IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 30, 2020, accepted July 11, 2020, date of publication July 30, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012983

Enhancement of Neural Network Based Multi Agent System for Classification and Regression in Energy System

CHONG TAK YAW[1], KEEM SIAH YAP[2], SHEN YUONG WONG[3], (Senior Member, IEEE), HWA JEN YAP[4], AND JOHNNY KOH SIEW PAW[2]

[1]Institute of Sustainable Energy, Universiti Tenaga Nasional, Kajang 43000, Malaysia
[2]Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Kajang 43000, Malaysia
[3]Department of Electrical and Electronics Engineering, Xiamen University Malaysia, Sepang 43900, Malaysia
[4]Department of Mechanical Engineering, University of Malaya, Kuala Lumpur 50600, Malaysia

Corresponding authors: Keem Siah Yap (yapkeem@uniten.edu.my) and Shen Yuong Wong (shenyuong.wong@xmu.edu.my)

This work was supported in part by the Universiti Tenaga Nasional under Grant 10436494/B/2019099, and in part by the Xiamen University Malaysia Research Fund under Grant XMUMRF/2018-C2/IECE/0001.

ABSTRACT Extreme Learning Machine improved the iterative procedures of adjusting weights by randomly selecting hidden neurons besides analytically determining the output weights. In this paper, the basic ELM neural network was enhanced with a simplified network structure to achieve regression performance. Next, to solve the pattern classification, a hybrid system was proposed which integrated the ELM neural network and MAS models. A MAS model is then designed with a novel trust measurement method to combine ELM neural networks. Firstly, ELM hybrid with Single Input Rule Module (SIRM-ELM) was designed. There was only a single input connected to the rules, where the rules were the hidden neurons of ELM and each represented a single input fuzzy rules. Results showed that the SIRM-ELM model was better than Support Vector Machine and traditional ELM. Secondly, an extreme learning machine based multi agent systems (ELM-MAS) was designed to improve ELM's capability. Its first layer was made up of at least one ELM where ELM acted as an individual agent, whereas another layer was made up of a single ELM acting as the parent agent. Lastly, Certified Belief in Strength (CBS) method was applied to the ELM neural network to form ELM-MAS-CBS, using the reputation and strength of individual agents as the trust measurement. The assembly of strong elements related to the ELM agents formed the trust management that allowed the improvement of the performance in MAS using the CBS method. Both of the developed models were evaluated on its application on the power generation system. The test accuracy rate of both models for circulating water systems was shown to be comparable to other algorithms. In short, the developed models had been verified using benchmark datasets and applied in power generation, where the results were satisfactory.

INDEX TERMS Certified belief in strength, extreme learning machine, multi agent system, single input rule module, power generation system.

I. INTRODUCTION

Feedforward Neural Networks (FNNs) is the most common approach for Artificial Neural Networks (ANNs) which is being used to recognize patterns. They are capable of handling non-linear as well as noisy data (e.g. data gathered from actual environments). Unfortunately, the downside of FNNs is its learning speed, due to:

(i) The slow error back-propagation (BP) and other gradient-based learning algorithms [1]–[3] being employed to train the neural networks, and
(ii) Its parameters are adjusted repeatedly using gradient-based learning algorithms.

In terms of pattern recognition, Multilayer Perceptron (MLP) [4], [5] and Radial Basis Function (RBF)

The associate editor coordinating the review of this manuscript and approving it for publication was Junxiu Liu.

networks [4], [6] are both superior. The MLP network was made up of a non-linear transformation of combined sigmoid functions of hidden neurons and can be used to recognize patterns. In the case of the RBF network, it solved problems by combining non-linear semi-parametric functions, for instance, the Gaussian kernel function. Nevertheless, trial-and-error or a set of cross validation is needed to predefine the number of kernel functions of RBF and hidden neurons of MLP, which might very well be a lengthy process. The training process was also lengthened due to the need to train the datasets so that it was compatible with the network.

Based on literature, RBF-based sequential learning model is fast becoming favored in terms of pattern recognition. Examples included the Resource Allocation Network (RAN) [7], Growing and Pruning RBF (GAPRBF) [8], Minimum Resource Allocation Network (MRAN) [9], Generalized Growing and Pruning RBF (GGAPRBF) [10] and Resource Allocation Network with Extended Kalman Filter (RANEKF) [11].

There has been much work being done to improve BP algorithms to eschew the local minima according to the better selection of activation function, dynamic variation of momentum and learning, and cost function. Simple Adaptive Momentum (SAM) can improve the convergence rate of BP [12]. The momentum coefficient was scaled giving to the likenesses shared between the changes in weights for the previous and current iterations, with lower computational overheads relative to conventional BP. Mitchell et. al. adjusted the momentum coefficient differently by accounting for the weights in the Multi-layer Perceptrons (MLP) in 2008. Such an approach proved to be superior to that of SAM. In 2011, Gradient Descent BP (GDAM) was proposed to increase its overall efficiency [13].

Despite the reported improved versions of the BP approaches, having to repetitively and iteratively adjust weights during training to accurately model a specific learning task of the training samples remains a challenge. In order to circumvent this [14]–[16], Extreme Learning Machine (ELM) is a new learning algorithm which was proposed by Huang *et al.* (2006a). In ELM, a single hidden layer feedforward neural network (SLFN) improved the iterative procedures of adjusting weights by randomly selecting hidden neurons in addition to calculating the output weights of SLFNs. Theoretically, ELM reported an excellent generalization performance at exceptionally quick learning speeds.

Most users preferred that the input samples were accurately classified. The output included an estimate of the classification strength. The possibility that the prediction was accurate compels the system that they can rely on it and make informed decisions [17]–[19].

Intelligent agents are regarded as a computer intelligence paradigm. Multi-Agent Systems (MAS) are widely applicable, namely in decision support [20], navigation [21], industrial steel processing [22], and power systems [23]. In this paper, MAS was studied as an approach that can be used to ensemble ELM networks, where each ELM acted as individual agent, and MAS structure merged with the entire predictions of ELM to create a classification system that performed well.

In MAS, a number of models can be used to describe agent links, such as Rasmussen, Pejtersen, & Goodstein (1994)'s decision ladder model and Bratman (1987)'s Belief, Desire, and Intention (BDI) model. In this paper, two models which are Haider, Tweedale, Urlings, and Jain (2006)'s Trust, Negotiation, and Communication (TNC), and Tweedale and Cutler (2006) were examined to develop a MAS model. The primary element in the TNC model was trust measurement. Due to the fact that trust was subjective, we need to investigate methods that can be used for trust computations to render it an objective quantity.

The main objective was to design, develop, and enhance ELM based neural network models that capitalized the advantage of ELM while avoiding their inherent limitation. The sub-objectives of this work included:

(Case 1) Enhancement of the existing ELM neural network for achieving regression performance with simplified network structures;

(Case 2) Propose of a hybrid system integrating the ELM neural network and MAS models for solving pattern classification; and,

(Case 3) Design of a MAS model with a novel trust measurement method to combine ELM neural networks.

A standard ELM is presented in this paper which hybrid with a simplified network structure and MAS model. These proposed models are the novel technique to aim to achieve better results.

