Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

Vol. 8, No. 1, Juli 2015

ISSN 0216 – 0544
e-ISSN 2301– 6914

CRITICAL TRAJECTORY - EXTREME LEARNING MACHINE TECHNIQUE FOR COMPUTING CRITICAL CLEARING TIME

[a]Irrine Budi Sulistiawati,[b]Ardyono Priyadi, [c]Adi Soepriyanto
[a,b]Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember ITS, Surabaya Indonesia
[c]Department of Electrical Engineering, Institut Teknologi Nasional Malang Indonesia
E-mail: irrine10@mhs.ee.its.ac.id

Abstract

Electric power system is called reliable if the system is able to provide power supply without interrupted. However, in large systems changing on the system or disturbance may affect the power supply. Critical clearing time is the time for deciding the system is a stable or an unstable condition. Critical clearing time has also relationship with setting relay protection to keep the system in the stable condition. Prediction of critical real time for online assessment is expected to be used for preventive action system. That's why critical clearing time still an interesting topic to be investigated.This paper calculating time of Extreme Learning Machine to predict critical clearing tim on system. Before predicted by Extreme Learning Machine, critical clearing time calculated using numerical calculation critical trajectory method with load changing and different fault occuring. Tested by Java-Bali 500 kv 54 machine 25 bus give result that Extreme learning machine is able to perform faster prediction of neural network.

*Keywords***:** *Critical Clearing Time, Neural Network, Extreme Learning Machine.*

INTRODUCTION

The Analysis development of electric power system is growing rapidly by entering the use of artificial intelligent in it. The use of conventional method is being abandoned because this kind of method takes a lot of time in the computation process, especially on transient stability analysis with its complicated non linear models, as well as the more complex problem that demand fast and accurate assessment results which is use for controlling system can be easily solved by artificial intelligent. The ability of artificial intelligent in terms of fast analysis, estimate even prediction made artificial intelligent as a main tools to execute electric power system assessment rather than another traditional method that was used previously.

The use of neural network as tools that is used in transient stability assessment becomes the attractiveness of many researches [1][2][3][4]. From several research that are used as reference [2], the use of Neural Network gives satisfactory performance results to judge system performance if we compare it with transient stability evaluation that need complicated calculation process and a lot of time to gain non linear solution.

According to [3] the use of artificial intelligent in transient stability assessment has a weakness in terms of input measurement. But, still according to [3], the solution of this problem is using neural network that has capability and knowledge in terms of learning and input processing process. Neural Network was widely used and recognized excellence to perform non linear mapping estimate from several inputs. Besides that, neural network can model artificial system as natural as possible.

However, with the development of artificial intelligent science, the use of neural network was regarded as the old method because of its learning algorithm process that is getting slower than required. The learning process that require more than couple hours, even several days, make this neural network is classified as a conventional method. Therefore, the development of neural network that rely on learning and calculating rate becomes the focus of several research recently.

According to [5][6], several research recently investigate the capability of feed forward neural network with many layer. This research concludes that continue activation function can gives better result than before. In fact, neural network perform a research that is called training by using several data that has been defined before. When perform forecasting for specific number of data, still according to [6], it appears that the

