International Journal of Engineering & Technology, 7 (4.35) (2018) 347-353

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



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

Chong Tak Yaw[1], Shen Yuong Wong[2*], Keem Siah Yap[3], Chin Hooi Tan[4]

[1]*Department of Electronics and Communication Engineering, Universiti Tenaga Nasional, Malaysia.*
[2]*School of Electrical and Computer Engineering, Xiamen University, Malaysia.*
[3]*Department of Electronics and Communication Engineering, Universiti Tenaga Nasional, Malaysia.*
[4]*Tenaga 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).

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

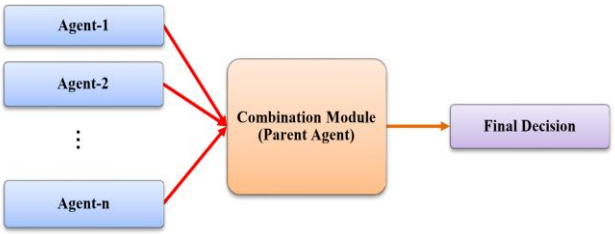


Fig. 1: Multi-agent Neural Network Scheme

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 a RBF network with a sophisticated learning algorithm (Fig. 2). Deliberate a series of N training samples (with a input vector and corresponding target output vector), $(\mathbf{x}_j, \mathbf{t}_j)$, i.e. $\mathbf{x}_j \in \mathbf{R}^M$ (M is number of input attributes) and $\mathbf{t}_j \in \mathbf{R}^C$ (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, ..., 5), in response to $\mathbf{x}_j$ is

$$\mathrm{ELM}^k(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_{ic}^k G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \mathbf{t}_j \quad \text{for } j = 1, \ldots, N \text{ and for } c = 1, \ldots, C \tag{1}$$

where $\mathbf{a}_i^k$ and b_i^k are the input weights and bias (learning parameters) of the hidden neurons, $\boldsymbol{\beta}_i^k$ is the output weights, and $G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j)$ is the output of the i^{th} hidden neuron given the input vector $\mathbf{x}_j$.

Definition of the $G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j)$ for additive hidden neuron and RBF neuron, was shown in Equations (2) and (3), respectively.

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \frac{1}{1+\exp\{-(\mathbf{a}_i^k \cdot \mathbf{x}_j + b_i^k)\}}, \quad b_i^k \in R \tag{2}$$

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-b_i^k \left\| \mathbf{x}_j - \mathbf{a}_i^k \right\|^2\}, \quad b_i^k \in R^+ \tag{3}$$

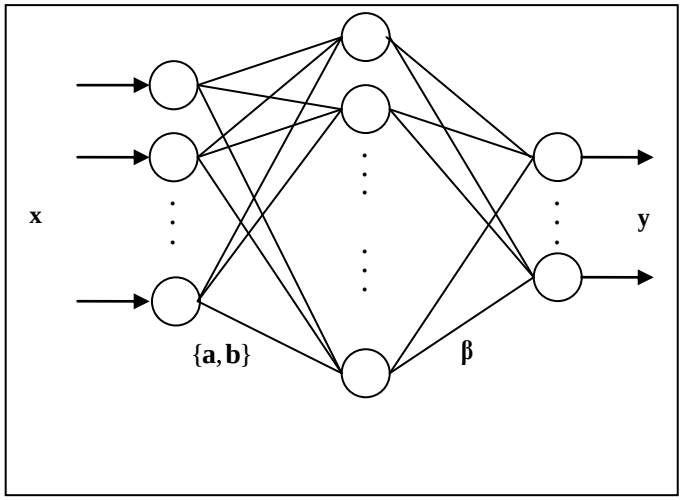


Fig. 2: Architecture of an ELM

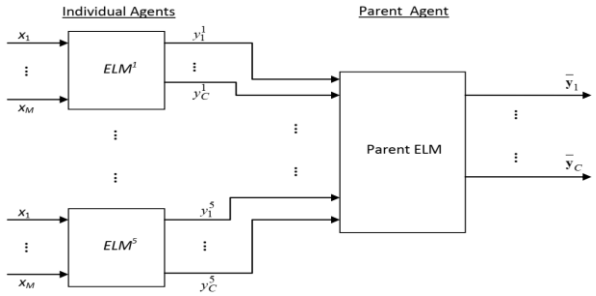


Fig. 3: Architecture of ELM-MAS with Several Individual Agents and a Parent Agent.