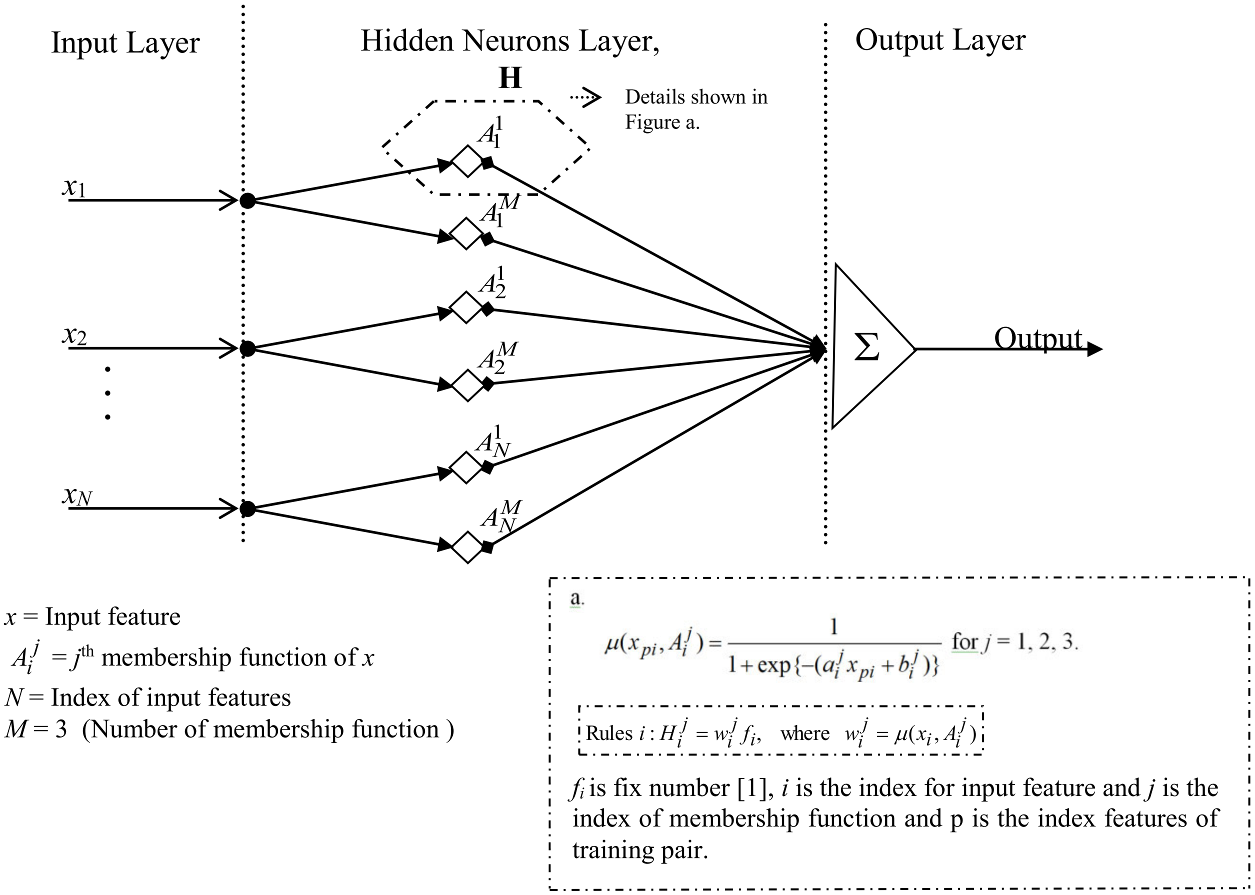
A. PROPOSED WORK

To achieve the main objective, the existing ELM neural network was enhanced with a simplified network structure to achieve regression performance. Next, to solve the pattern classification, a hybrid system was proposed which integrated the ELM neural network and MAS models. A MAS model is then designed with a novel trust measurement method to combine ELM neural networks. Each case is tested with application to testing its capability.

II. METHODS

A. (CASE 1) EXTREME LEARNING MACHINE WITH SINGLE INPUT RULE MODULE (SIRM-ELM)

To assess ELM, this section details the proposal of a novel unprecedented technique in ELM ideology for regression problems, which was ELM-based model using ELM hybrid with Single Input Rule Module (SIRM), denoted as SIRM-ELM. In SIRM-ELM, there was only a single input that connected to rules, where each of the hidden neurons of ELM represented a single input fuzzy rule. Hence, the number of hidden neuron of ELM determines the number of fuzzy rules.


Input Layer
Hidden Neurons Layer, H
Output Layer
Details shown in Figure a.
x_1
x_2
x_N
A_1^1
A_1^M
A_2^1
A_2^M
A_N^1
A_N^M
Σ
Output
x = Input feature
A_i^j = j^{th} membership function of x
N = Index of input features
$M = 3$ (Number of membership function)
a.
$\mu(x_{pi}, A_i^j) = \frac{1}{1+\exp\{-(a_i^j x_{pi} + b_i^j)\}}$ for $j = 1, 2, 3.$
Rules $i : H_i^j = w_i^j f_i$, where $w_i^j = \mu(x_i, A_i^j)$
f_i is fix number [1], i is the index for input feature and j is the index of membership function and p is the index features of training pair.

respectively. Nevertheless, the major dilemma was that the numbers of the fuzzy rules were kept increasing until the

FIGURE 1. Overview of SIRM-ELM; (a) General of SIRM-ELM Model; (b) General details for each hidden neurons in Figure 1(a).

Conventionally, when the "if-then" rules of the fuzzy inference method were used all the input and output items were assigned to antecedent and consequent parts respectively. Nevertheless, the major dilemma was that the numbers of the fuzzy rules were kept increasing until the system and arrangement of the rules became complicated [24]. Therefore, Yubazaki [25]–[31] developed an enhanced SIRM connected type fuzzy inference method that consociates the fuzzy rules module outputs significantly. The areas that SIRM method was applied to include the control of anti-swing and positioning for the overhead traveling crane [25], the control to stabilize inverted pendulum systems [27]–[29], the control of the 1st and 2nd order lag system with dead time [26], [31], non-linear function identification [31], and others, of which decent results were acquired [24].

The assumption is that a system consists of n input source and one output source. However, the system can also be extended with plural output sources. This is the basic, with n input source for SIRM:

$$\begin{aligned} SIRM-1: &\left\{R_1^j : if\ x_1 = A_1^j\ then\ \Delta_{u_1} = C_1^j\right\}_{j=1}^{m_i}, \\ &\quad\cdots \quad \cdots \\ SIRM-i: &\left\{R_i^j : if\ x_i = A_i^j\ then\ \Delta_{u_i} = C_i^j\right\}_{j=1}^{m_i}, \\ &\quad\cdots \quad \cdots \\ SIRM-n: &\left\{R_n^j : if\ x_n = A_n^j\ then\ \Delta_{u_n} = C_n^j\right\}_{j=1}^{m_n} \end{aligned} \tag{1}$$

In Equation (1), each SIRM independently corresponded to n input sources. The SIRM-i, where the i refers to i^{th} input source, R_i^j is the j^{th} rule in the SIRM-i, x_i refers to the i^{th} input source variable in the preceding section, and Δu_i is the variable in the following part of the SIRM-i. A_i^j and C_i^j are the membership functions of the x_i whereas Δu_i is the j^{th} rule in the SIRM-i. Additionally, $i = 1, 2, \ldots,$ n is the index number of the SIRM whereby $j = 1, 2, \ldots, m_n$ is the index number of the rules in the SIRM-i.

Fig. 1 showed the structure of SIRM-ELM, with the steps to train the data as shown below. Refer to Fig. 1 for the details definition of variables and parameters.

Step 1: Haphazardly set the input weights a_i^j, as well as bias, b_i^j (for $i = 1, 2 \ldots, N$ whereas for $j = 1, 2, 3$) of hidden neurons. Take into account that a_i^j and b_i^j are parameters of membership function for SIRM, A_i^j. The weights are generated based on $\alpha D - \omega$, where D is uniform distribution function that randomly generates a number between 0 to 1, α and ω are the parameters. By default, $\alpha = 2, \omega = 1$. As the results, the a_i^j and b_i^j are in the range of -1 to $+1$.

Step 2: For the training pair $(\mathbf{x}_{pi}, t_p)$ where $\mathbf{x}_{pi}$ is i^{th} feature of p^{th} training pair and t_p is target output (for $p = 1, 2, \ldots, P$). Determine the hidden layer output matrix $\mathbf{H}$ using the membership function $\mu(x_{pi}, A_i^j)$. For simplicity, the membership function can be denoted as μ_{pi}^j

$$\mu(x_{pi}, a_i^j, b_i^j) = \frac{1}{1 + \exp\{-(a_i^j x_{pi} + b_i^j)\}} \tag{2}$$

$$\mathbf{H} = \begin{bmatrix} \mu_{11}^1 & \mu_{11}^2 & \mu_{11}^3 & \mu_{12}^1 & \mu_{12}^2 & \cdots & \mu_{1N}^1 & \mu_{1N}^2 & \mu_{1N}^3 \\ \mu_{21}^1 & \mu_{21}^2 & \mu_{21}^3 & \mu_{22}^1 & \mu_{22}^2 & \cdots & \mu_{2N}^1 & \mu_{2N}^2 & \mu_{2N}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{P1}^1 & \mu_{P1}^2 & \mu_{P1}^3 & \mu_{P2}^1 & \mu_{P2}^2 & \cdots & \mu_{PN}^1 & \mu_{PN}^2 & \mu_{PN}^3 \end{bmatrix}_{P \times 3N} \tag{3}$$

Step 3: The output weights, β, were computed. Since it is a high possibility that $\mathbf{H}$ is a non-symmetrical matrix, the inverse matrix cannot be resolved. To circumvent this problem, a Moore-Penrose pseudo inverse matrix method is utilized, hence work out the output weights of β by the formula below,

$$\beta = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{T}, \tag{4}$$

where $\mathbf{T}$ is target output matrix, i.e.,$\mathbf{T} = [t_1 \ \ t_2 \ \ \ldots \ \ t_N]^T$

Step 4: After the output weights of SIRM-ELM were calculated, prediction of a set of new and unlabeled samples $\mathbf{z}$ can be computed, i.e., λ is the membership function, $\mathbf{h}$ is the hidden layer whereby $\mathbf{y}$ is the prediction output.

$$\lambda(z_{qi}, a_i^j, b_i^j) = \frac{1}{1 + \exp\{-(a_i^j z_{qi} + b_i^j)\}} \tag{5}$$

$$\mathbf{h} = \begin{bmatrix} \lambda_{11}^1 & \lambda_{11}^2 & \lambda_{11}^3 & \lambda_{12}^1 & \lambda_{12}^2 & \cdots & \lambda_{1N}^1 & \lambda_{1N}^2 & \lambda_{1N}^3 \\ \lambda_{21}^1 & \lambda_{21}^2 & \lambda_{21}^3 & \lambda_{22}^1 & \lambda_{22}^2 & \cdots & \lambda_{2N}^1 & \lambda_{2N}^2 & \lambda_{2N}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda_{Q1}^1 & \lambda_{Q1}^2 & \lambda_{Q1}^3 & \lambda_{Q2}^1 & \lambda_{Q2}^2 & \cdots & \lambda_{QN}^1 & \lambda_{QN}^2 & \lambda_{QN}^3 \end{bmatrix}_{Q \times 3N} \tag{6}$$

$$\mathbf{y} = \mathbf{h}\beta \tag{7}$$

where $q = 1, 2, \ldots.Q$ and Q is the number of test samples.