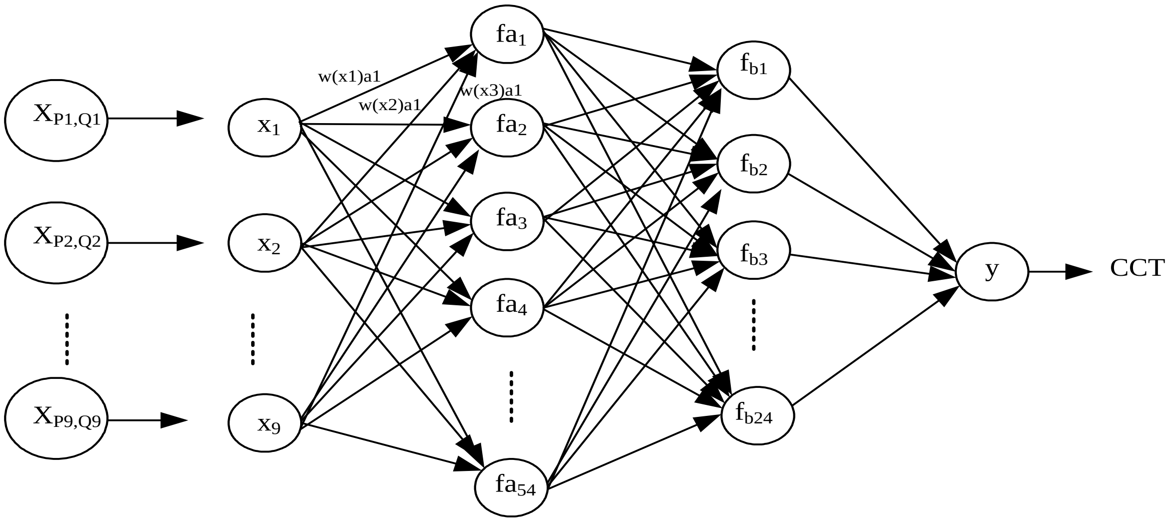

Figure 1. Neural Network Architecture

result of feed forward neural network with single layer and some specific hidden N node and the using activation function non linear indicate that the observation is fixed or unchanged at some specific hidden N node. This means that input weight which is the layer between input layer and first hidden layer and hidden bias layer need arrangement so that the result is as good as the learning algorithm result of feed forward neural network. Old algorithm of neural network indicate that the parameter of feed forward neural network has to be determined first and depend on weight and bias layer.

This research will perform analysis of time estimate of critical clearing time by using Neural Network and Extreme Learning Machine and the result from both method will be compared. To examine the effectiveness from the method that has been used, IEEE 3-machine 9-bus system and Java-Bali 500 kV 54-machine 25-bus system will be tested.

NEURAL NETWORK AND EXTREME LEARNING MACHINE

This research contains the use of artificial intelligent for calculation technique of critical clearing time. Neural Network and Extreme Learning Machine are used for performing big estimation of critical clearing time in appraisal of transient stability of electric power system.

Neural Network

Neural network was introduced for the first time at 1948 by McCulloch and Pitts, it attract the researches attention because the neural network can adopt the working process of human brain and can be used for solve the problem by model system linear function to gain desired result.

This research uses back propagation with several layers that capable to arrange weight from input to hidden layer by error way from hidden layer more than by error way from output layer. Besides that, the capability of BP Neural Network that can be used for non linear activation function and network with many parallel calculation and can model linear function make this neural network become option to solve the problem rather than another method.

According of Figure 1, The architecture of neural network, the steps of neural network stages are starts with input unit that accept input x$_i$ that is passed down to hidden layer in front of them [6]. Input unit (*x*) is through several weights (*w*) and interconnected for output (*y*). In every hidden layer, input unit will be multiplied by weight and will be summed and bias will be added to the equation

In this research, input consists of two neurons; each represents active power and reactive power of system. Hidden layer are consist of two layers, the first layer use tan-sig activation function and second layer use log-sig activation function. The weight of hidden layer can be calculated based on Equation (1).

$$Z\_in_j = w_{(x1)a1}x_1 + w_{(x2)a1}x_2 + ... + w_{(x9)a54}x_9$$

$$Z\_in_j = V_{0j} + \sum_{i=1}^{n} w_{ij} + x_i \qquad (1)$$

Every neuron weight and bias of learning process is obtained by activation function as seen in Equation (2).

$$Z_j = f(Z\_in_j) \qquad (2)$$

The activator that is used are sigmoid function that follow the Equation (3) :

$$Z_j = \frac{1}{1+\exp^{(-z\_in_j)}} \qquad (3)$$

Next step is output unit that can be achieved by multiply weight and sum the result as well as add bias at calculation process. Output layer use one neuron with purelin activation function :

$$Y_{\_in_k} = W_{0k} + \sum_{j=1}^{p} Z_j W_{jk} \qquad (4)$$

When the result of feed forward learning process was not the same with the target output, the n algorithm process of back propagation was started. In this back propagation algorithm, output that is different with target will be sent back to hidden layer toward the input layer. This process will be called as back propagation that can be seen at Figure 2 .

Learning process of back propagation neural network is supervised learning form, which is by seeing the suitability between output and target. Back propagation is started by compare output and target. If the output is not suitable with target, then the error that has been appeared will be used to improve weight so that the desired compatible output will be obtained.

