Extreme Learning Machine Neural Networks for Multi-Agent System in Power Generation

Chong Tak Yaw¹, Shen Yuong Wong²*, Keem Siah Yap³, Chin Hooi Tan⁴

¹Department of Electronics and Communication Engineering, Universiti Tenaga Nasional, Malaysia.
²School of Electrical and Computer Engineering, Xiamen University, Malaysia.
³Department of Electronics and Communication Engineering, Universiti Tenaga Nasional, Malaysia.
⁴Tenaga Nasional Berhad, Kuala Lumpur, Malaysia.
*Corresponding author E-mail: yapkeem@uniten.edu.my

Abstract

Extreme Learning Machine (ELM) is widely known as an effective learning algorithm than the conventional learning methods from the point of learning speed as well as generalization. The hidden neurons are optional in neuron alike whereas the weights are the criteria required to study the linking among the output layer as well as hidden layers. On the other hand, the ensemble model to integrate every independent prediction of several ELMs to produce a final output. This particular approach was included in a Multi-Agent System (MAS). By hybrid those two approached, a novel extreme learning machine based multi-agent systems (ELM-MAS) for handling classification problems is presented in this paper. It contains two layers of ELMs, i.e., individual agent layer and parent agent layer. Several activation functions using benchmark datasets and real-world applications, i.e., satellite image, image segmentation, fault diagnosis in power generation (including circulating water systems as well as GAST governor) were used to test the ELM-MAS developed. Our experimental results suggest that ELM-MAS is capable of achieving good accuracy rates relative to others approaches.

Keywords: Extreme Learning Machine; Power System Generation; Multi-Agent System.

1. Introduction

Lately, Extreme Learning Machine (ELM) is widely known as an effective learning algorithm than the conventional learning methods from the point of learning speed as well as generalization [1-6]. ELM has the capability to make universal approximation with haphazard biases and input weights [7]. Otherwise speaking, the hidden neurons are optional in neuron identical whereas the weights are the criteria required to study the connection between the output layer and hidden layers. According to Huang et al. [8], the ELM is extremely effective and inclines to universal prime in contrast to the conventional feedforward neural network (FNN). Furthermore, ELM is able to achieve the greatest generalization bound of the conventional FNN, in which each parameter was learned with activation functions that were normally utilized [9]. In the context of generalization and efficiency, ELM performed much better than that of traditional FNN, as evidenced in diverse types of problems [1-6]. The ELM can also be applied to other fields, namely hyperspectral images [10], biomedical analysis [11-12], system modelling [13-14], chemical process [15], action recognition [16], power systems [17], and others.

The focus of some research group was on ensemble model to integrate every independent prediction of several ELMs to produce a final output [18-22]. This particular approach was included in a Multi-Agent System (MAS) as well [23]. In the past decade, Multi-Agent Systems (MASs) had been a focus of attention. It had been successfully applied by researchers for tackling problems in different domains, as shown by its extensive applications in decision support [24], military support [25], healthcare [26], control systems [27], e-Commerce [28], and knowledge management [29]. The MAS’s general structure was illustrated in Figure 4.1, in which the base platform was consisted of a collection of ELMs known as individual agents. Normally, an individual ELM (individual agent’s) outcome was delivered to a parent agent, which was the decision combination module where the ultimate decision was made in this structure.

In general, the average output in the decision combination module was derived from the methods including weighted average [7], exact average [20], voting [30], and confusion matrix [31]. Regrettably, in the decision combination module, these aforementioned approaches often needed additional algorithms to generate outcome. In this paper, decision combination module based on ELM was proposed. Meta-learning can roughly be described in terms of at least a learners are learning information, where it is a common approach used to combine the outcomes of multiple learners [32-33]. The model is achieved by multiple ELMs as hidden neurons, while the meta-learner learns from the outputs of hidden neurons. Theoretical analysis and experimental results from several studies using artificial and benchmark regression datasets illustrated that Meta-ELM, which is trained by multiple ELMs, could give good performance with a lower computational cost [34]. Meta-ELM [34] is an ELM with a special design, i.e., used ELMs as hidden neurons. However, in this paper, an ELM-MAS (extreme learning machine based multi-agent systems) is designed from another perspective. It has two layers of full ELMs: the first layer made up of at least an ELMs and each ELM is considered as an individual agent; the second layer consists of a single ELM and it is the parent agent. This two layers structure of the proposed ELM-MAS resembles a typical multi-agent neural network (Fig. 1).
2. The Algorithms of ELM and ELM-MAS