Step 5: After compute the output of ELM for testing samples, determine the root mean squared error (RMSE), i.e.,

$$RMSE_{test} = \sqrt{\frac{\sum_{q=1}^{Q} (y_q - d_q)^2}{Q}} \tag{8}$$

where y_q and d_q were prediction and actual output respective to $\mathbf{z}_q$.

The capability of SIRM-ELM was applied to the NO_x emission of power generation plant.

1) REAL-WORLD APPLICATION: NO_X EMISSION OF POWER GENERATION PLANT

Nitrogen occurred naturally in the atmosphere as an inactive gas. In addition, our atmosphere contains just about 78% N_2 by volume in the air. The NO_x was referring to nitrogen oxides but mostly include nitrogen monoxide, also identified as nitric oxide, NO as well as nitrogen dioxide, NO_2. There were also others in the family, including N_2O, N_2O_4 and N_2O_5.

The presence of atmospheric NO_x posed direct and indirect effects on human health and ecosystems, i.e. animals and plants, in the environment. NO_x reacted with components such as water, oxygen and other chemicals to form smog and acidic pollutants which leads to the formation of acid rain. In turn, acid rain, together with dry deposition and cloud, may cause damages and deterioration to cars and buildings.

NO_x is mainly released during the combustion process of fossil fuels like coal, oil and natural gas. According to European Environment Agency (EEA) technical report (1990 - 2013), 21% of the NO_x gas emissions in the European Union were from energy production and distribution, which was approximately 1,600 kilotonne. However, the growth of power generation industries was expected to be increasing by 18.7 gigawatts (GW) in the coming years, 2016 - 2018, due to the price and availability of natural gas. Hence, the prediction of NO_x emission is vital for the power generation sector and the issue should be taken seriously.

For real-world application in this study, the NO_x emission of an open cycle gas turbine in a power generation plant (located at Port Dickson, Malaysia) had been investigated [32]. The objective was the development of a neural network model to predict NO_x emission. There were 150 input attributes taken from the parameters of the power generation plant such as the loading of gas turbine, temperature, pressure, etc. The quantity of NO_x (in ppm) emitted from the gas turbine was the targeted output.

B. (CASE 2) HYBRIDIZING ELM-MAS (EXTREME LEARNING MACHINE AND MULTI AGENT SYSTEMS)

Extreme Learning Machine (ELM) has been well recognized as a more effective learning algorithm with better generalization and faster learning speed, in comparison to the conventional learning methods [33]–[39]. In addition, ELM is well-known for its capability to produce universal approximation using input weights and haphazard biases [40].

In essence, the link among the output and hidden layers are studied using primarily the input weights with optional hidden neurons.

ELM is tremendously effective and inclined towards global optimum in divergence to the CFNN (conventional feedforward neural network), according to Huang et al. [41], [42]. Moreover, ELM is capable to attain the utmost generalization bound of the CFNN, in which each parameter is learned with activation functions that are usually exploited [86]. In terms of efficiency and generalization, ELM has shown enhanced accomplishments compared to the traditional FNN [33]–[39]. Besides, ELM is applicable to other fields and not limited to chemical processes [43], hyperspectral images [44], action recognition [45], biomedical analysis [46], [47], power systems [48], system modeling [49], [50].

Currently, the focus of the research on ELM was to assimilate each independent prediction of some ELMs to create an optimum output using an ensemble model [51]–[55]. This approach has been adopted as evidenced in Multi Agent System (MAS) particularly [56]. MAS has been a center of attention in modern years as it has been effectively functional by researchers with extensive applications in different sectors including health care [57]–[59], e-Commerce [56], [60]–[62], military support [63], [64], knowledge management [65]–[68], decision support [69]–[72], as well as control systems [73]–[76]. Fig. 2 showed the common structure of MAS, in which the ground platform consists of a group of ELMs which are the individual agents. In general, the ultimate combination module, which consisted of the outcomes of ELM's individual agents delivered to the corresponding parent agent, formed the structure.

The common exercise for meta-learning is used to combine the outcomes of various learners. It is interpreted as knowledge is learned by at least a learner [77], [78]. The model was developed by several ELMs which act as the hidden neurons, and the outcomes of hidden neurons learned from meta-learner. Experimental results and theoretical analysis based on a number of studies using benchmark regression and artificial datasets which were trained by several ELMs, provide good performance at the expense of a lower computational rate [79]. The Meta-ELM [79] was a special design with ELM where ELMs as hidden neurons. Nonetheless, an ELM-MAS was designed from another perception. In this section, ELM-MAS had two layers of full ELMs: consisted of at least an ELMs where every ELM was reflected as an individual agent in the first layer; consisted of a single ELM and acted as the parent agent in the second layer. Therefore, this double layers' arrangement of the proposed ELM-MAS resembled a classic MAS as shown in Fig. 2.

As shown in Fig. 3, depending on the type of activation function it utilized, an ELM can either be a feedforward or RBF network with a sophisticated learning algorithm. A series of N training samples (with an individual target output vector as well as input vector), $\mathbf{t}_j \in \mathbf{R}^C$ (C is the number of classes) and $(\mathbf{x}_j, \mathbf{t}_j)$, i.e. $\mathbf{x}_j \in \mathbf{R}^M$ (M is the number of input attributes), consisting of L number of hidden neurons, were utilized to train an ELM. Five ELMs acted as individual agents in this case and had different random input weights respectively. The output of every ELM^k (for $k = 1, 2, \ldots, 5$) shown in Fig. 4, in response to $\mathbf{x}_j$ is

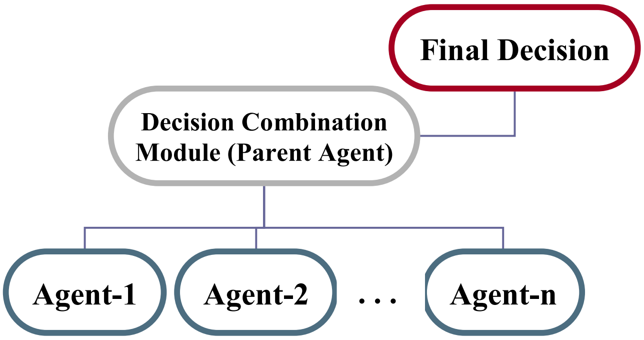


FIGURE 2. A general structure of MAS (Multi Agent System).

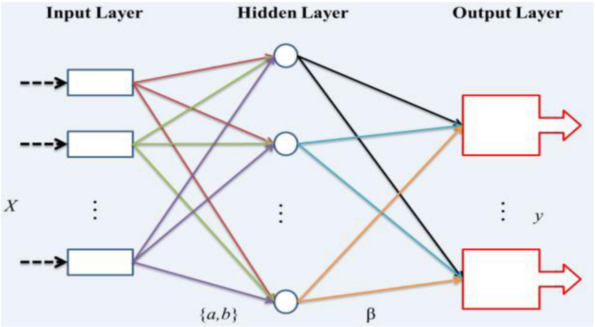


FIGURE 3. Architecture of an ELM (Extreme Learning Machine).

$$\mathrm{ELM}^k(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_{ic}^k G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \mathbf{t}_j \quad \text{for } j = 1, \ldots, N \quad \text{and for } c = 1, \ldots, C \tag{9}$$

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

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \frac{1}{1 + \exp\{-(\mathbf{a}_i^k \cdot \mathbf{x}_j + b_i^k)\}}, \quad b_i^k \in R \tag{10}$$

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-b_i^k \left\| \mathbf{x}_j - \mathbf{a}_i^k \right\|^2\}, \quad b_i^k \in R^+ \tag{11}$$

Equations (10) and (11) respectively revealed the definition of the $G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j)$ for additive sigmoid hidden neuron as well as RBF hidden neuron.

The training procedures were given as followed. Stage 1: Assigned the input weights $\mathbf{a}_i^k$ and b_i^k randomly for $k = 1, 2, \ldots, 5$ and $i = 1, \ldots, L$.