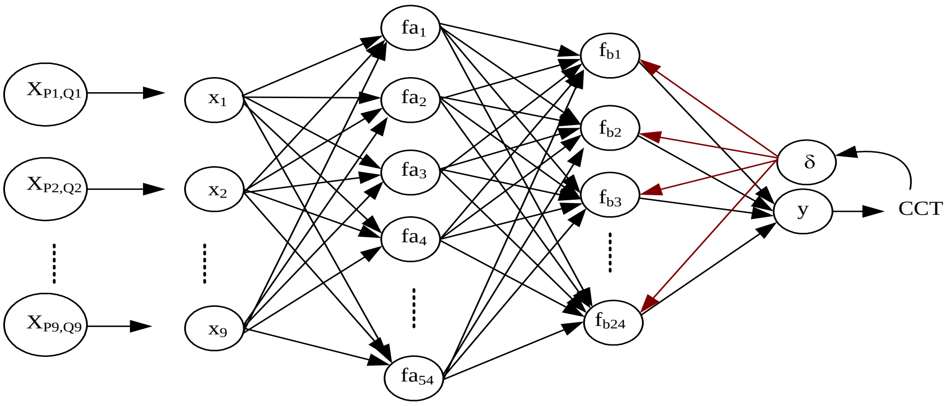

Figure 2. Architecture Back Forward

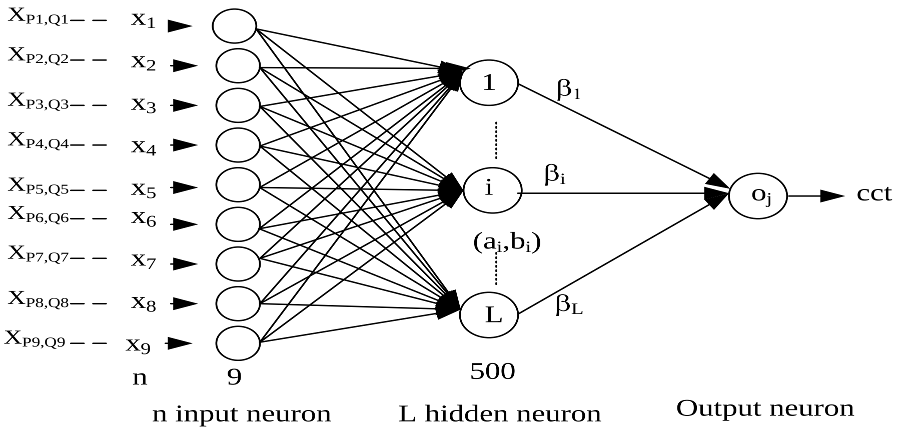

Figure 3. Architecture Extreme Learning Machine

This weight improvement process is performed by set back the unsuitable output to hidden layer to be forwarded to input layer and then fix the weight by Equation (5).

$$W_{kj}(t+1) = W_{kj}(t) + \alpha.\delta_k.Z_j \quad (5)$$

Every hidden unit (z_J) will calculate activation value and send it to outer layer. Each unit that calculate output layer activation (y_k) and compare it with target value (t_k) to determine error factor (δ_k) will be used to return output (y_k) to the next layer

Extreme Learning Machine

The use of tuning process at input weight and hidden bias make algorithm of neural network require time at the learning process [5][6]. Learning process with gradient descent at neural network algorithm that use many iteration make this algorithm of neural network require much computation time.

Calculation process that use algorithm of neural network is growing with the **discovery of** new algorithm which is Extreme Learning Machine. First discovered by Guang-Bin Huang, this method can choose input weight and bias at hidden layer randomly [5][6]. Therefore, this method does not require much time to calculation process like algorithm of neural network. Besides that, this method can achieve small training error and weight and capable to give a good and fast generalization performance. The architecture of extreme learning machine can be described as Figure 3.

The architecture of extreme learning machine as seen in the figure can be explained as:

a_i = the vector of input weight that connect hidden node to *i* and input node or center from hidden node to *i*.

b_i = threshold from hidden node to *i*

β_i = the weight vector that connect hidden node with output node.

Normalization

Initial normalization was performed at extreme learning machine to make activation function to produce output between [0,1] or [1,1]. In accordance with reference [6], it was formulated as :

$$X_n = 2x\left(X_p - \min\left\{X_p\right\}\right)/\left(\max\left\{X_p\right\} - \min\left\{X_p\right\}\right) - 1 \quad (6)$$

With :

X_n = the value of normalization result that goes between [-1,1]
X_p = the value of real data that is not normalized yet.
min {X_p} = minimum value at data set.

