam (ERAV) has released the Decision No. 07/QĐ-ĐTĐL in which stipulates the process of forecasting power load demand in the national electricity system [1]. Also, the Decision consists all related principles, procedures, document and methods of forecasting electricity power demand which could be employed as establishment-basis of the National Electricity System Investment and Development Plan.

Most of those methods, however, have not been launched to forecast the peak value of a load curve diagram (P_{max}) yet. Other techniques, as mentioned above, have similar principles and high accuracy-requirements with power load demand prediction methods. Some common techniques are listed as below:

SARIMAt model: a time series is defined as a set of variable values. This model is a multi-step simulation method in which commonly consists of determination, estimation, assessment, and prediction phases.

Regressive model: to establish a statistical-based data system, it is important to consider all variables which could impact on load power demand, i.e. temperature, holiday, weather, special seasonal day, etc. The characteristics of those data has led to the use of the time-series regressive model with autoregressive errors. This method is remarkable for short-term load forecasting [2].

[*] Corresponding author: Tel.: (+84) 986.523.475
Email: cuongvv@hcmute.edu.vn

Fuzzy logic rules: this approach has been widely implemented in automatic control, data classification, sample identification, making analysis and decision, expert system, prediction and so on. The development of fuzzy logic rules is based on results obtained from independent analysis related to load demand of some experts [2]. In order to launch fuzzy logic rules, it is required to meet all below conditions: (i) set-up a weather-based model and (ii) end-use behavior model (based on practical observation in different observing duration, i.e. a day, a week, a month, etc.), and (iii) historical load data (i.e. diurnal data, weekly data, or monthly data, etc.).

Artificial neural networks (ANNs): ANNs are widely used in electrical forecasting due to its high possibility of treatment hard algorithms and complicated correlation. ANNs has achieved successes in solving a wide range of issues, such as: (i) planning, monitoring, analysing, securing, designing, and forecasting load demand; (ii) energy analysis; and (iii) error dianosing; of which forecasting load demand, energy analysis, and error dianosing are the most common works of this method. ANNs techniques could be categorised into 3 forms:

- *Recurrent neural network (RNN)*: this network contains at least one feed-back data connection, in which allows events to be coded as time series data. The feed-back connection could be occured internally or externally. Also, RNN could employ historical models and data as inheritance, generalise and forecast future load profiles [4].

- *Feed-forward back propagation (FFBP)*: FFBP is one of the widest-used modified neural networks. It could be implemented to any issue which seriously require to map a model. In terms of function, it could generate an output related model from a specific input model. Moreover, FFBP could learn and upgrade its computing process based on a simple relevant idea. By continuously generate necessary corrections, FFBP could release both learning and corresponding models simultaneously which adapt to each system's input. If the result is identified as wrong value, then the weights of the network will be replaced and recomputed. Consequently, future results shall be more accurate.

- *Radial basis function network (RBFN)*: in terms of model recognition, an RBFN could be represented as perceptron network architecture. Any non-linear system could be estimated to be an RBF system approximately. This is the key which makes the RBF be appropriated with the model identification problems. Theoretically, an RBF is a 3 layer-neural network (with 1 hidden layer). However, the output of network always transforms linearisingly corresponding to the connective weights.

All above models could be categorised into 2 groups: (1) traditional group regarding ARIMA and regressive models; and (2) artificial neural group relevant to fuzzy logic and neural network.

Traditional methods could be combined with multi-model forecasting techniques to create hybrid forms. Various results attained from different hybrid simulations are listed in [3]. Nonetheless, these mentioned approaches could not perform correctly the complicated non-linear correlation between the power load and factors which could impact on it. Furthermore, most traditional methods, i.e. autoregressive model, seem to be more appropriated with short-term predictions only. It means that implementing traditional approaches for long-term forecasting could generate unexpected errors.

This paper aims to calculate the P_{max} value for Vietnam national electricity system to 2030 based on historical data of power consumption and GDP growth rate. Due to the fact that the correlation of input parameters is unexplicit, it is suggested to employ ANNs to solve that prolem of correlation.

FFBP technique will be implemented due to its possibilities of self-learning and auto-modified the weights of network. For this reason, the output results are expected to be more accurate.