Stage 2: Computation of the hidden layer output matrix for ELM^k, $\mathbf{H}^{\mathbf{k}}$, as follows where $k = 1, 2, \ldots, 5$.

$$\mathbf{H}^k = \begin{bmatrix} G(\mathbf{a}_1^k, b_1^k, \mathbf{x}_1) & \ldots & G(\mathbf{a}_L^k, b_L^k, \mathbf{x}_1) \\ : & \ldots & : \\ G(\mathbf{a}_1^k, b_1^k, \mathbf{x}_N) & \ldots & G(\mathbf{a}_L^k, b_L^k, \mathbf{x}_N) \end{bmatrix}_{N \times L} \tag{12}$$

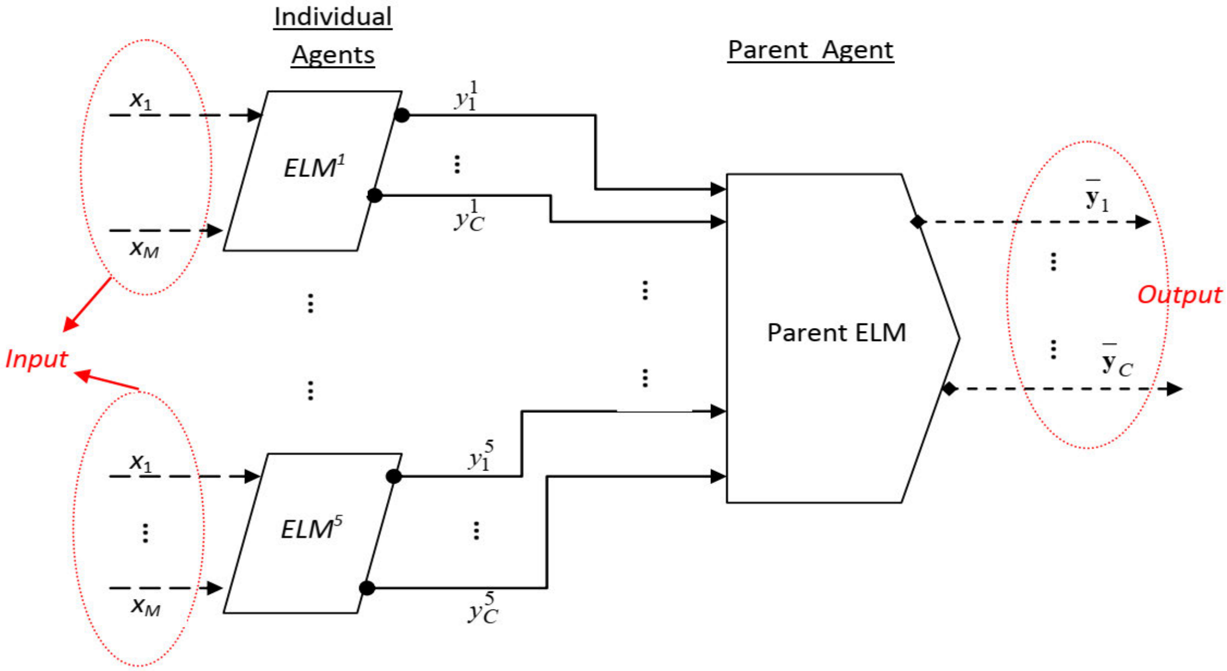


FIGURE 4. Design of ELM-MAS with parent agent and several individual agents.

Stage 3: Calculation of the β^k, output weights of ELM^k. As for the reason that $\mathbf{H}$ is probably a non-symmetrical matrix, the inverse matrix can't be solved. Therefore, a Moore-Penrose pseudo inverse matrix technique was embraced to evade this problematic, which was represented by the following calculation,

$$\beta^k = \left((\mathbf{H}^k)^T(\mathbf{H}^k)\right)^{-1}(\mathbf{H}^k)^T\mathbf{T}, \tag{13}$$

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

Stage 4: After the output weights of ELM^k were calculated, the outputs of ELM^k were computed using the training samples.

$$\mathbf{y}^k = \text{ELM}^k(\mathbf{x}_j) = \sum_{i=1}^{L}\beta_{ic}^k G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j)$$
$$\text{for } j = 1, \ldots, N \quad \text{and for } c = 1, \ldots, C, \tag{14}$$

Stage 5: Randomly assigned the input weights for parent ELM, i.e., q_i and $\mathbf{p}_i$ $(i = 1, \ldots, L_1)$, where L_1 is the number of hidden neuron of parent ELM.

Stage 6: Computation of the $\mathbf{S}$, hidden layer output matrix for parent ELM, shown as follow

$$\mathbf{S} = \begin{bmatrix} G(\mathbf{p}_1, q_1, \mathbf{w}_1) & \ldots & G(\mathbf{p}_{L_1}, q_{L_1}, \mathbf{w}_1) \\ \vdots & \ldots & \vdots \\ G(\mathbf{p}_1, q_1, \mathbf{w}_N) & \ldots & G(\mathbf{p}_{L_1}, q_{L_1}, \mathbf{w}_N) \end{bmatrix}_{N\times L_1} \tag{15}$$

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

Stage 7: Used the output of ELM^k to compute the output weights of parent ELM, α by the calculation beneath,

$$\alpha = \left((\mathbf{S})^T(\mathbf{S})\right)^{-1}(\mathbf{S})^T\mathbf{T}, \tag{16}$$

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

As soon as every sample were trained with Stage 1 until Stage 7, the ELM-MAS can be utilized for validation of an unknown $\mathbf{z}$, input vector based on the $\mathbf{a}^k$, $\mathbf{b}^k$, β^k, $\mathbf{p}$, $\mathbf{q}$ and α i.e.,

$$\mathbf{h}^k = [\,G(\mathbf{a}_1^k, b_1^k, \mathbf{z}) \quad \ldots \quad G(\mathbf{a}_L^k, b_L^k, \mathbf{z})\,]_{1\times L} \tag{17}$$
$$\mathbf{y}^k = \mathbf{h}^k\beta^k \tag{18}$$
$$\mathbf{s} = [\,G(\mathbf{p}_1, q_1, \mathbf{v}) \quad \ldots \quad G(\mathbf{p}_L, q_L, \mathbf{v})\,]_{1\times L_1} \tag{19}$$
$$\bar{\mathbf{y}} = \mathbf{s}\alpha \tag{20}$$

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

In addition to Equations (10) & (11), there were some activation functions that had been used in this case, i.e.,

$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-\left\|\mathbf{a}_i^k \cdot \mathbf{x}_j + b_i^k\right\|^2\}, \quad b_i^k \in R \tag{21}$$
$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-(\mathbf{a}_i^k \cdot \mathbf{x}_j + b_i^k)\}, \quad b_i^k \in R \tag{22}$$
$$G(\mathbf{a}_i^k, b_i^k, \mathbf{x}_j) = \exp\{-b_i^k\left|\mathbf{x}_j - \mathbf{a}_i^k\right|\}, \quad b_i^k \in R^+ \tag{23}$$

The proposed Meta-ELM [77] and ELM-MAS have similar comparable structure. Nevertheless, they had alterations