Mathematically, Extreme learning machine can be translated as follows. Refer to N sample that can be expressed as (x_i,t_i) with :

$$x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in R^n \quad (7)$$

$$t_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in R^m \quad (8)$$

Determine activation function g and number of node at hidden layer L. For N hidden layer and activation function in $g(x)$ then :

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i.x_j + b_i) = o_j \quad (9)$$

$$, \text{ for } j = 1, ..., N$$

With

$w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ is a weight vector that connect i hidden node and input node.
$\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is a weight vector that connect i hidden node and output node
$w_i x_j$: multiply from weight vector and input
b_i : threshold from hidden node to i

From standard SLFN with $\tilde{N}$ hidden node with activation function $g(x)$, it can predict N sample with zero error which is mean that $\sum_{j=1}^{\tilde{N}} \left\| o_j - t_j \right\| = 0$ which is mean $o_j = t_j$ then :

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i.x_j + b_i) = t_j, \quad j = 1, ..., N \quad (10)$$

$H\beta = T$ can be explained in Equation (11).

H is hidden layer of matrix output from neural network; i^{th} column from H is hidden output of i^{th} that connect with input x_1, x_2,...,x_N . $g(w_1 x_1 + b_1)$ is output of hidden neuron that connect with input x_i, β is the matrix of input weight and T is desired target or output.

$$H = (w_1, ..., w_{\tilde{N}}, b_1, ..., b_{\tilde{N}}, x_1, ..., x_N)$$

$$= \begin{bmatrix} g(w_1.x_1 + b_1) & \cdots & g(w_{\tilde{N}}.x_1 + b_{\tilde{N}}) \\ \vdots & \vdots & \vdots \\ g(w_1.x_N + b_1) & \cdots & g(w_{\tilde{N}}.x_N + b_{\tilde{N}}) \end{bmatrix}_{Nx\tilde{N}} \quad (11)$$

where $$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}xm} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{nxm}$$

Unlike conventional method, at *Extreme Learning Machine*, input weight (w_i) and hidden bias of layer (b_i) do not need tuning and hidden layer of matrix output (H) can be obtained without iteration. Output weight can be determined from $H\beta = T$ from the solution by using *Least-Square* (LS) with $\hat{\beta}$ for linear system can be seen in Equation (13)

$$\hat{\beta} = H^{\dagger} T \quad (13)$$

Denormalization

After output had been obtained from learning process, denormalization was performed, in accordance with (Zhu, Q.Y., dkk 2005), it can be formulated as :

$$X_d = 0.5\left(X_n + 1\right)\left(\max\left\{X_p\right\} - \min\left\{X_p\right\}\right) + \min\left\{X_p\right\} \quad (14)$$

With :
X_d= data value after denormalization
X_n = output data after denormalization
min {X_p}= minimum value of set data
max {X_p}= maximum value of set data

After learning that use *extreme learning machine* gave result, then *extreme learning machine* testing was performed with data that has never been taught before. Weight, bias and number of hidden were using weight, bias and number of hidden that had been used at learning process. To see the effectiveness of extreme learning machine method, error percentage calculation that occurred was also performed as seen in Equation (15).

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - t_i\right)^2 \quad (15)$$

where
N= number of data
y_i= estimate data output
t_i= actual weight data

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_{prediction} - y_{t\arg et}}{y_{t\arg et}}\right|100\% \qquad (16)$$

where

$y_{prediction}$	= prediction value JST
y_{target}	= actual value that occurred
N	= number of data that has been processed

From the simulation of proposed method, then calculate the speed of the simulation predictions using neural network and extreme learning machine. From the result, we can see that extreme learning machine can predict critical clearing time faster than neural network. Plotting picture from both method will show in the result.

SIMULATION

To examine the effectiveness of the method that has been used, simulation was perform at 3 Generator 9 Bus system that can be seen at this picture below. We call this system as System 1. The second simulation was done with Java Bali 500 kV 54-machine 25-bus system, and this system named as system 2

Simulations done by giving disturbance at some point and calculate the critical clearing time. Neural network method has been tested to perform prediction from critical clearing time above, and then prediction was performed again by extreme learning machine method. The result from both method above will be compare to see the effectiveness of them by calculating the speed of both of these methods in predicting the critical clearing time

RESULT AND DISCUSSION

On the system of Fouad and Anderson 3 Machine 9 Bus system, simulation was done by perform three phase short circuit as disturbance at several point in point A, B and C that we call as Fault 1, 2 and 3.