Depending on the type of activation function it utilizes, an ELM can be either a feedforward or an RBF network with a sophisticated learning algorithm (Fig. 2). Deliberate a series of $N$ training samples (with an input vector and corresponding target output vector), $(x_j,t_j)$, i.e., $x_j \in \mathbb{R}^M$ ($M$ is number of input attributes) and $t_j \in \mathbb{R}^C$ ($C$ is number of classes) are used to train an ELM that consists of $L$ number of hidden neurons. Five ELMs are generated as individual agents in this paper, with each having different random input weights. As shown in Fig. 3, the output of each ELM ($k$ for $k = 1, 2, \ldots, 5$), in response to $x_j$ is

$$EML_k(x_j) = \frac{\sum_{i=1}^{L} \beta_{ik}G(\alpha_{ik},b_{ik},x_j)}{1+\exp\{-\alpha_{ik}\cdot x_j + b_{ik}\}}, \quad \beta_{ik} \in \mathbb{R}$$  \hspace{1cm} (1)

where $\alpha_{ik}$ and $b_{ik}$ are the input weights and bias (learning parameters) of the hidden neurons, $\beta_{ik}$ is the output weights, and $G(\alpha_{ik},b_{ik},x_j)$ is the output of the $i$th hidden neuron given the input vector $x_j$.

Definition of the $G(\alpha_{ik},b_{ik},x_j)$ for additive hidden neuron and RBF neuron, was shown in Equations (2) and (3), respectively.

$$G(\alpha_{ik},b_{ik},x_j) = \frac{1}{1+\exp\{-\alpha_{ik}\cdot x_j + b_{ik}\}}, \quad b_{ik} \in \mathbb{R}$$  \hspace{1cm} (2)

$$G(\alpha_{ik},b_{ik},x_j) = \exp\{-b_{ik}\cdot [x_j - a_{ik}^T]^2\}, \quad b_{ik} \in \mathbb{R}^+$$  \hspace{1cm} (3)

The procedure of the training phase is as follows.

**Step 1:** Haphazardly designate the input weights $a_{ik}^j$ and $b_{ik}^j$ for $i = 1, \ldots, L$ and $k = 1, 2, \ldots, 5$

**Step 2:** Calculation of the hidden layer output matrix for $EML_k$ (for $k = 1, 2, \ldots, 5$), $H^k$, as follows.

$$H^k = \begin{bmatrix} G(a_{i1}^k,b_{i1}^k,x_1) & \ldots & G(a_{i1}^k,b_{i1}^k,x_N) \\ \vdots & & \vdots \\ G(a_{iL}^k,b_{iL}^k,x_1) & \ldots & G(a_{iL}^k,b_{iL}^k,x_N) \end{bmatrix}_{N \times L}$$  \hspace{1cm} (4)

**Step 3:** Computation of the output weights of $EML^j$, $\beta^j$. Due to the reason that $H$ is possibly a non-symmetrical matrix, the inverse matrix cannot be solved. As results, a Moore-Penrose pseudo inverse matrix method was adopted to circumvent this problem, which was demonstrated by the equation below,

$$\beta^j = (H^j)^T(H^j)^{-1}(H^j)^T T,$$  \hspace{1cm} (5)

where $T = [t_1, \ldots, t_N]^T$ is the corresponding targeted output vectors.

**Step 4:** After the calculation of $EML^j$’s output weights, the outputs of $EML^j$ were computed using the training samples.

$$x^j = EML^j(x_j) = \sum_{i=1}^{L} \beta_{ij}G(\alpha_{ij},b_{ij},x_j) \quad \text{for} \quad j = 1, \ldots, N, \quad c = 1, \ldots, C$$  \hspace{1cm} (6)

**Step 5:** The input weights for parent ELM, i.e., $p_i$ and $q_i$ for $i = 1, \ldots, L$, where $L_i$ is the number of hidden neuron of parent ELM are assigned at random.