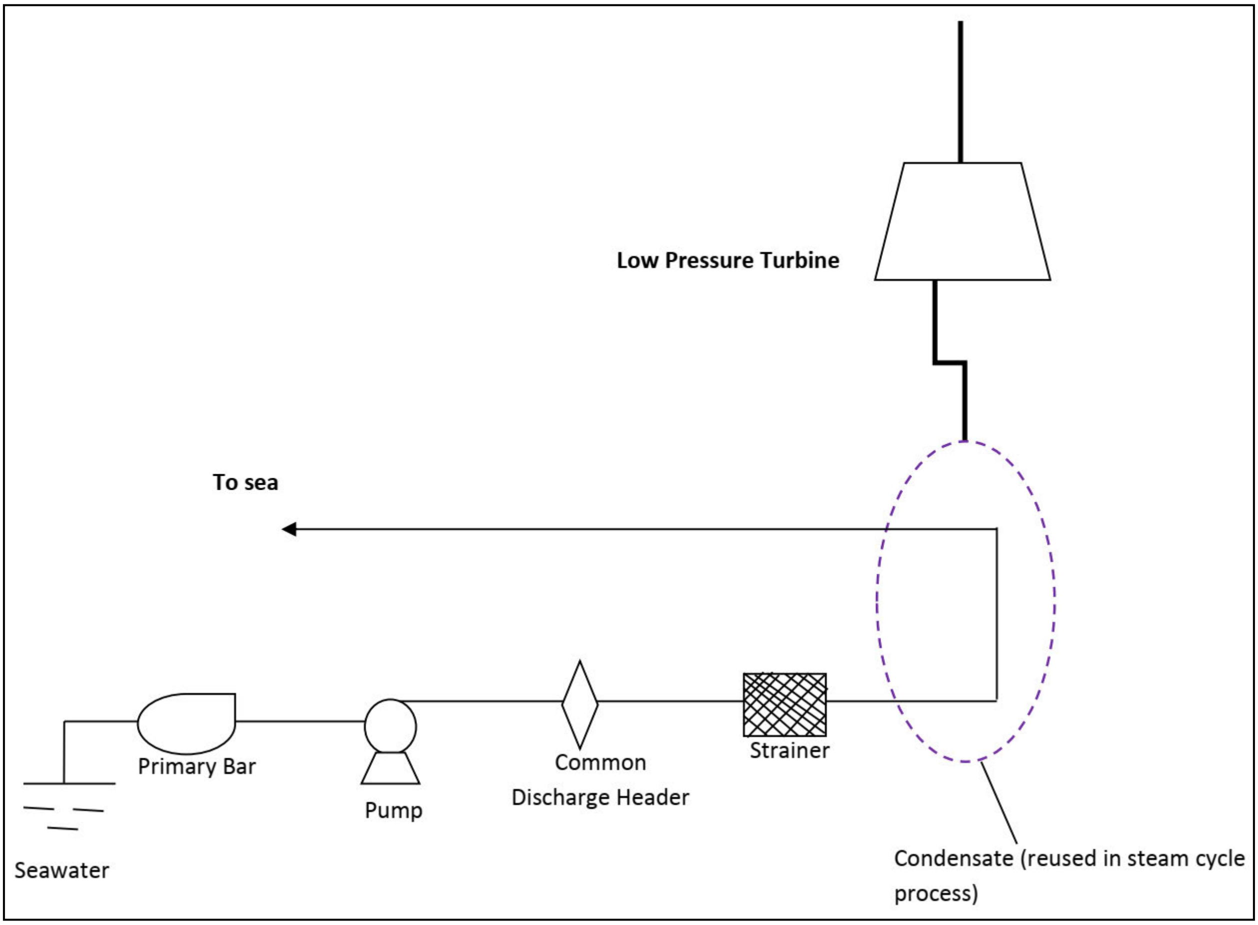


FIGURE 5. **The structure of CWS (Circulating Water System).**

in the way they handled training datasets, and also subtle distinctions in structural representation as shown below.

1) Meta-ELM partitioned the full training dataset into some random subsets and each ELM (hidden neurons) learned a subset. On the other hand, all ELM^k (individual agent) of the proposed ELM-MAS trained by the same full dataset (training).
2) The hidden neurons in Meta-ELM were ELMs. Whereas for ELM-MAS, the individual agents were ELM^k.
3) A three layers' neural network for the Meta-ELM. However, parent ELM and ELM^k of ELM-MAS had six layers of neural network (a complete three layers structure).

1) REAL-WORLD APPLICATION: NO$_X$ EMISSION OF POWER GENERATION PLANT

Fig. 5 revealed a CWS (Circulating Water System) in a power generation plant belonged to TNB (Tenaga Nasional Berhad) in Penang, Malaysia [80]–[82]. The system contained turbine condensers between the outfall, the sea water's inlet where the water was directed back in the sea, drum strainer and piping. In CWS, turbine condenser was the major component that instantaneously functioned in the elimination of heat from the LPS (Low-Pressure Steam) plus the conservation of the turbine backpressure at the lowest possible yet constant level.

2) REAL-WORLD APPLICATION: GAST GOVERNOR

The gas turbine monitoring and control were frequently introduced by the Energy Management Systems [83]. In addition, the energy control center was commonly utilized by these computer-based systems [84], [85]. During steady-state operation, gas turbine application software and other analysis software were being presented into the Energy Management Systems to examine and forecast the behavior of gas turbines [86]. Even though this software was an influential tool, it's capabilities to support the operating engineers in creating the finest judgments were restricted in the period when unexpected otherwise unplanned approaches of 2 tasks were discovered. The triggers of abnormal modes in the system operation were network faults, frequency deviations or either reactive and active power imbalances in most cases. Therefore, system shutdown (complete or partial) can be occurred in an unintentional task [86]. As a consequence, experienced operation engineers will be the one who is making the judgements for the restoration of the gas turbine

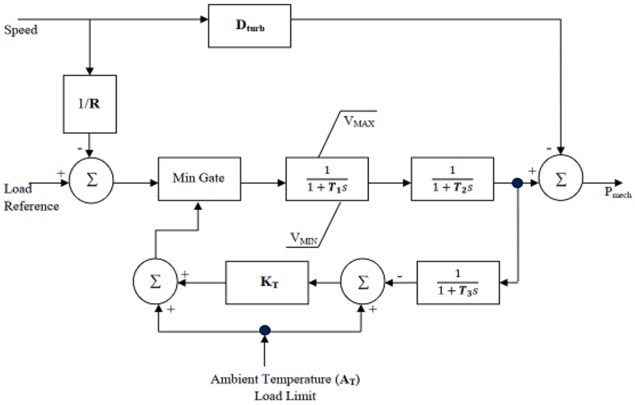


FIGURE 6. GAST, The governor model.

under these emergency situations. Therefore, the knowledge of experienced operation engineers as well as the conventional application software are both essential for balancing reactive and active power, efficiency in diagnosis of network faults, and network restoration [86]. Hence, developments of efficient and fast techniques of forecasting unusual system behavior are essential.

From the record, Malaysia has experienced numerous large-scale blackout occurrences for the past years [84], [85]. In 2005, a number of gas turbine plants were consecutively tripped out unconsciously and followed with a frequency fall of about 1.5 Hz which subsequently led to depletion in a total of 5760 MW. Therefore, some studies were conducted to witness how the combined cycle power plants react with the drops in frequency [87], [88]. These gas turbine models developed by Rowen [89] and Mello *et al.* [90] to replicate the real-world plants, which were then used to determine the reactions of the frequency variations. However, there has not been a detailed analysis to study the behavior of plant variables during frequency drops.

The vital dynamic structures of industrial gas turbines driving generators connected to electric power systems indicates by the governor model (GAST). Speed variances from nominal were planned to be minute (approximately five percent). Fig. 6 showed GAST, which contained of a combustion chamber's time constant, T_2, as well as a load-limiting feedback path, in addition a forward path with governor time constant, T_1. The parameter that adjusted the gain of the load-limited (A_T) feedback path is the constant, K_T. T_3 indicated the time constant of the exhaust gas measuring system. Lastly, the load limit was susceptible to turbine exhaust temperature.

C. (CASE 3) IMPROVED HYBRIDIZING ELM BASED MULTI AGENT SYSTEM USING CERTIFIED BELIEF IN STRENGTH (ELM-MAS-CBS)

This section presented an application of ELM in the MAS. This model is named ELM-MAS-CBS (Extreme Learning Machine in Multi Agent Systems) where Certified Belief in Strength (CBS) acted as a trust measurement technique. The strength and reputation of the ELM neural network (individual agents of MAS) are used by the CBS method to develop the trust measurement. With this method, strong elements that were linked to the individual agents (ELM) were gathered to develop the trust measurement to improve on MAS.

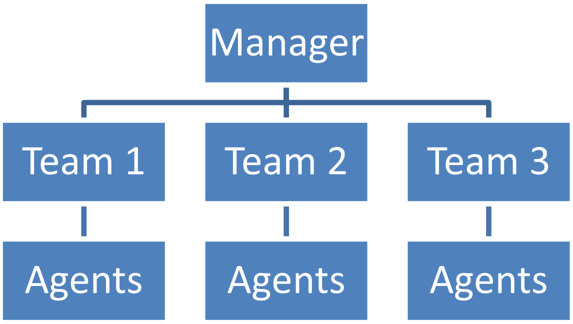


FIGURE 7. Summary of ELM-MAS-CBS model.

As the information in [53], rejection and recognition accuracy rates based trust measurement had been suggested. There were 2 groups utilized where the primary consisted of three modified FMM (Fuzzy min-max) agents whereby another group consisted of three modified Fuzzy ARTMAP (FAM) agents. Better performances were reported in the model as compared to other tactics stated in. On the other hand, there was one more trust measurement tactic suggested which based on Bayesian formalism with FMM MAS [91]. To attain the trust measurement, the FMM in the model was used as a learning agent in MAS and tailed by combination with Bayesian formalism. The results in proposed model showed improvement as compared to other tactics [91].

A technique called Certified Belief in Strength (CBS) is the latest development of MAS model for trust measurement, which was based on the reputation and strength of individual FMM based agents [91]. Consequently, trust was the strong element related to the FMM agents that enabled the CBS technique to increase the performance of the MAS in the training practice. The result showed that the improvement of the accuracy rates of the individual agents [91].

Therefore, an extended version of CBS method by using ELM (Extreme Learning Machine) based MAS (Multi Agent System) (from now designated as ELM-MAS-CBS). In Multi-Agent Classifier System for Certified Belief in Strength (MACS-CBS), it used FMM of several hyperboxes. In the proposed model, a ''team'' idea was employed with individual ELM-based agents.