The procedure of the training phase is as follows.

Step 1: Haphazardly designate the input weights $\mathbf{a}_i^k$ and b_i^k for i = 1,… L and k = 1, 2, ...5

Step 2: Calculation of the hidden layer output matrix for ELM^k (for k = 1, 2, ...5), $\mathbf{H}^k$, as follows.

$$\mathbf{H}^k = \begin{bmatrix} G(\mathbf{a}_1^k, b_1^k, \mathbf{x}_1) & \ldots & G(\mathbf{a}_L^k, b_L^k, \mathbf{x}_1) \\ \vdots & \ldots & \vdots \\ G(\mathbf{a}_1^k, b_1^k, \mathbf{x}_N) & \ldots & G(\mathbf{a}_L^k, b_L^k, \mathbf{x}_N) \end{bmatrix}_{N \times L} \tag{4}$$

Step 3: Computation of the output weights of ELM^k, $\boldsymbol{\beta}^k$. Due to the reason that $\mathbf{H}$ 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,

$$\boldsymbol{\beta}^k = \left((\mathbf{H}^k)^T (\mathbf{H}^k)\right)^{-1} (\mathbf{H}^k)^T \mathbf{T}, \tag{5}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^T$ is the corresponding targeted output vectors.

Step 4: After the calculation of ELM^k's output weights, the outputs of ELM^k were computed using the training samples.

$$\mathbf{y}^k = \mathrm{ELM}^k(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_{ic}^k G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) \text{ for } j = 1, \ldots, N, \; c = 1, \ldots, C \tag{6}$$

Step 5: The input weights for parent ELM, i.e., $\mathbf{p}_i$ and q_i for i = 1,… L_1, where L_1 is the number of hidden neuron of parent ELM are assigned at random.

Step 6: The hidden layer output matrix for parent ELM, S, is computed as below

$$\mathbf{S} = \begin{bmatrix} G(\mathbf{p}_1, q_1, \mathbf{w}_1) & \ldots & G(\mathbf{p}_{L1}, q_{L_1}, \mathbf{w}_1) \\ \vdots & \ldots & \vdots \\ G(\mathbf{p}_1, q_1, \mathbf{w}_N) & \ldots & G(\mathbf{p}_{L_1}, q_{L_1}, \mathbf{w}_N) \end{bmatrix}_{N \times L_1} \tag{7}$$

where $\mathbf{w}_j$ is the combination outputs of ELM^k (for k = 1, 2,...5) in response to $\mathbf{x}_j$, i.e., $\mathbf{w}_j = [\mathbf{y}_j^1 \quad \mathbf{y}_j^2 \quad \mathbf{y}_j^3 \quad \mathbf{y}_j^4 \quad \mathbf{y}_j^5]$, $\mathbf{y}_j^k \in \mathbf{R}^C$, and $\mathbf{w}_j \in \mathbf{R}^{5C}$.