2. Methodology

The correlation of electric load and related traditional factors, such as: GDP, socio-economic factors (i.e. power consumption per capita, power consumption per product, electric tariff, etc.), are strongly impacted by temporal factors (i.e. reducing factor of technology cost, high rate of electrification, etc.). As temporal factors are extremely difficult to be quantified precisely, the mentioned correlation becomes to be unexplicit. In order to solve an unexplicit and complicated algorithm, ANNs is considered as the most effective method and common implementation. This method is employed to compute the correlation by approximating nonlinear functions.

A neural network (NN) is commonly trained by a supervising-based algorithm like back-propagation. This algorithm is provided with historical and related data to modify the weights and the thresholds of network so that errors of prediction could be minimised in training set. If the training algorithm is logical and precise, then the learning result has performed a fairly unknown function by reasonably simulating the correlation of input and output data. In other words, the training loop has created a corresponding response between the input and output data. Therefore, by adding an updated signal as input, a corresponding forecasted signal will be trained and attained as output. The operational principle of a NN is illustrated as figure 1. An input data X will be trained by NN to become an output data Y and their correlation is performed as a mathematical logic. Each artificial neuron (node) will connect and receives signals x_i with corresponding weights w_i from other nodes, then the total weight of input data (hereinafter referred as *net input weighted sum*) is defined as:

$$a = \sum_{i=1}^{n} w_i x_i \tag{1}$$

Where: a is the linear component of neurons (also called as the net input weighted sum of neurons); x_i are the net input data; w_i are the corresponding weights of each input data; and n is the number of neuron input.

The activation function f acts as a "state switch" of a neuron. It transforms a net input signal a to the net output signal z, where $z = f(a)$.

Where: f is the activation function of a neuron and z is the output nonlinear signal of a neuron.

2.1 FFBP algorithm

Figure 2 illustrates a simulation model of direct connection neural network; of which x_1, x_2, and x_3 are defined as input neural data; z is hidden layer; y is the output layer (result); v is the net input weighted sum; and w is the net output weighted sum. Calculation formulas of z and y are defined as below:

2.1.1 Hidden layer (z) [5]

$$net_q = v_q^T x = \sum_{j=1}^{m} v_{qj} x_j \tag{2}$$

$$z_q = a(net_q) = a(v_q^T x) = a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \tag{3}$$

2.1.2 Output layer (y) [5]

$$net_i = w_i^T z = \sum_{q=1}^{r} w_{iq} z_q = \sum_{q=1}^{r} w_{iq} a\left(\sum_{j=1}^{m} v_{qj} x_j\right) \tag{4}$$

$$y_i = a(net_i) = a\left(\sum_{q=1}^{r} w_{iq} z_q\right) = a\left(\sum_{q=1}^{r} w_{iq} a\left(\sum_{j=1}^{m} v_{qj} x_j\right)\right) \tag{5}$$

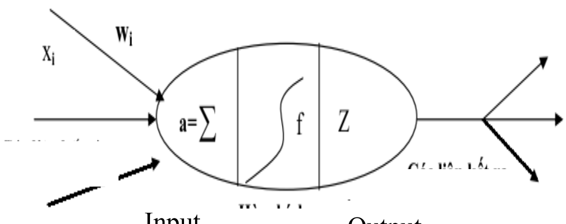

Fig. 1. Operational principle of a neural network [5].

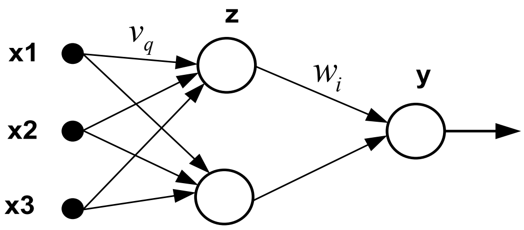

Fig. 2. Direct connection NN architecture [5].