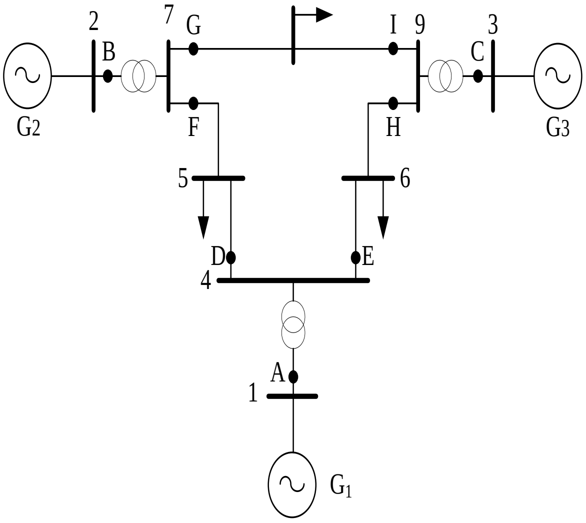

Figure 4. Fouad Anderson 3 Generator 9 Bus System [7]

We change one load bus with various capacity then give three phase short circuit in every load changes.

From the simulation that has been run, the prediction critical clearing time was obtained as seen in Table 1.

Table 1. Critical Clearing Time From Critical Trajectory Method on Fouad and Anderson 3 Machine 9 Bus System

INPUT		CCT ON (s)		
P (MW)	Q (Mvar)	FAULT 1	FAULT 2	FAULT 3
95	35	0.3485	0.2145	0.2335
105	45	0.3635	0.2165	0.2375
115	55	0.3805	0.2195	0.2405
125	65	0.3995	0.2215	0.2435
135	75	0.4205	0.2245	0.2475
145	85	0.4445	0.2265	0.2505
155	95	0.4715	0.2295	0.2535

Table 2. is the result of critical clearing time prediction using neural network. While table 3 is the result of the prediction using extreme learning machine. The results obtained showed that extreme learning machine capable of predicting critical clearing time with smaller error when compared with the prediction using neural network

Table 2. Prediction CCT Using Neural Network on Fouad and Anderson 3 Machine 9 Bus System

INPUT		CCT ON (s)		
P (MW)	Q (Mvar)	FAULT 1	FAULT 2	FAULT 3
95	35	0.3512	0.2148	0.2347
105	45	0.3631	0.2166	0.2373
115	55	0.3884	0.2199	0.2418
125	65	0.4050	0.2217	0.2447
135	75	0.4213	0.2234	0.2483
145	85	0.4425	0.2270	0.2495
155	95	0.4815	0.2293	0.2555

Table 3. Prediction CCT Using Extreme Learning Machine on Fouad and Anderson 3 Machine 9 Bus System

INPUT		CCT ON (s)		
P (MW)	Q (Mvar)	FAULT 1	FAULT 2	FAULT 3
95	35	0.3487	0.2151	0.2341
105	45	0.3633	0.2163	0.2368
115	55	0.3802	0.2194	0.2406
125	65	0.3979	0.2215	0.2439
135	75	0.4197	0.2245	0.2470
145	85	0.4441	0.2265	0.2509
155	95	0.4735	0.2289	0.2533

Calculation error of prediction using neural network gives the greatest error result is 0.0079. While extreme learning machine method gives a prediction error of 0.0244 for system 1. The graphic of critical prediction error can be seen on Figure 5.

Table 4. Comparison Speed Prediction System 1

INPUT		TIME SIMULATION (s)		
P (MW)	Q (Mvar)	CRITICAL TRAJECTORY	NN	ELM
95	35	0.7584	0.3387	0.0312
105	45	0.8154	0.3030	0.0468
115	55	0.8133	0.2852	0.0312
125	65	0.8034	0.3008	0.0312
135	75	0.7931	0.3053	0.0312
145	85	0.7975	0.3365	0
155	95	0.7995	0.3231	0.0312

Amount 500 neurons hidden layers on extreme learning machine and neural network with 24 hidden neurons use to predicting cct. Simulation repeated with amount of neurons neural network less than previous simulation. The results of the comparison of the speed predictions can be seen in the Figure 6.