Step 7: Used the output of ELM^k to calculate the output weights of parent ELM, $\boldsymbol{\alpha}$ by the equation below,

$$\boldsymbol{\alpha} = \left((\mathbf{S})^T (\mathbf{S})\right)^{-1} (\mathbf{S})^T \mathbf{T}, \tag{8}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^T$ 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 $\mathbf{a}^k$, $\mathbf{b}^k$, $\mathbf{p}$, $\mathbf{q}$, $\boldsymbol{\beta}^k$ and $\boldsymbol{\alpha}$ can be done using the ELM-MAS i.e.,

$$\mathbf{h}^k = [G(\mathbf{a}_1^k, b_1^k, \mathbf{z}) \quad \ldots \quad G(\mathbf{a}_L^k, b_L^k, \mathbf{z})]_{1\times L} \tag{9}$$

$$\mathbf{y}^k = \mathbf{h}^k \boldsymbol{\beta}^k \tag{10}$$

$$\mathbf{s} = [G(\mathbf{p}_1, q_1, \mathbf{v}) \quad \ldots \quad G(\mathbf{p}_L, q_L, \mathbf{v})]_{1\times L1} \tag{11}$$

$$\bar{\mathbf{y}} = \mathbf{s}\boldsymbol{\alpha} \tag{12}$$

where $\mathbf{h}^k$ and $\mathbf{y}^k$ are hidden layer output and of ELM^k respectively, $\mathbf{v} = [\mathbf{y}^1 \quad \mathbf{y}^2 \quad \mathbf{y}^3 \quad \mathbf{y}^4 \quad \mathbf{y}^5]$ is combination outputs of the ELM^k in response to **z**, **s** and $\bar{\mathbf{y}}$ 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.,

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-\left\|\mathbf{a}_i^k \cdot \mathbf{x}_j + b_i^k\right\|^2\}, \quad b_i^k \in R \tag{13}$$

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-(\mathbf{a}_i^k \cdot \mathbf{x}_j + b_i^k)\}, b_i^k \in R \tag{14}$$

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-b_i^k \left|\mathbf{x}_j - \mathbf{a}_i^k\right|\}, \; b_i^k \in R^+ \tag{15}$$

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 ELM^k (individual agent) 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, ELM^k are considered as individual agents. Lastly, the Meta-ELM is a three layers neural network, but ELM^k 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 paper, there are two benchmark datasets (e.g. Satellite Image and Image Segmentation) which were used to exam the enactment of ELM-MAS. The datasets' specifications were detailed in Table I [35]. All experiments were run on MATLAB (ver.2010) using a private computer equipped with Intel(R) Core(TM) i7 2.9 GHz CPU and 8 G RAM.

Table I: Specification of Benchmark Datasets [35]

Dataset	# Attributes	# Classes	# Training	# Testing
Satellite Image	36	6	4,435	2,000
Image Segmentation	19	7	1,500	810

In the experiment, we referred to the suggestion of Liang [35] to fix the number of hidden neuron of each ELM^k (i.e., L) to 400 for Satellite Image and 180 for Image Segmentation. On the other hand, 2/3 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_1) via a validation process. For each type of the activation function of ELM-MAS, training and validation processes start by setting L_1 = 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 insubstantial 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.

Table II: Validation Accuracy Rates of Sigmoid Activation Function

L_1	Satellite Image		Image Segmentation	
	Validation Accuracy (%)	Time (s)	Validation Accuracy (%)	Time (s)
50	89.35	2.38	**94.52**	**0.29**
100	**89.53**	**2.40**	94.20	0.31
150	89.45	2.39	94.21	0.33
200	89.50	2.43	93.77	0.37
250	89.40	2.52	93.50	0.38

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

Activation Function	Satellite Image			Image Segmentation		
	Test Accuracy (%)	Time (s)	# Hidden Neurons	Test Accuracy (%)	Time (s)	# Hidden Neurons
RBF	88.71	7.51	150	94.70	0.56	100
Sigmoid	89.27	2.53	200	94.48	0.30	50
Gaussian	89.47	2.97	400	94.62	0.36	100
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 IV: Comparison with Other ELM networks

Activation Function	Algorithm	Satellite Image			Image Segmentation		
		Test Accuracy (%)	Time (s)	# Hidden Neurons	Test Accuracy (%)	Time (s)	# Hidden Neurons
RBF	ELM [35]	89.01	319.14	400	94.53	12.20	180
	ELM-MAS	88.71	7.51	150	94.70	0.56	100
	Ensemble-ELM [20]	89.28	343.85	400	91.23	72.08	180
Sigmoid	ELM [35]	88.93	302.48	400	94.88	10.00	180
	ELM-MAS	89.27	2.53	200	94.48	0.30	50
	Ensemble-ELM [20]	89.01	1591.76	400	94.79	74.17	180

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 ELM^k 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.