2.1.3 Objective function

The total squared deviation between the objective d and the output of neural network y (also referred as *the total error*) must be minimised. The calculation can be expressed as:

$$\begin{aligned} J(w) &= \frac{1}{2}\sum_{i=1}^{p}(d_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{p}(d_i - a(net_i))^2 \\ &= \frac{1}{2}\sum_{i=1}^{p}\left(d_i - a\left(\sum_{q=1}^{r} w_{iq} z_q\right)\right)^2 \rightarrow \min \end{aligned} \tag{6}$$

The mentioned objective function has been considered as a classic function to express the learning ability of NN. Recursive algorithm rules and Generalised Delta learning rules are herein employed to be reformed using the steepest-descent method.

2.1.4 Updating weight vectors

Weight vectors of a NN shall be updated continuously to revise the learning process of network and to compute the final weights afterward. In other words, this is the final iterative round in which the minimum value of objective function is released and a full set of network weights is finalised.

The connection between the hidden-layer and output-layer can be described as [5]:

$$\begin{aligned} \Delta w_{iq} &= -\eta.\frac{\partial J}{\partial w_{iq}} = -\eta.\frac{\partial J}{\partial y_i}.\frac{\partial y_i}{\partial net_i}.\frac{\partial net_i}{\partial w_{iq}} \\ &= \eta.(d_i - y_i).\frac{\partial a(net_i)}{\partial net_i}.z_q = \eta.\delta_{oi}.z_q \end{aligned} \tag{7}$$

Where:

$$\delta_{oi} = -\frac{\partial J}{\partial y_i}.\frac{\partial y_i}{\partial net_i} = (d_i - y_i)\frac{\partial a(net_i)}{\partial net_i} \tag{8}$$

And η is learning factor.

The connection between the input-layer and hidden-layer can be written as [5]:

$$\Delta v_{qj} = -\eta.\frac{\partial J}{\partial v_{qj}} = -\eta\sum_{i=1}^{p}\frac{\partial J}{\partial y_i}.\frac{\partial y_i}{\partial net_i}.\frac{\partial net_i}{\partial z_q}.\frac{\partial z_q}{\partial net_q}.\frac{\partial net_q}{\partial v_{qj}} \tag{9}$$

$$\Delta v_{qj} = \eta \sum_{i=1}^{p} \left(d_i - y_i\right) \frac{\partial a\left(net_i\right)}{\partial net_i} w_{iq} \frac{\partial a\left(net_q\right)}{\partial net_q} .x_j = \eta \delta_{hq} x_j \quad (10)$$

$$\begin{aligned} \delta_{hq} &= -\frac{\partial J}{\partial net_q} = -\frac{\partial J}{\partial z_q} . \frac{\partial z_q}{\partial net_q} \\ &= \frac{\partial a\left(net_q\right)}{\partial net_q} . \sum_{i=1}^{p} \left(d_i - y_i\right) . \frac{\partial a\left(net_i\right)}{\partial net_i} w_{iq} \end{aligned} \quad (11)$$

When the binary sigmoid function is employed for activation function $a(.)$, function (12) below is used.

$$\delta_{hq} = \frac{\partial a\left(net_q\right)}{\partial net_q} . \sum_{i=1}^{p} \delta_{oi} . w_{iq} \quad (12)$$

Then equation (7) and (10) are rewritten as:

$$\delta_{oi} = \frac{1}{2}\left(1 - y_i^2\right)\left(d_i - y_i\right) \quad (13)$$

$$\delta_{hq} = \frac{1}{2}\left(1 - z_q^2\right) . \sum_{i=1}^{p} \delta_{oi} . w_{iq} \quad (14)$$

$$\begin{aligned} \Delta w_{iq} &= \frac{1}{2} \eta . \left(1 - y_i^2\right) . \left(d_i - y_i\right) . z_q . \Delta v_{qj} \\ &= \frac{1}{2} \eta . \left(1 - z_q^2\right) . x_j . \sum_{i=1}^{p} \delta_{oi} . w_{iq} \end{aligned} \quad (15)$$

Where:

$$\frac{\partial a\left(net\right)}{\partial net} = \frac{1}{2}\left[1 - a^2\left(net\right)\right] \quad (16)$$

2.1.5 Modifying weight vectors

According to figure 6, as connecting weights were chosen randomly initially, they will be modified to response the error value e of network afterward.

$$w_{iq}^{*} = w_{iq} + \Delta w_{iq} \quad (17)$$

$$v_{qj}^{*} = v_{qj} + \Delta v_{qj} \quad (18)$$

2.2 Common activation function

Activation functions commonly used in direct connection NN are listed as: logsig(n) (see fig.3); tansig(n) (see fig.4); and purelin(n) (see fig.5).