The Fig. 7 is shown that the ELM-MAS-CBS model consisted of three levels. The bottom level contained a few individual agents (ELM-based agents); the middle level contained some teams of ELM-based. In addition, the new approach which is applied the CBS technique into the individual ELM-based agents. The final decision is on the last level and is selected by the Manager from the peak CBS team as the output. As for this section, the number of agents used in a team was set as 5 ($K = 5$), whereby the number of teams was set as 3 ($T = 3$). In addition, an ELM-based agent was

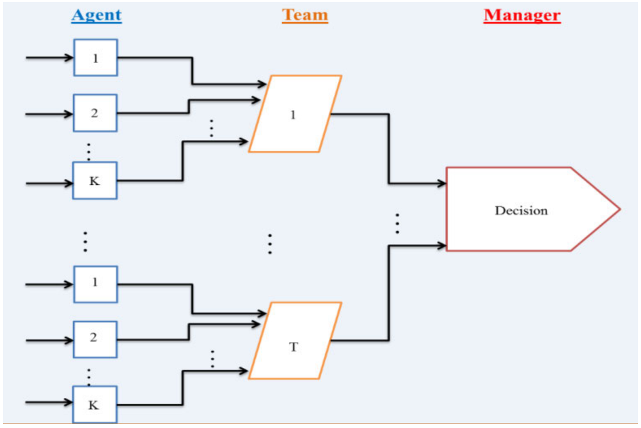


FIGURE 8. The design of of ELM-MAS-CBS model.

designated as ELM^{tk} (for $k = 1, \ldots, K$, for $t = 1, \ldots, T$). The design of ELM-MAS-CBS is shown in Fig. 8.

The stages of validation and training were as detailed below.

Stage 1: Randomly allocated the input weights $\mathbf{a}_i^{tk}$ and b_i^{tk}. In the training process, variables run for $i = 1, \ldots L$ (where L is the number of hidden neuron of ELM), for $k = 1, 2, \ldots, K$, and $t = 1, .., T$.

Stage 2: Calculated H^{tk}, the hidden layer output matrix for ELM^{tk} as follows, where $\mathbf{x}_j$ is the input vector, N is the number of training samples and G is the activation function.

$$G(\mathbf{a}_i^{tk}, b_i^{tk}, \mathbf{x}_j) = \exp\{-b_i^{tk}\left\|\mathbf{x}_j - \mathbf{a}_i^{tk}\right\|^2\} \quad \text{(RBF)} \tag{24}$$

$$G(\mathbf{a}_i^{tk}, b_i^{tk}, \mathbf{x}_j) = \frac{1}{1+\exp\{-(\mathbf{a}_i^{tk}\cdot\mathbf{x}_j + b_i^{tk})\}} \quad \text{(Sigmoid)} \tag{25}$$

$$\mathbf{H}^{tk} = \begin{bmatrix} G(\mathbf{a}_1^{tk}, b_1^{tk}, \mathbf{x}_1) & \ldots & G(\mathbf{a}_L^{tk}, b_L^{tk}, \mathbf{x}_1) \\ \vdots & \ldots & \vdots \\ G(\mathbf{a}_1^{tk}, b_1^{tk}, \mathbf{x}_N) & \ldots & G(\mathbf{a}_L^{tk}, b_L^{tk}, \mathbf{x}_N) \end{bmatrix}_{N\times L} \tag{26}$$

Stage 3: Compute β^{tk}, the output weights of ELM^{tk} by using the following equation,

$$\beta^{tk} = \left((\mathbf{H}^{tk})^T(\mathbf{H}^{tk})\right)^{-1}(\mathbf{H}^{tk})^T\mathbf{T}, \tag{27}$$

where the respective targeted output vectors, $\mathbf{T} = [t_1, \ldots, t_N]^T$.

Stage 4: Compute the outputs ELM^{tk}, i.e.,

$$\mathbf{y}^{tk} = ELM^{tk}(\mathbf{x}_j) = \sum_{i=1}^{L}\beta_i^{tk}G(\mathbf{a}_i^{tk}, b_i^{tk}, \mathbf{x}_j)$$
$$\text{where } j = 1, \ldots, N \tag{28}$$

Stage 5: After that calculate for accuracy rates of the ELM^{tk} as the following equation.

$$A^{tk} = \frac{N^{tk}}{N} \times 100\% \tag{29}$$

where A^{tk} and N^{tk} are accuracy rate and number correctly classified samples of ELM^{tk}.

Stage 6: Calculate the output of ELM^{tk} based on Equation (28) by using the validation samples.

Stage 7: Set an initial bid coefficient (C_{bid}) is 0.01 [91] and initial strength of CBS for all team is 100 (S = [100 100 100]) [91]. In addition, the strength was in proportion to initial team bid as follows [92],

$$B^t = C_{bid}S^t \tag{30}$$

Stage 8: Calculate the trust element, C^t by using the validation samples as shown in Equation (31). Determine C^k by using equation (29) in order to find the accuracy rate of the agents in each team. After that, the peak accuracy rate of ELM was selected (designated as ELM^{tw} where the winner of the team, w) and then indicating its team by inserting into Equation (31) and then surrender it to the top level which is the manager layer.

$$C^t = C_{bid}(S^t + A^{tw}) \tag{31}$$

Stage 9: Giving to a proposed paper [91], the Equation (30) act as the penalty and reward to revise the strength using the Equation (32), where R is reward and P is penalty. If an agent makes an incorrect prediction, $P = B^t$ while $R = 0$; otherwise $P = 0$ while $R = B^t$.

$$S^t(new) = S^t(now) - P + R \tag{32}$$

Stage 10: After S^t was revised, therefore both the B^t and the A^{tk} were also revised using the Equation (30) and (29), respectively.

The ELM-MAS-CBS can be used for prediction of a newly arrived and unknown input vector $\mathbf{z}$ after all the samples were trained using Stage 1 to Stage 10.

Stage 11: Load all the b_i^{tk}, $\mathbf{a}_i^{tk}$, A^{tk}, β^{tk}, S^t, and C^t from the completed training process in Stage 1 till Stage 10. The variables were ran for $k = 1, 2, \ldots, K$, for $i = 1, \ldots, L$, and $t = 1, .., T$ in all stages / equations of validation process.

Stage 12: Calculate h^{tk}, the hidden layer output matrix for ELM^{tk} as follows.

$$\mathbf{h}^{tk} = [\,G(\mathbf{a}_1^{tk}, b_1^{tk}, \mathbf{z}) \quad \ldots\; G(\mathbf{a}_L^{tk}, b_L^{tk}, \mathbf{z})\,]_{1\times L} \tag{33}$$

Stage 13: Compute the outputs of ELM^{tk},

$$\mathbf{y}^{tk} = \mathbf{h}^{tk}\beta^{tk} \tag{34}$$

Stage 14: The selection of the peak accuracy rates was from each team (designated as A^{tU}), and then compute the trust elements of teams using Equation (35).

$$C^t = C_{bid}(S^t + A^{tU}) \tag{35}$$

$$A^{tU} = \arg\max_k\left(A^{tk}\right) \tag{36}$$

Stage 15: Determine the peak of the C^t from all teams (designated as C^V), where the winner from all teams, V, i.e.,

$$C^V = \arg\max_t\left(C^t\right) \tag{37}$$

Stage 16: Using the Equation (34) to find the final output of ELM-MAS-CBS, where $k = U$ (winning agent of the winning team) and $t = V$ (winning team).