4.1. Circulation Water Systems (CWS)

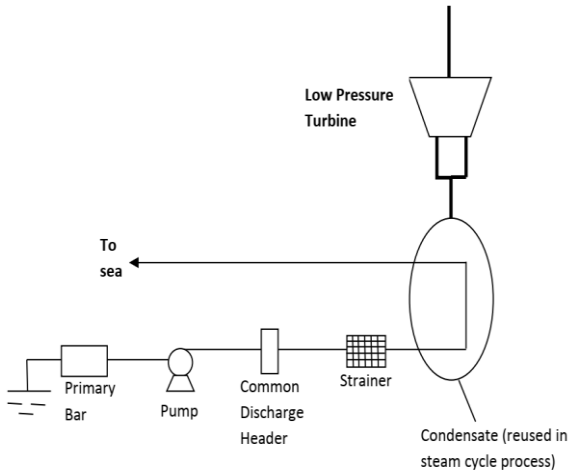


Fig. 4: Circulating Water Systems

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 tur bine 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 capability 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).

Table V: Description of Dataset in the CWS [38]

Class	Indication	# Training	# Validation	# Testing
1	Heat transfer in condenser is efficient and no blockage in piping system	219	109	219
2	Heat transfer in condenser is not efficient and no blockage in piping system	231	116	231
3	Heat transfer in condenser is efficient and significant blockage in piping system	271	136	272
4	Heat transfer in condenser is not efficient and significant blockage in piping system	279	139	278

Table VI: Experimental Results for Different Activation Functions in CWS

Activation Function	Circulating Water Systems		
	Test Accuracy (%)	Time (s)	# Hidden Neurons
RBF	96.37	0.13	50
Sigmoid	96.50	0.07	50
Gaussian	96.63	0.25	100
Laplace Act.	**96.96**	**0.26**	**100**
Laplace Basis	96.88	0.79	200

Table VII: Comparison with Different Approaches of CWS Dataset

Algorithm	Circulating Water System		
	Test Accuracy (%)	Time (s)	# Hidden Neurons
ELM-MAS	96.96	0.26	100
FAM [29]	95.70	-	18
SVM [30]	97.10	-	124

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, T_1, as well as a combustion chamber's time constant, T_2, in addition to a load-limiting feedback path. The load limit was susceptible to turbine exhaust temperature, whereby T_3 indicated the time constant of the exhaust gas measuring system. The constant, K_T, was utilized for adjusting the gain of the load-limited (A_T) feedback path.

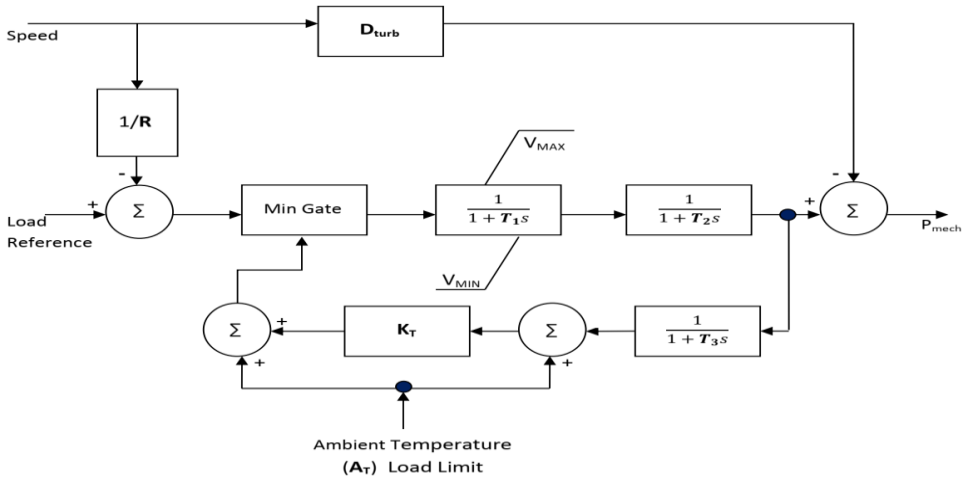


Fig. 5: Governor Model, GAST.