Each function has its own specific accuracy and reliability. Thus, choosing an appropriate function to be activated for a network requires a scrupulous consideration to the accuracy of output signal.

2.3 Supervised-learning rules

When a set of input data is given as $\{x_1, d_1\}$, $\{x_2, d_2\}$, …, $\{x_q, d_q\}$, then initial weights of network are chosen randomly. When an input data x_q is launched into the network, the output data of network y_q will be compared to objective output d_q. The supervised-learning rules are based on the iterative computation of network error value $e_q = d_q - y_q$. Network weights and threshold shall be adjusted and updated to shorten the difference between output data and objective output value. Figure 6 illustrates a reinforced supervised-learning diagram of FFBP.

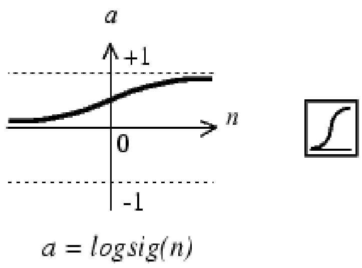

Fig. 3. The activation function a = logsig(n).

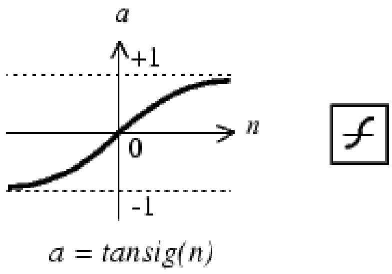

Fig. 4. The activation function a = tansig(n).

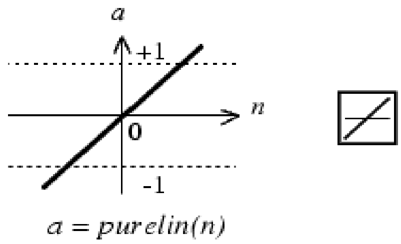

Fig. 5. The activation function a = purelin(n).

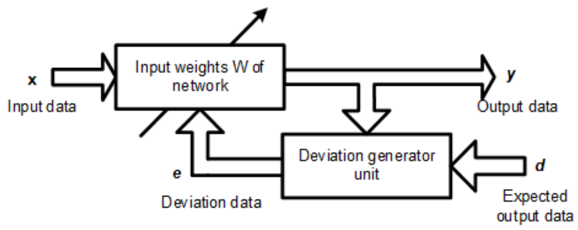

Fig. 6. A reinforced diagram of FFBP.

2.4 Important concerns in training process

If the final result of training process could not meet the expected output result, it is compulsory to restart the training process. As the weight vectors and activation thresholds of network will be regenerated randomly once the training process restarted, the result will be changed significantly. It is advised to do many experiments for obtaining the optimal set of weight vectors and thresholds.

In cases of no set of result is found, then the number of neuron in hidden-layer should be increased to improve the accuracy of training process and speed of learning algorithm. Similarly, the number of hidden layer could be increased. However, too many hidden layers or too many neurons in a layer will lead to the reduction of training accuracy due to the increasing number of computed parameters and high demand of system memory. As a result, the expected correlation between input and output data could not be found. In terms of learning ability, there were many studies on the impact of increasing number of hidden layer and neuron on network learning ability and speed. However, there still has no convinced answer for this issue.

In some cases, increasing the number of hidden layer and neuron on network could not lead to a reasonable result. It is seriously to review and/or replace the training function. Each training function has its own computing algorithm to determine the convergence rate of data set. Thus, considering the requirements of problem and choosing an appropriate training function are extremely important in early stage of work.

3. Implementing FFBP algorithm to forecast the P_{max} value of Vietnam national electric system

3.1 Choosing input variables

GDP growth rate (%/yr) and electric power demand (GWh) are identified as input variables of forecasting simulation.