**Step 6:** Hidden layer output matrix for parent ELM, $S$, is computed as below

$$S = \begin{bmatrix} G(p_1,q_1,w_1) & \ldots & G(p_{L_1},q_{L_1},w_1) \\ \vdots & & \vdots \\ G(p_{1},q_{N},w_N) & \ldots & G(p_{L_1},q_{L_1},w_N) \end{bmatrix}_{N \times L_1}$$  \hspace{1cm} (7)

where $w_i$ is the combination outputs of $EML^k$ (for $k = 1, 2, \ldots, 5$) in response to $x_j$, i.e., $w_j = [y_1^j \ y_2^j \ y_3^j \ y_4^j \ y_5^j]$, $y_k^j \in \mathbb{R}^C$, and $w_j \in \mathbb{R}^{5C}$.

**Step 7:** Used the output of $EML^k$ to calculate the output weights of parent ELM, $a$ by the equation below,

$$a = [S^T(S)^{-1}]^T T,$$  \hspace{1cm} (8)

where $T = [t_1, \ldots, t_N]^T$ is the particular targeted output vectors.
Once the training of all samples were completed following Step 1 to Step 7, prediction of unknown input vector \( z \) based on the \( a_k, b_k, p, q, \beta \), and \( \alpha \) can be done using the ELM-MAS i.e.,

\[
h^k = [G(a_k^b, b_k, z)] \rightarrow G(a_k^b, b_k, z)_{v,L} \tag{9}
\]

\[
y^k = h^k \beta^k \tag{10}
\]

\[
s = [G(p, q, v)] \rightarrow G(p, q, v)_{v,L} \tag{11}
\]

\[
\gamma = sa \tag{12}
\]

where \( h \) and \( y \) are hidden layer output and of ELM respectively, \( v = [y^1 y^2 y^3 y^4 y^5] \) and \( s \) and \( \gamma \) are hidden layer of final output of the prediction respectively. Besides equations (2) and (3), there are several activation functions that have been used in this paper, i.e.,

\[
G(a_k^b, b_k, x_j) = \exp\left[-\left|x_j + b_k^j\right|^2\right], \quad b_k^j \in R \tag{13}
\]

\[
G(a_k^b, b_k, x_j) = \exp\left[-(a_k^j \cdot x_j + b_k^j)\right], \quad b_k^j \in R \tag{14}
\]

\[
G(a_k^b, b_k, x_j) = \exp\left[-(a_k^j \cdot x_j - a_k^j)\right], \quad b_k^j \in R^+ \tag{15}
\]

Equations (13), (14), and (15) are the Gaussian activation function, Laplacian activation function and Laplacian Basis function. The proposed ELM-MAS and the Meta-ELM [34] have a similar structure. However, they have subtle distinctions in structural representation, and also differences in the way they handle training datasets. Firstly, Meta-ELM partitions the entire training dataset into several random subsets and each ELM (hidden neurons) learns a subset. However, all ELMs (individual agents) of the proposed ELM-MAS trained by the same entire training dataset. Secondly, ELMs in Meta-ELM are considered as hidden neurons. But for ELM-MAS, ELMs are considered as individual agents. Lastly, the Meta-ELM is a three layers neural network, but ELM and parent ELM of ELM-MAS have a full three layers neural network structure hence total of six layers. As a conclusion, the structure of Meta-ELM is the same as a standard ELM (as shown in Fig. 2) but with ELM in the hidden neuron. This is different from the proposed ELM-MAS with two layers of ELMs (six layers of neurons) which formed a multi-agent system.

3. Experimental and Results using Benchmark Data

In this experiment, we refer to the suggestion of Liang [35] to fix the number of hidden neuron of each ELM (i.e., \( L \)) to 400 for Satellite Image and 180 for Image Segmentation. On the other hand, 1/3 of the training samples were utilized for training while 1/3 were used to decide the optimum amount of neurons of the parent ELM (i.e., \( L_p \)) via a validation process. For each type of the activation function of ELM-MAS, training and validation processes start by setting \( L_k = 50 \) units and then increased by an increment of 50 units. As an example, Table II shows the training and validation processes based on sigmoid activation function. From the Table II, the number of hidden neurons with the most excellent validation result is chosen for ELM-MAS’s performance evaluation.