The training data is assembled on the output of the GAST block, which is the mechanical power, P_{mech} 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

Class	Range	# Training
1	$0 < R < 1$	100
2	$0 < D_{turb} < 0.5$	50
3	$0 < A_T \leq 1$	100
4	$0.01 < T_1 < 0.5$	50
5	$0.01 < T_2 < 0.5$	50
6	$0.01 < T_3 < 5$	140
7	$0 < K_T < 5$	140
	TOTAL	630

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

Class	# Training	# Validation	# Testing	# Training
1	40	20	40	100
2	20	10	20	50
3	40	20	40	100
4	20	10	20	50
5	20	10	20	50
6	56	28	56	140
7	56	28	56	140
			TOTAL	630

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

Activation Function	GAST		
	Test Accuracy (%)	Time (s)	# Hidden Neurons
RBF	74.31	0.09	70
Sigmoid	67.68	0.09	104
Gaussian	69.21	0.10	101
Laplace Act.	72.74	0.10	94
Laplace Basis	**76.79**	**0.10**	**65**

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 (RBF) 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 J510050684.

References

[1] Huang GB, Zhu QY & Siew CK (2004), Extreme Learning Machine: a new learning scheme of feedforward neural networks, in: *Proceedings of the IEEE International Joint Conference on Neural Networks* (IJCNN2004), vol. 2, Budapest, Hungary, July 25-29, pp.985–990.

[2] Huang GB, Zhu QY, Mao K, Siew CK, Saratchandran P, & Sundararajan N (2006), Can threshold networks be trained directly?, *IEEE Trans. Circuits Syst. II*, 53 (3), pp.187–191.

[3] Huang GB, Zhu QY, & Siew CK (2006), Extreme Learning Machine: theory and applications, *Neurocomputing*, 70 (1), pp.489–501.

[4] Huang GB, Zhou H, Ding X, & Zhang R (2012), Extreme Learning Machine for regression and multiclass classification, *IEEE Trans. Systems, Man, and Cybernetics*, Part B: Cybernetics, 42 (2), pp.513–529.

[5] Huang GB, & Chen L (2007), Convex incremental Extreme Learning Machine, *Neurocomputing*, 70 (168), pp.3056–3062.

[6] Huang GB, & Chen L (2008), Enhanced random search based incremental Extreme Learning Machine, *Neurocomputing*, 71 (16), pp.3460–3468.

[7] Yap KS, & Yap HJ (2012), Daily Maximum Load Forecasting of Consecutive National Holidays using OSELM-Based Multi-Agents System with Average Strategy, *Neurocomputing*, 81, pp.108-112.

[8] Huang G, Huang GB, Song S, & You K (2015), Trends in extreme learning machine: A review. *Neural Networks*, 61, pp.32-48.

[9] Liu X, Lin S, Fang J, & Xu Z (2015), Is extreme learning machine feasible? A theoretical assessment (part I). *IEEE Transactions on Neural Networks and Learning Systems*, 26 (1), pp.7-20.

[10] Zhou Y, Peng J, & Chen CP (2015), Extreme learning machine with composite kernels for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6), pp.2351-2360.

[11] You ZH, Lei YK, Zhu L, Xia J, & Wang B (2013), Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. *BMC Bioinformatics*, 14(8), p.S10.

[12] Song Y, Crowcroft J, & Zhang J (2012), Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine. *Journal of neuroscience methods*, 210 (2), pp.132–146.