Table 1 indicates historical data according to the PDP 7 rev [6], including: (i) X_1 is the historical data of GDP growth rate (%/yr); (ii) X_2 is the historical data of yearly electric power demand (GWh); and *Y_target* is defined as the historical data of P_{max} (MW). Those data will be imported into the network training algorithm afterward.

In order to check the accuracy of training process, historical data is categorised into 2 groups: input data set and test data set. The input data set is commonly bigger than the test-set; and it often accounts for 70%-90% of total data. The test data set, meaningly, is used to assess the accuracy of training process after completed. There is two ways to classify a test data set: (1) one is choosing randomly some samples of initial input data, and (2) the other is collecting some nearest data as its accuracy is normally much higher than further data [4].

Table 1. Historical data using for network training

Year	X_1 (%/yr)	X_2 (GWh)	Y_target Pmax (MW)
A. Input data set			
1990	5.10	8,678	1,660
1991	6.00	9,152	1,850
1992	8.60	9,654	2,005
1993	8.10	10,665	2,143
1994	9.30	12,284	2,408
1995	9.54	14,636	2,796
1996	9.34	16,946	3,177
1997	8.15	19,151	3,595
1998	5.80	21,665	3,875
1999	4.80	23,739	4,329
2000	6.80	26,745	4,615
2001	6.20	30,187	5,655
2002	6.30	34,073	6,552
2003	6.90	38,461	7,408
2004	7.50	43,414	8,283
2005	7.55	49,008	9,255
2006	6.98	53,845	10,187
2007	7.13	59,159	11,286
2008	5.66	64,998	12,636
2009	5.40	71,415	13,867
2010	6.42	78,466	15,416
B. Test data set			
2011	6.24	94,658	16,490
2012	5.25	105,474	18,603
2013	5.42	115,069	20,010
2014	5.98	128,435	22,210
2015	6.20	141,800	25,295

According to the table 1, 21 samples of data (from 1990 to 2010) will be imported to the neural network to be trained. When the training process is completed, then using the 5 nearest value of P_{max} (from 2011 to 2015) to test the forecasting value of P_{max} which will be delivered by the network.

GDP growth rate and electric load power demand are forecasted in the base-scenario as indicated in table 2 [6], [7].

Table 2. Base-scenario of Vietnam power system

Year	GDP growth rate [%/yr] [6]	Electric load power demand [GWh] [7]
2020	7	230,195
2025	7	349,949
2030	7	511,268

3.2 Constructing the network training process

Employing the Levenberg – Marquardt back-propagation (Trainlm) to be the training algorithm of this simulation. The Trainlm provides a training process with 2 inputs (GDP growth rate and electric load power demand, correspondingly), and 1 output (corresponding to P_{max} value). The recommended architecture of network consists of 1 hidden layer with 10 neurons and 1 output layer with 1 neuron. A purline activation function is suggested to be applied for that both layers. The logsig and tansig functions are denied to be implemented in this case as the output value could become saturated if the input value is higher than the network training threshold. The purline linear function, on the contrary, is appropriate to most extrapolate problems. (see fig. 7). Input data is set and shown in table 3.

3.3 Testing error of training process

Y_target is defined as the historical data of P_{max} (MW), while *Y_training* is computed result by NN. By practical experiment, it has an evidence that increasing the number of hidden layer and neuron in a layer will make the speed of learning process reduced and the result has some trivial changes.

Then the *Y_training* values are compared to the *Y_target* data. The average change between that both values (also referred as *the mean error of training process*) is less than 1.65% (see table 4). The highest error is remarked at 3.1% in 1993.