In the Table II, the snowballing amount of hidden neuron is not enriched the accuracy rate. This was due to a condition termed overfitting, where the neural networks overestimate the targeted problem’s complexity. In contrast, it also significantly reduced generalization capability, which resulted in substantial variation in predictions. Hence, using feedforward neural network to allocate the proper number of hidden neurons with the intention to avoid overfitting was of the utmost importance in function approximation.

In general, ELMs in Meta-ELM are considered as hidden neurons. The proposed ELM-MAS with two layers of ELMs (six layers of neurons) which formed a multi-agent system.

#### Table II: Validation Accuracy Rates of Sigmoid Activation Function

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Lₜ</th>
<th>Validation Accuracy (%)</th>
<th>Time (s)</th>
<th>Validation Accuracy (%)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellite Image</td>
<td>50</td>
<td>89.35</td>
<td>2.38</td>
<td>94.52</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>89.53</td>
<td>2.40</td>
<td>94.20</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>89.45</td>
<td>2.39</td>
<td>94.21</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>89.50</td>
<td>2.43</td>
<td>93.77</td>
<td>0.37</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>89.40</td>
<td>2.52</td>
<td>93.50</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table II outlined the results by means of ELM-MAS in the context of the training time (seconds), test accuracy, as well as the number of hidden neurons for all activation functions.

#### Table III: Test Accuracy Rates of the ELM-MAS with Different Activation Functions

<table>
<thead>
<tr>
<th>Activation Function</th>
<th>Test Accuracy (%)</th>
<th>Time (s)</th>
<th># Hidden Neurons</th>
<th>Test Accuracy (%)</th>
<th>Time (s)</th>
<th># Hidden Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>88.71</td>
<td>7.51</td>
<td>150</td>
<td>94.70</td>
<td>0.56</td>
<td>100</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>89.27</td>
<td>2.53</td>
<td>200</td>
<td>94.48</td>
<td>0.30</td>
<td>50</td>
</tr>
<tr>
<td>Gaussian</td>
<td>89.47</td>
<td>2.97</td>
<td>400</td>
<td>94.62</td>
<td>0.36</td>
<td>100</td>
</tr>
</tbody>
</table>

Laplace Act.        | 89.96             | 2.40    | 150             | 94.87             | 0.38    | 200             |
Laplace Basis       | 89.86             | 7.32    | 100             | 95.39             | 0.90    | 250             |

Table III outlined the results by means of ELM-MAS in the context of the training time (seconds), test accuracy, as well as the number of hidden neurons for all activation functions.

#### Table IV: Comparison with Other ELM networks

<table>
<thead>
<tr>
<th>Activation Function</th>
<th>Satellite Image</th>
<th>Image Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>89.31</td>
<td>14</td>
</tr>
<tr>
<td>ELM-MAS</td>
<td>89.71</td>
<td>7.51</td>
</tr>
<tr>
<td>Ensemble-ELM [20]</td>
<td>89.28</td>
<td>343.85</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>88.93</td>
<td>302.48</td>
</tr>
<tr>
<td>ELM-MAS</td>
<td>89.27</td>
<td>2.53</td>
</tr>
<tr>
<td>Ensemble-ELM [20]</td>
<td>89.01</td>
<td>1591.76</td>
</tr>
</tbody>
</table>

Table IV compared other ELM networks with the proposed ELM-MAS.
Comparison was also made between the proposed ELM-MAS and other variants of ELMs, such as ELM [35] and ensemble ELM [20]. Based on Table IV, the test accuracy rates of ELM-MAS were comparable (if not greater) with ELM (RBF) and ELM (Sigmoid). In term of processing times, the comparisons between ELM and ELM-MAS was inconclusive because of improvement of computer hardware in past several years. In general, the processing time of ELM-MAS is considerably fast (within few seconds). In addition, it can be faster if training of all ELMs were conducted in parallel.

4. Application in Power Generation

Application of the developed ELM-MAS on power generation system is discussed in the following section.