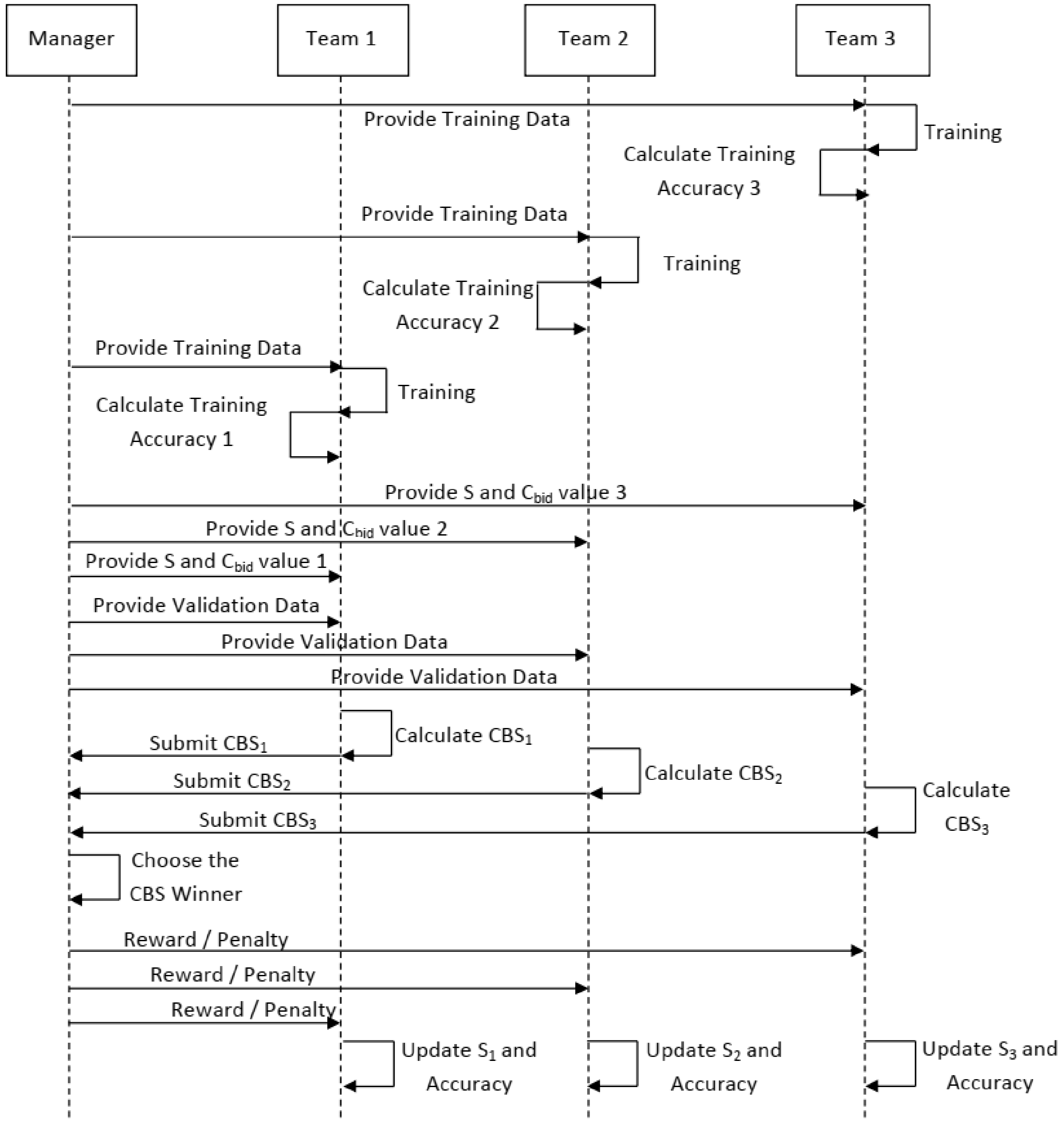


FIGURE 9. Arrangement of algorithm for ELM-MAS-CBS model.

Flowcharts were delineated in Fig. 9 to simplify the procedures taken by the training phase and the validation phase.

The capability of ELM-MAS-CBS was applied to the CWS (circulation water systems) and GAST governor for power generation.

1) REAL-WORLD APPLICATION: CIRCULATING WATER SYSTEMS

The Circulating Water datasets was explained in (Case 2) section. Despite the hybridization of ELM and MAS as described, this section explored the enhancement of ELM-MAS's capability in dealing with CWS dataset after the Certified Belief in Strength was applied on the ELM neural network, i.e. individual agents of MAS. This means that the trust measurement was achieved based on strength and reputation of every agent. To form the trust management, strong elements associated with the ELM agents were gathered which let the CBS enhanced the capability in MAS.

2) REAL-WORLD APPLICATION: GAST GOVERNOR

The explanation of GAST governor dataset had been used in the (Case 2, part 2) section.

III. RESULTS AND DISCUSSION

A. (CASE 1) EXTREME LEARNING MACHINE WITH SINGLE INPUT RULE MODULE (SIRM-ELM)

The applicability of the SIRM-ELM model was investigated. Four benchmark regression datasets were obtained from UCI machine repository, namely Abalone, Balloon, Strike and Space-ga, to utilize for performance evaluation of SIRM-ELM. Only addictive Sigmoid hidden neuron (SigAct) was utilized in the analysis. Table 1 showed the details of the computer and software specifications which was used to perform all analysis in this paper. The specifications of the datasets were shown in Table 2.

TABLE 1. Details of Personal Computer and Software Packages.

Items	*Specification*
Personal Computer	Asus
CPU	Intel(R) Core(TM) i7 2.5 GHz
Software	Matlab 7.11.0.584 (R2010b) (ver.2010)
Operating Systems	Windows 8.1
RAM	8 GB
Programming Language	Matlab Language

TABLE 2. Details of Benchmark Regression Datasets.

Datasets	*# Attributes*	*# Samples (Training)*	*# Samples (Testing)*	*# Total Samples*
Abalone	8	3000	1177	4177
Balloon	2	1334	667	2001
Strike	6	416	209	625
Space-ga	6	2071	1036	3107

In all experiments, four benchmark regression datasets with training and validation samples were calculated using the train-validation-test technique as suggested by literature [35].

The number of membership functions of an input attribute is tested for 1, 2 or 3, (i.e., $j = 1, 2, 3$) for all the regression datasets. In addition, the RMSE is based on default range for a_i^j and b_i^j for all rules (i.e., $i = 1, 2 \ldots, 3N$). Note that in SIRM-ELM, the number of fuzzy rule was equivalent to number of hidden neuron of ELM. For each dataset, the experiments were conducted for 50 times with random a_i^j and b_i^j and the mean results are documented.

The outcomes of the proposed SIRM-ELM were also compared to the results of other ELM-based methods. As seen from Table 3, the RMSE of SIRM-ELM is better when compare with OS-ELM [19], SVM [19] and ELM [1].

TABLE 3. RMSE of SIRM-ELM, ELM, SVM, and OS-ELM.

Algorithm	*Abalone*	*Balloon*	*Strike*	*Space-ga*
SIRM-ELM	**0.07598**	**0.04432**	0.2656	**0.03591**
OS-ELM [19]	0.0771	-	-	-
SVM [19]	0.0764	0.059	**0.2282**	0.0648
ELM [1]	0.0761	0.0553	0.2985	0.0624

1) REAL-WORLD APPLICATION: NO$_X$ EMISSION OF POWER GENERATION PLANT

A total of 3,405 data samples had been collected for training and testing of SIRM-ELM. Out of 3,405 data samples, 2,270 were trained while the balances of 1135 were tested (Table 4). An experiment is conducted on the testing datasets for fifty rounds and the mean results were recorded. The quantity of membership function of an input attribute was tested for 1, 2 or 3, (i.e., $j = 1, 2, 3$) and the results (Table 5).

TABLE 4. Details of NO$_X$ Emission Datasets.

Dataset	*# Attributes*	*# Training*	*# Testing*
NO$_x$ Emission	150	2,270	1,135

TABLE 5. Results for NO$_X$ Emission of SiRM-ELM Using Differences of Number of Membership Function.

# Membership function of an input attribute	*RMSE*
1	**0.030358**
2	0.056454
3	0.805105

Based on the results of Table 5, the a_i^j and b_i^j were in default setting (in Step 1). After the number of membership function of an input attribute was set as 1, the a_i^j and b_i^j need to be tuned in different ranges in order to get the lowest RMSE. All the tuning results were shown in Table 6.

TABLE 6. Results for NO$_X$ Emission of SIRM-ELM Using Different Ranges of Weights.

Range		*RMSE*
a_i^j	b_i^j	
-1 to +1	-1 to +1	0.030358
-2 to 2	-1 to +1	0.032502
0 to +1	-1 to +1	0.031703
-1 to 0	-1 to +1	0.031173
-1 to +1	0 to +1	0.033823
-1 to +1	-1 to 0	0.033294
-1 to +1	0.5 to +1	0.031032
-1 to +1	**0.5**	**0.028647**

In the experiment of using ELM, two-third of the data samples were trained while the remaining one-third were tested through a validation process to calculate the utmost applicable number of neurons for L (parent ELM). For the sigmoid activation function, the training and validation processes were set with $L =$ fifty units and after that amplified by an increase of fifty units. Table 7 showed the details of the testing processes and the corresponding results based on the sigmoid activation function. The results showed that the greatest RMSE obtained was 0.027086.

In essence, this section presented a framework of Extreme Learning Machine with Single Input Rule Module, which was

TABLE 7. RMSE of NO_X Emission for ELM.

L	RMSE
50	0.036684
100	0.029197
150	0.027448
200	0.027086
250	0.027091
300	0.027233

deemed a significant innovation in ELM ideology (hereafter denoted as SIRM-ELM). Adopting Single Input Rule Module in the ELM hidden layer can be a good alternative to the commonly used activation function, i.e., Sigmoid (SigAct). SIRM-ELM had been tested with Sigmoid hidden neuron using benchmark regression datasets, i.e. Abalone, Balloon, Strike and Space-ga. The results in Table 3 demonstrated that OS-ELM [19], SVM [19] and ELM [1] were better in the proposed model.

Due to the exciting results in the benchmark studies, the SIRM-ELM was used and applied to the NO_X emission in power generation plant.

B. (CASE 2) HYBRIDIZING ELM-MAS (EXTREME LEARNING MACHINE AND MULTI AGENT SYSTEMS)

In the following section, the performance of the ELM-MAS was tested using two benchmark datasets (namely Image Segmentation and Satellite Image). The description of the datasets was displayed in Table 8 [79].

TABLE 8. Description of two Benchmark Datasets.

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

Referring to the model proposed by Liang [79], the number of hidden neuron of each *L* (i.e., ELM^k) was fixed to 180 for Image Segmentation and 400 for Satellite Image. Two-third of the training samples were used for training while the remaining one-third were utilized to work out the most suitable number of neurons of the L_1 (i.e., parent ELM) through a validation process. For each type of the activation function of ELM-MAS, validation and training processes were started by setting $L_1 = 10$ units and then amplified by an increment of 10 units.

An experiment is conducted on the testing datasets for fifty rounds and the mean results are documented. As an example, Table 9 showed a summary of validation and training processes based on sigmoid activation function. From Table 9, the number of hidden neurons with the greatest validation outcome was chosen for ELM-MAS's performance evaluation.

TABLE 9. Summary of Accuracy Rates (Validation) Using Sigmoid Activation Function.