Table 3. Setting of input data

Parameter	Value	Description
Epochs	1000	The maximum iteration of training process
Goal	0	Error objective
max_fail	1000	The maximum time for error detection
min_grad	0.0000001	The minimum processing gradient
mu	0.001	Beginning *mu* value
mu_dec	0.1	The *mu* decline coefficient
mu_inc	10	The *mu* increase coefficient
mu_max	1E+10	The maximum value of *mu*
Show	25	The number of epochs indicated
Show Command Line	FALSE	Create output command
Show Window	TRUE	Show training window
Time	Inf	Duration of training

Table 4. Input data and trained results

Year	Y_target	Y_training	Error%
1990	1,660	1621.3	2.3
1991	1,850	1836.2	0.7
1992	2,005	1954.9	2.5
1993	2,143	2076.2	3.1
1994	2,408	2403.8	0.2
1995	2,796	2720.8	2.7
1996	3,177	3081.9	3.0
1997	3,595	3530.1	1.8
1998	3,875	3792.8	2.1
1999	4,329	4224.5	2.4
2000	4,615	4735.2	2.6
2001	5,655	5543.1	2.0
2002	6,552	6429.5	1.9
2003	7,408	7204.6	2.7
2004	8,283	8179.5	1.2
2005	9,255	9083.2	1.9
2006	10,187	9988.5	1.9
2007	11,286	11432.3	1.3
2008	12,636	12412.8	1.8
2009	13,867	13537.2	2.4
2010	15,416	15118.0	1.9

Table 5. Test results of training

Năm	Y_target	Y_training	Error%
2011	16490	16943.3	2.7
2012	18603	18819.5	1.2
2013	20010	20520.2	2.5
2014	22210	22900.5	3.1
2015	25295	25268.9	0.1

Table 6. Forecasted results P_{max} value

Year	GDP growth rate (%/yr)	Electric load power demand (GWh)	Forecasted P_{max} value (MW)
2020	7	230,195	40,332
2025	7	349,949	60,835
2030	7	511,268	87,558

4. Forecasted values of P_{max}

When applying the test value set to assess the accuracy of FFBP model, then the comparison results are shown in table 5. The average error of the model is remarkable at 1.92%, approximately. This is a reasonable and acceptable result of a long-term forecasting model. For this reason, the model continues being used to forecast the future electric power demand (GWh) and corresponding P_{max} values. Forecasting conditions are kept as mentioned in table 2. Results of forecasting are shown in table 6.

5. Conclusion

In this paper, a FFBP method is suggested to be implemented as a new forecasting model for the maximum load power demand (P_{max}) of Vietnam national power system. This approach is strongly based on the correlation between the electric power consumption and GDP growth rate. The forecasted error is compared to the practical data. Some positive results are highlighted as:

(1) There is a considerable correlation between the GDP growth rate, demand of electric power consumption, and the maximum load power demand;

(2) The mean error of suggested method is less than 1.92%. This is an acceptable range of error for a long-term prediction model in which the correlation between input variables is unexplicit.

The results show that the P_{max} values of the year 2020, 2025, and 2030 are 40,332 MW, 60,835 MW, and 87,558 MW, respectively. In comparison to the base-scenario of PDP 7 rev, when integrating the two variables of GDP growth rate and electric power consumption, then forecasted values of P_{max} are 3.4%-4.2% lower than that of values of PDP 7 rev.

References

[1] Decision No. 07/QĐ-ĐTĐL on the process of forecasting power load demand in the national electricity system. Hanoi, 14/03/2013.

[2] Z. Ismail, A. Yahya, K. A. Mahpol, Forecasting Peak Load Electricity Demand Using Statistics and Rule Based Approach, American Journal of Applied Sciences. 6(8) (2009) 1618-1625.

[3] L. Ghods, M. Kalantar, Different Methods of Long-term Electric Load Demand Forecasting; A Comprehensive Review, Iranian Journal of Electrical and Electronic Engineering. Vol 7(4) (2011).

[4] B. S. Kermanshahi, H. Iwamiya, Up to year 2020 Load Forecasting Using Neural Nets, Electric Power System Research. Vol 24(9) (2002) 789-797.

[5] D. Howard, B. Mark, The Mathworks: User Guide Neural Network Toolbox For Use with MATLAB.

[6] Decision No. 428/QĐ-TTg on The Revised National Power Development Plan VII (PDP 7 rev). 18/03/2016.

[7] N. H. M. Vu, N. T. P. Khanh, V. V. Cuong, P. T. T. Binh, Forecast on Vietnam Electricity Consumption to 2030, the 2017 International Conference on Electrical Engineering and Informatics (ICELTICs 2017). IEEE (2017) 81-86.