![Circulating Water System](image1)

Fig. 4 demonstrated the Circulating Water System (CWS) in a power generation plant of Tenaga Nasional Berhad (TNB) located at Penang, Malaysia [36-37]. As shown, the system comprised of turbine condensers, drum strainer, and piping between the sea water’s inlet and the drain, where water will be sent back into the sea. Turbine condenser is the major component in the CWS which simultaneously function in the removal of heat from low pressure steam as well as the maintenance of the turbine backpressure at the bottom potential nevertheless constant level. Undeniably, the heat transfer’s efficiency in a condenser had a major impact on the condenser vacuum. Maintenance of the turbine backpressure at a low level via an effective heat transfer process allowed the turbine to work at high efficiency for power generation. Nonetheless, in the case of excessive amount of gases due to cooling presented in the condenser, the vacuum level would be affected and the heat transfer efficiency in the condenser would be reduced. Additionally, hygiene of the condenser tubes had substantial effect on the capacity of the condenser to transfer heat from the exhaust steam to the cooling water. Accumulation of mud plus fine solid materials like sand, shells and seaweed in the water which slipped away during the filtering process of the circulating water system will be resulted in blockage, which ultimately led to inefficient heat transfer in the condenser tubes.

A series of 2500 datasets were gathered and segregated into training, validation, and testing sets (Table V) [38]. Before the tests, the ELM-MAS model was trained and validated to attest the most appropriate number of hidden neurons. As shown in Table VI, training ELM-MAS with a Laplacian activation function at 96.96% attained the highest test accuracy. The test accuracy rate of ELM-MAS is comparable (if not superior) to other approaches including FAM [39] and SVM [40] (Table VII).

4.2. GAST Governor

No more Modern power plants could generate and supply high quality electricity to customers. Many computers had been installed with simulation programs to analyze the characteristics of power systems in the planning phase as well as the actual operations which included monitoring and control activities. System analysis software was executed repeatedly in the planning stage of a power plant [41]. Modification and adjustment of the input data into the software were made based on the engineers’ experiences and heuristic knowledge till satisfactory plans had been determined. Currently the development of the programs in the power plant analysis as well as planning are according to the mathematical models besides only implemented for the numerical computation. Thus, development of techniques and methodologies for
incorporating practical knowledge of planning were used in sophisticated approaches and were applied into the system planning [41].

Computer based in Energy Management Systems were frequently introduced into gas turbine monitoring and control. Besides that, these computer-based systems were commonly utilized in energy control center. Gas Turbine analysis software as well as other application software was being introduced into the Energy Management Systems with intentions of examining as well as forecasting the behaviour of gas turbines during steady-state operations [41]. Although these software was a powerful tool, its abilities in assisting the operation engineers in making the best decisions were very limited during the time when unplanned or unexpected modes of two operations were detected. Most of the abnormal modes in the system operation were triggered by either reactive or active power imbalances, network faults or frequency deviations. A partial or complete systems blackout can be happened in an unplanned operation [41]. Thus, experienced operation engineers will be making the decisions under these emergency situations for restoration of the gas turbine into normal state. There is a need to incorporate the knowledge of the experienced operation engineers with conventional application software into a situation such as operational strategies for network restoration, efficiency in diagnosis of network faults, and balancing reactive and active power [41]. Therefore, the development of fast and efficient methods of predicting abnormal system behaviour is required.

In recent years, Malaysia had experienced several large-scale blackout incidents [42]. In the latest incident occurred in 2005, several gas turbine plants sequentially set off inadvertently following a frequency drop of about 1.5 Hz, which caused a total generation loss of 5760 MW. Following this, several studies were performed to observe responses of combined cycle power plants in relation to frequency drops [42-45]. These models were based on gas turbine models developed by Rowen [46] and Mello et al. [47] to simulate practical plants, which are then utilized to compute responses to frequency changes. Nevertheless, comprehensive analysis on behavior of plant variables during frequency drops has not been done.

The major priority for most power generation companies is to attain maximum availability of the gas turbine. To attain maximum availability, the company must prevent accidental shutdowns and if accidental shutdown does occur, the recovering time must be minimized. Consistent monitoring of the gas turbine condition is the best method to achieve maximum availability so that minor problems can be detected before they evolve into major problems. In this case it is believed that an Artificial Intelligent gas turbine condition monitoring system can be used to determine the condition of gas turbine parameters during contingencies to minimize trouble shooting time and restore the gas turbine to its normal operating condition.