[13] Yang Y, Wang Y, & Yuan X (2012), Bidirectional extreme learning machine for regression problem and its learning effectiveness. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9), pp.1498– 1505.

[14] Yan Z, & Wang J (2014), Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(3), pp.457–469.

[15] Zhang Y, & Zhang P (2011), Optimization of nonlinear process based on sequential extreme learning machine. *Chemical Engineering Science*, 66(20), pp.4702–4710.

[16] Nizar A, Dong Z, & Wang Y (2008), Power utility nontechnical loss analysis with extreme learning machine method. *IEEE Transactions on Power Systems*, 23(3), pp.946–955.

[17] Minhas R, Mohammed AA, & Wu QMJ (2012), Incremental learning in human action recognition based on snippets. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(11), pp.1529– 1541.

[18] Zhao G, Shen Z, Miao C, & Gay R (2008), Enhanced Extreme Learning Machine with stacked generalization. in: *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp.1191–1198.

[19] Sun ZL, Choi TM, Au KF, & Yu Y (2008), Sales forecasting using Extreme Learning Machine with applications in fashion retailing. *Decision Support Systems*, 46 (1), pp.411–419.

[20] Lan Y, Soh YC, & Huang GB (2009), Ensemble of online sequential Extreme Learning Machine. *Neurocomputing*, 72 (135), pp.3391–3395.

[21] Heeswijk MV, Miche Y, Lindh-Knuutila T, Hilbers P, Honkela T, Oja E, & Lendasse A (2009), Adaptive ensemble models of extreme learning machines for time series prediction. in: *Proceedings of the 19th International Conference on Artificial Neural Networks*: Part II, Vol. 5769 of ICANN'09, Springer-Verlag, Berlin, Heidelberg, pp.305–314.

[22] Heeswijk MV, Miche Y, Oja E, & Lendasse A (2011), GPU-accelerated and parallelized ELM ensembles for large-scale regression. *Neurocomputing*, 74 (16), pp.2430–2437.

[23] Quteishet A, Lim CP, Tweedale J, & CJain L (2009), A Neural Network-based Multi-agent Classifier System, *Neurocomputing*, 72, pp.1639-1647.

[24] Ossowski S, Fernandez A, Serrano JM, Hernandez JZ, Garcia-Serrano AM, Perez-de-la-Cruz JL, Belmonte MV, & Maseda JM (2004), Designing Multi agent Decision Support System the Case of Transportation Management. in: *The 3rd International Joint Conference on Autonomous Agents and Multi agent Systems*, vol.3 pp.1470-1471.

[25] Tolk A (2005), An Agent-Based Decision Support System Architecture for the Military Domain. in: *G. Phillips-Wren and L. Jain, (Eds.) Intelligent Decision Support Systems in Agent-Mediated Environments*, IOS Press.

[26] Hudson DL, & Cohen ME (2002), Use of Intelligent Agents in the Diagnosis of Cardiac Disorders. *Computers in Cardiology*, pp.633-636.

[27] Ossowski S, Hernandez JZ, Iglesias CA, & Ferndndez A (2002), Engineering Agent Systems for Decision Support. in: *The 3rd International Workshop Engineering Societies in the Agents World*, pp.184-198.

[28] Gwebu K, Wang J, & Troutt MD (2005), Constructing a Multi-Agent System: An Architecture for a Virtual Marketplace. in*: G. Phillips-Wren and L. Jain, (Eds.) Intelligent Decision Support Systems in Agent-Mediated Environments*, IOS Press.

[29] Singh R, Salam A, & Lyer L (2003), Using Agents and XML for Knowledge Representation and Exchange: An Intelligent Distributed Decision Support Architecture. *in: The 9th Americans Conference on Information Systems*, pp.1854-1863.

[30] Cao JW, Lin ZP, Huang GB, & Liu N (2012), Voting based Extreme Learning Machine, *Information Sciences*, 185, pp.66–77.