The next step is to compare the speed of artificial intelligent in predicting critical clearing time. By changing the number of neurons in the hidden layer neural network, the simulation was doing to see the speed neural network in predicting critical clearing time and compared with the speed of extreme learning machine in predicting critical clearing time. The comparison of the speed from the proposed method can be seen on table 4.

safd

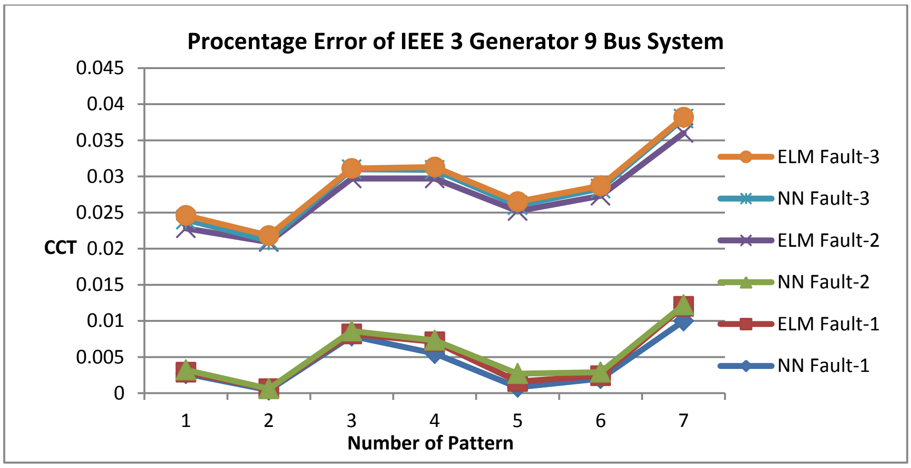

Figure 5. Percentage error prediction critical clearing time for system 1

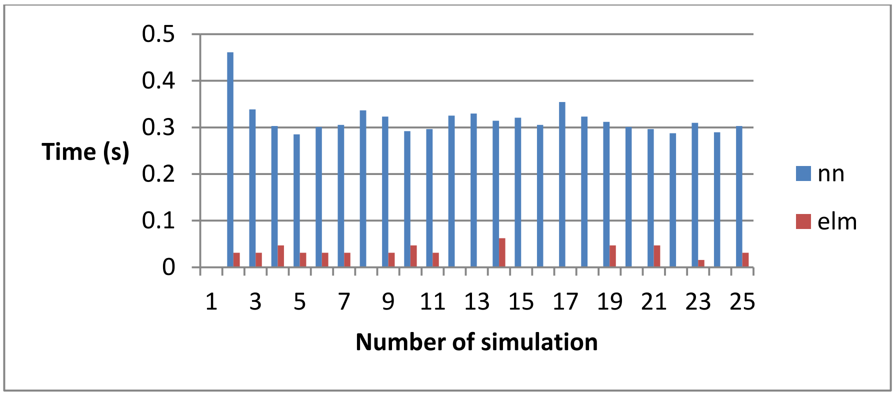

Figure 6. Prediction Time Simulation for System 2

From figure 6 we can see that the time required to predict the critical clearing time shows that the extreme learning machine is able to provide faster predictions than neural network.

Process training using neural network, with conventional gradient based learning algorithm like backpropagation (BP) determination of parameters such as the weight of input, hidden bias that connects between the layers to one another is determined manually, thus requiring a learning speed that is a long and often get stuck on local minima , Extreme learning machine perform a random selection process to select the input weights and hidden bias. It makes extreme learning machine has learing fast speed and give better results.

The next step is calculating the critical clearing time for Java Bali 500 kV 54-machine 25-bus system. This system we call as system 2 and can be seen on figure 7. Three phase short circuit is given on three points, on point B, C and G are referred to Fault 1, Fault 2 and Fault 3. The simulation results are to be obtained critical clearing time as follows on Table 5.

Table 5. Prediction CCT Using Critical Trajectory Method on Java Bali 500 kV 54-machine 25-bus system

INPUT		CCT ON (s)		
P (MW)	Q (Mvar)	FAULT 1	FAULT 2	FAULT 3
1162	355	0.6653	0.2546	0.1918
1187	340	0.6904	0.2557	0.1918
1207	360	0.7135	0.2563	0.1921
1232	385	0.7473	0.2571	0.1925
1252	405	0.7785	0.2579	0.1923
1272	425	0.8128	0.2587	0.193
1277	430	0.8219	0.2589	0.193
1297	450	0.861	0.2596	0.1929
1302	455	0.8716	0.2598	0.1931