The GAST denotes the key dynamic features of industrial gas turbines driving generators linked to electric power systems. Speed differences from nominal were envisioned designate minute (roughly 5%). Figure 5 illustrated the model, which comprised of a forward path with governor time constant, T1, as well as a combustion chamber’s time constant, T2, in addition to a load-limiting feedback path. The load limit was susceptible to turbine exhaust temperature, whereby T3, indicated the time constant of the exhaust gas measuring system. The constant, K1, was utilized for adjusting the gain of the load-limited (A0) feedback path.

The training data is assembled on the output of the GAST block, which is the mechanical power, Pmech, for a normal operating gas turbine [48]. A total of 630 data were gathered for all 7 input attributes in the GAST with variation within their operating range values [49] (Table VIII). As shown in Table IX, the data were pre-allocated into training, validation, as well as test sets.

Table VIII: Details of the 7 Attributes in GAST

<table>
<thead>
<tr>
<th>Class</th>
<th>Range</th>
<th># Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 &lt; R &lt; 1</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>0 &lt; Dmax &lt; 0.5</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>0 &lt; A2 &lt; 1</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>0.01 &lt; T1 &lt; 0.5</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>0.01 &lt; T2 &lt; 0.5</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>0.01 &lt; T3 &lt; 0.5</td>
<td>140</td>
</tr>
<tr>
<td>7</td>
<td>0 &lt; K1 &lt; 0.5</td>
<td>140</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>630</td>
</tr>
</tbody>
</table>

Table IX: The Pre-allocated with their respective indications of the GAST dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th># Training</th>
<th># Validation</th>
<th># Testing</th>
<th># Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>20</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>10</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>20</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>10</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>10</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>56</td>
<td>28</td>
<td>56</td>
<td>140</td>
</tr>
<tr>
<td>7</td>
<td>56</td>
<td>28</td>
<td>56</td>
<td>140</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td>630</td>
</tr>
</tbody>
</table>

Table X displayed the experimental results for utilizing ELM- MAS in the context of the test accuracy, training time (seconds), as well as the number of hidden neurons for every activation function in GAST. Laplace Basis activation function attained the greatest test accuracy rate of 76.79%.
Table X: Experimental Results for Different Activation Function in GAST

<table>
<thead>
<tr>
<th>Activation Function</th>
<th>Test Accuracy (%)</th>
<th>Time (s)</th>
<th># Hidden Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>74.31</td>
<td>0.09</td>
<td>70</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>76.68</td>
<td>0.09</td>
<td>104</td>
</tr>
<tr>
<td>Gaussian</td>
<td>69.21</td>
<td>0.10</td>
<td>101</td>
</tr>
<tr>
<td>Laplace Act.</td>
<td>72.74</td>
<td>0.10</td>
<td>94</td>
</tr>
<tr>
<td>Laplace Basis</td>
<td>76.79</td>
<td>0.10</td>
<td>65</td>
</tr>
</tbody>
</table>

5. Conclusion

In this study, a novel ELM-MAS model with two layers of ELMs is established. The established model was validated by utilizing benchmark datasets such as satellite image and image segmentation. According to the results, the test accuracy rates of ELM-MAS were comparable (if not superior) to ELM (RF) and ELM (Sigmoid). Furthermore, the developed model was evaluated by applying it on the power generation system including circulating water systems as well as governor (GAST). Thus far, our results demonstrated that the test accuracy rates of ELM-MAS for circulating water systems was comparable (if not superior) to other algorithms. Although results obtained from the benchmark studies (using satellite image and image segmentation datasets) as well as applications in power generation (circulating water systems and governor, GAST) were reassuring, further research with datasets from diverse application fields were essential for validating the suitability of ELM-MAS application in actual world.

Acknowledgement

This work was funded by Universiti Tenaga Nasional (UNITEN) under grant number JS10050684.

References

[31] Marom ND, Rokach L, Shmilovici A (2010), Using the confusion matrix for improving ensemble classifiers, Electrical and Electronics Engineers in Israel 26th (IEEEI), pp.555-559.