[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.
[32] Chan PK, & Stolfo SJ (1993), Experiments on multistrategy learning by metalearning. in: *Proceedings of the Second International Conference on Information and Knowledge Management*, pp.314–323.
[33] Guo Y, R¨uger SM, Sutiwaraphun J, & Forbes-Millott J (1997), Meta-learning for parallel data mining. in: *Proceedings of the Seventh Parallel Computing Workshop*, pp.1-2.
[34] Liao SZ, & Feng C (2014), Meta-ELM: ELM with ELM Hidden Nodes. *Neucomputing*, 128, pp.81-87.
[35] Liang NY, Huang GB, Saratchandran P, & Sundararajan N (2006), A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *Neural Networks*, 17(6), pp.1411-1423.
[36] Quteishat AMA, & Lim CP (2008), A Modified Fuzzy Min-max Neural Network with Rule Extraction And Its Application To Fault Detection and Classification. *Journal of Applied Soft Computing*, 8(2), pp.985-995.
[37] Tan SC, & Lim CP (2004), Application of An Adaptive Neural Network With Symbolic Rule Extraction To Fault Detection And Diagnosis An A Power Generation Plant. *IEEE Trans. Energy Conversion*, 19(2), pp.369–377.
[38] Abdul Aziz, NLA, Yap KS, Afif Bunyamin M (2013), A hybrid fuzzy logic and extreme learning machine for improving efficiency of circulating water systems in power generation plant. *IOP Conf. Series: Earth and Environmental Science*, 16(1), p.012102.
[39] Tan SC, Lim CP, & Rao MVC (2007), A Hybrid Neural Network Model for Rule Generation and its Application to Process Fault Detection and Diagnosis. *Engineering Applications of Artificial Intelligence*, 20, pp.203–213.
[40] Yap KS, Lim CP, & Au MT (2011), Improved GART Neural Network Model for Pattern Classification and Rule Extraction with Application to Power Systems. *IEEE Trans. Neural Networks*, 22(12), pp.2310-2323.
[41] Wong KP (1993), Artificial Intelligence and Neural Network Applications in Power Systems. *IEEE 2nd International Conference on Advances in Power System Control, Operation and Management*, Hong Kong, pp.37-46.
[42] Large-scale blackout in Malaysia (1996*). in Kaigai Denryoku (Foreign Power): Japan Electric Power Information Center, Inc*., pp.103–104.
[43] Inoue T, Sudo Y, Takeuchi A, Mitani Y, & Nakachi Y (1999), Development of a combined cycle plant model for power system dynamic simulation studies. *Trans. Inst. Elect. Eng. Jpn*., 119-B, (7), pp.788–797.
[44] Suzaki S, Kawata K, Sekoguchi M, & Goto M (2000), Combined cycle plant model for power system dynamic simulation study. *Trans. Inst. Elect. Eng. Jpn.*, 120-B (8/9), pp.1146–1152.
[45] Kunitomi K, Kurita A, Okamoto H, Tada Y, Ihara S, Pourbeik P, Price WW, Leirbukt AB, & Sanchez-Gasca JJ (2001), Modeling frequency dependency of gas turbine output. *Proc. IEEE/Power Eng. Soc. Winter Meeting*, vol.2, pp.678-683.
[46] Rowen WI (1983), Simplified mathematical representations of heavy-duty gas turbines. *Trans. Amer. Soc. Mech. Eng*., 105, pp.865–869.
[47] de Mello FP, & Ahner DJ (1994), Dynamic models for combined cycle plants in power system studies. *IEEE Trans. Power Syst.*, 9, pp.1698–1708.
[48] Yaw CT, Namas Khan NR, Ungku Amirulddin UA, Hashim AH, & Harun MN (2009), Development of Gas Turbines Model in MATLAB to Investigate Response in the Event of Major System Contingencies. *PowerTech* 2009, Bucharest, Romania.
[49] PSS/E™ 26, Program Application Guide: Volume II, December 1998, 1990-1998 Power Technologies, Inc.