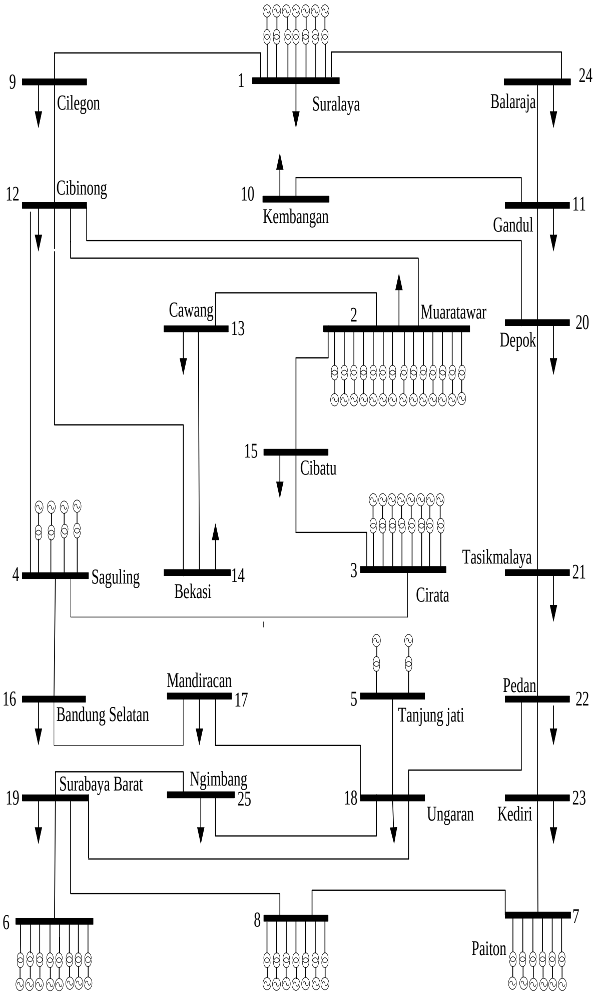

Figure 7. Java Bali 500 kV 54-machine 25-bus system

Table 6 is the result of critical clearing time prediction using neural network for Java Bali 500 kv 54 generator 25 bus system. While Table 7 is the result of prediction critial clearing time using extreme learning machine.

Table 6. Prediction CCT Using Neural Network on Java Bali 500 kV 54-machine 25-bus system

INPUT		CCT ON (s)		
P (MW)	Q (Mvar)	FAULT 1	FAULT 2	FAULT 3
1162	355	0.6678	0.2545	0.1906
1187	340	0.6936	0.2554	0.1914
1207	360	0.7159	0.2564	0.1917
1232	385	0.7402	0.2570	0.1926
1252	405	0.7839	0.2580	0.1921
1272	425	0.8076	0.2587	0.1934
1277	430	0.8128	0.2588	0.1937
1297	450	0.8388	0.2592	0.1928
1302	455	0.8319	0.2591	0.1931

Table 7. Prediction CCT Using Extreme Learning Machine on Java Bali 500 kV 54-machine 25-bus system

INPUT		CCT ON (s)		
P (MW)	Q (Mvar)	FAULT 1	FAULT 2	FAULT 3
1162	355	0.6542	0.2546	0.1923
1187	340	0.6862	0.2556	0.1924
1207	360	0.7170	0.2561	0.1922
1232	385	0.7537	0.2570	0.1933
1252	405	0.7769	0.2572	0.1938
1272	425	0.8043	0.2577	0.1938
1277	430	0.8128	0.2579	0.1938
1297	450	0.8528	0.2596	0.1946
1302	455	0.8634	0.2601	0.1951

Error prediction critical clearing time on Java Bali 500 kv can be seen on Figure 8.

As simulation on system 1, we count time simulation neural network and extreme learning machine predicting critical clearing time on Java Bali 500 kv 54 machine 25 bus. The speed of both methods observed to find out which method can predict faster.

Table 8. Comparison Speed Prediction System 2

INPUT		TIME SIMULATION (s)	
P (MW)	Q (Mvar)	NN	ELM
95	35	0.3387	0.0312
105	45	0.3030	0.0468
115	55	0.2852	0.0312
125	65	0.3008	0.0312
135	75	0.3053	0.0312
145	85	0.3365	0
155	95	0.3231	0.0312

The result of the calculation speed of both methods in predicting the critical clearing time can be plot in Figure 9.

From the simulation results shown in figure 6 and figure 9 shows that the extreme learning machine is able to predict critical clearing time faster than neural network. By changing the number of hidden layers in neural network, results obtained showed extreme learning machine fixed faster and gives accurate results.

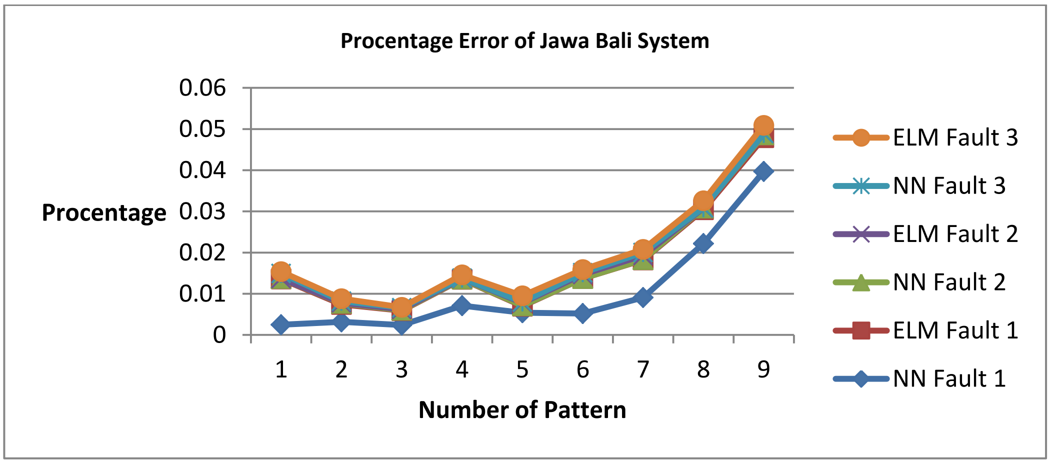

Figure 8 Percentage error prediction critical clearing time for system 2

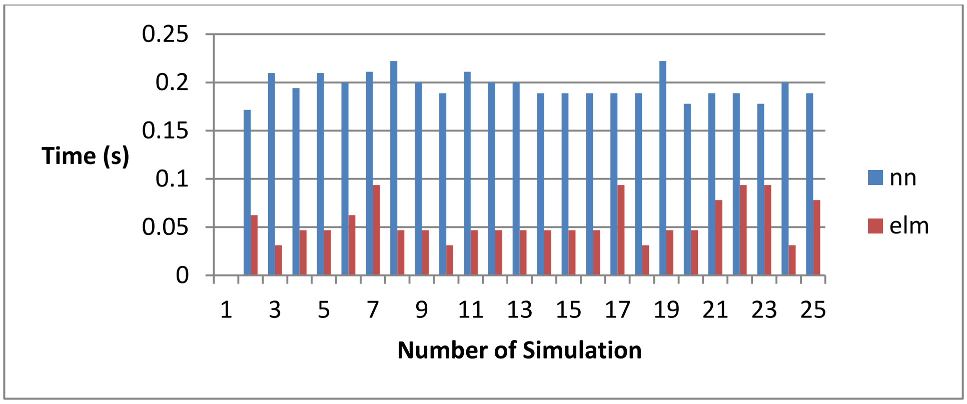

Figure 9. Prediction Time Simulation for system 2

CONCLUSION

From the simulation results show that the prediction of critical clearing time using extreme learning machine method proven faster and more accurate when compared with neural network method. Therefore, when used to predict critical clearing time in real time, the proposed method can be used.

REFERENCES

[1] Karami, A., "Power System Transient Stability Margin Estimation Using Neural Network", *Journal of Electrical Power and energy Systems*, vol.9, pp. 983-991, 2011.

[2] A. L. Bettiol, A. Souza, J. L. Todesco, J. R. Tesch Jr, "Estimation of Critical Clearing Times Using Neural Networks", *IEEE Bologna PowerTech Conference*, Bologna, Italy, June 23-26, 2003

[3] Lin, Y.J., "Explaining Critical Clearing Time With The Rules Extracted From A Multilayer Perceptron Artificial Neural Network", *Journal of Electrical Power and Energy Systems*, vol. 32, 2010

[4] Huang, G.B., Zhu, Q.Y. and Siew, C.K.,"Extreme Learning Machine: A New Learning Scheme of Feed-forward Neural Networks", *Proceedings of International Joint Conference on Neural Networks*, 2004.

[5] Huang, G.B., Zhu, Q.Y. and Siew, C.K., "Extreme Learning Machine: Theory and Application", *Journal of Science Direct neuro-computing*, vol. 70, pp. 489-501, 2006.

[6] Zhu, Q.Y., Qin, A.K., Suganthan, P.N. and Huang, G.B, "Evolutionary Extreme Learning Machine", *The Journal of The Pattern Recognition Society*, vol. 38, pp. 1759-1763, 2005.

[7] Anderson PM, Fouad AA. Power System Control and Stability. Iowa State: University Press: Ames, IA, 1977.