L1	**Image Segmentation**	**Satellite Image**
	Validation Accuracy (%)	Validation Accuracy (%)
10	93.35	87.92
20	94.44	89.02
30	**94.58**	89.25
40	94.50	89.37
50	94.52	89.35
60	94.44	89.36
70	94.37	**89.57**
80	94.52	89.45
90	94.26	89.52
100	94.20	89.53
110	94.29	89.54
120	94.39	89.48
130	94.25	89.48
140	94.21	89.52
150	94.21	89.45
200	93.77	89.50
250	93.50	89.40

TABLE 10. Summary of Accuracy Rates (Test) of the ELM-MAS Using Different Activation Function.

Activation Function	*Image Segmentation*			*Satellite Image*		
	Test Accuracy (%)	**Time (s)**	**# Hidden Neurons**	**Test Accuracy (%)**	**Time (s)**	**# Hidden Neurons**
Sigmoid	94.48	0.30	50	89.27	2.53	200
Laplace Act.	94.87	0.38	200	**89.96**	**2.40**	**150**
RBF	94.70	0.56	100	88.71	7.51	150
Gaussian	94.62	0.36	100	89.47	2.97	400
Laplace Basis	**95.39**	**0.90**	**250**	89.86	7.32	100

Table 10 defined the outcomes by means of ELM-MAS in the context of the test accuracy, training time (seconds), along with the number of hidden neurons for different kinds of activation function. The best results in Table 10 are 89.96% for satellite image used Laplace Act. and 95.39% for image segmentation using Laplace Basis.

An evaluation was also made among other variants of ELMs and the proposed ELM-MAS, such as ELM [93] and ensemble ELM [52]. The test accuracy rates (ELM-MAS) were comparable to ELM (Sigmoid) as well as ELM (RBF) is shown in Table 11.

TABLE 11. Comparison With Other ELM Network.

Activation Function	*Algorithm*	*Image Segmentation*		*Satellite Image*	
		# Hidden Neurons	**Test Accuracy (%)**	**# Hidden Neurons**	**Test Accuracy (%)**
RBF	ELM [93]	180	94.91	400	89.03
	ELM-MAS	100	**94.70**	150	88.71
	Ensemble-ELM [52]	180	91.23	400	**89.28**
Sigmoid	ELM [93]	180	95.07	400	88.97
	ELM-MAS	50	94.48	200	**89.27**
	Ensemble-ELM [52]	180	**94.79**	400	89.01

TABLE 12. Description of Benchmark CWS Datasets.

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

The capability of ELM-MAS was applied to the CWS (circulation water systems) and GAST governor for power generation.

1) REAL-WORLD APPLICATION: NO$_X$ EMISSION OF POWER GENERATION PLANT

Based on Table 12, there were 2500 data samples collected in total and then divided into validation, testing, as well as training sets [94]. ELM-MAS was validated as well as trained to decide on the optimum quantity of hidden neurons prior to the commencement of the tests.

An experiment is conducted on the testing datasets for 50 runs and the mean results are documented. The test accuracy's results were shown in Table 13 and the peak test accuracy of ELM-MAS was 96.96%, accomplished by training with a Laplacian activation function. A comparison was also made between the proposed model trained using other classifiers and a Laplacian activation function, including SVM [94] as well as FAM [95]. In Table 14, the test accuracy rate of ELM-MAS was comparable to SVM [94] as well as FAM [95]. The result for SVM is the highest is because of the complexity of neurons.

TABLE 13. Summary of Accuracy Rates (Test) for Different Activation Functions in CWS Description of Benchmark CWS Datasets.

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

TABLE 14. The Comparison of CWS Datasets.

Algorithm	*Circulating Water System*		
	# Hidden Neurons	**Time (s)**	**Test Accuracy (%)**
ELM-MAS	100	0.26	96.96
FAM [95]	18	-	95.70
SVM [94]	**124**	-	**97.10**

2) REAL-WORLD APPLICATION: GAST GOVERNOR

For a standard operating gas turbine, all training data were gathered on the output of the GAST block, i.e. the mechanical power, P$_{\text{mech}}$ [96]. As listed in Table 15, there were 630 data in total were collected for all the 7 input features in the GAST. These input features were varied within their operating range values [97]. The datasets shown in Table 16 were pre-assigned into validation, test, along with training sets. An experiment is conducted on the testing datasets for fifty rounds and the mean results are documented.

Table 17 displayed the outcomes for using ELM-MAS in the context of the number of hidden neurons, test accuracy, as well as training time in seconds for all activation functions in GAST. The greatest test accuracy rate in Table 17 was 76.79% in the Laplace Basis activation function. The result is the highest due to the low complexity of neurons. On the other hand, the results were also compared with SVM where the SVM result is 77.68%.

TABLE 15. **The Overview of Gast Datasets.**

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

TABLE 16. **Information of the Validation, Testing, and Training, and Their Respective Indications of the Gast Datasets.**

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

TABLE 17. **Accuracy Rates (Test) for Different Activation Function in GAST.**

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

In the summary, this section described a new proposed model with two layers of ELMs which is called ELM-MAS. The ELM-MAS model was certified by using two benchmark datasets (satellite image and image segmentation). The results of ELM-MAS were comparable to ELM (Sigmoid) and ELM (RBF). Moreover, application on power generation system containing governor model (GAST) as well as CWS (circulating water systems) with this model was conducted for assessment. Thus far, the results showed ELM-MAS for CWS was comparable to other algorithms.

Even though outcomes attained from the applications in power generation as well as benchmark studies were encouraging, further research should be conducted with application in other fields for the validation of ELM-MAS

C. (CASE 3) IMPROVED HYBRIDIZING ELM BASED MULTI AGENT SYSTEM USING CERTIFIED BELIEF IN STRENGTH (ELM-MAS-CBS)

Throughout this section, the capability of ELM-MAS-CBS is tested using three benchmark datasets (i.e. Wine, Pima Indians Diabetes (PID) and Iris). Each team had 5 agents (N_1) based on ELM. The number of teams had been set as 3 ($T = 3$) for the experiment. Only Sigmoid and RBF were used in this experiment. Table 18 shown the information of the datasets [54]. The experiments were run in MATLAB (ver.2010) on a private computer equipped with Core(TM) Intel(R) 8 G RAM 2.9 GHz CPU and i7.

TABLE 18. **Details of Benchmark Datasets.**

Dataset	*# Classes*	*# Attributes*	*# Data Samples*
Iris	3	4	150
Wine	3	13	178
PID	23	8	768

Based on the [54], three benchmark datasets were valued using the adopted train-validation-test method in the experiment. An experiment is conducted on the testing datasets for 50 rounds and the mean outcomes are documented. The evaluation of the Wine was based on the tenfold cross-validation method. The meaning is that each Wine dataset was divided into ten subsets, where one is for validation, eight are for training and the remaining is for testing. As for the case of Iris, all of the data samples were used for training (10 % for validation and 90 % for training) as well as for testing. 20% of the PID samples were used to determine the most appropriate number of neurons (i.e., L) through a validation process while 60% were used for training. All the experiments were repeated 10 times.

Sigmoid activation function (SigAct) and Radial Basis Function (RBFun) were the two types of activation functions that were used in each benchmark datasets. The test accuracy rates in Table 18 are based on SigAct for Iris, Wine and PID. In addition, the test accuracy rates based on RBFun for the three benchmark datasets shown in Fig 10. The number of hidden neurons, L with the best test accuracy rate in both Table 19 and Fig. 10 were selected for valuing the capability of ELM-MAS-CBS. An experiment is conducted on the testing datasets for 50 rounds and the mean outcomes are documented.

The outcomes for using ELM-MAS-CBS in terms of the number of hidden neurons and the test accuracy for both activation function in the benchmark datasets is summarized in Table 20. The results of RBFun have the peak test accuracy rate as matched to the SigAct.

The ELM-MAS-CBS was matched with other approaches. As for the comparison, MACS-CBS (Iris datasets) is the highest compared with others but ELM-MAS-CBS (RBFun) is the highest in PID and Wine. Therefore, Fig. 11 displayed that the test accuracy rates of ELM-MAS-CBS were comparable.

The capability of ELM-MAS-CBS was applied to the CWS and GAST governor for power generation.

1) REAL-WORLD APPLICATION: CIRCULATING WATER SYSTEMS

The test was conducted after the ELM-MAS-CBS was validated and trained to discover the ideal number of hidden neurons. The outcomes of test accuracy were recorded

TABLE 19. Details of Benchmark Datasets.

# Hidden Neurons, L	*Iris*	*Wine*	*PID*
	Test Accuracy (%)	**Test Accuracy (%)**	**Test Accuracy (%)**
5	94.53	95.56	74.18
10	97.73	98.33	76.73
15	98.60	98.33	76.67
20	98.67	**98.89**	**77.19**
25	98.60	97.78	76.54
30	**98.73**	98.89	77.12
35	98.60	98.89	76.73
40	98.53	98.89	76.67
45	98.53	98.33	76.54
50	98.40	98.33	76.54

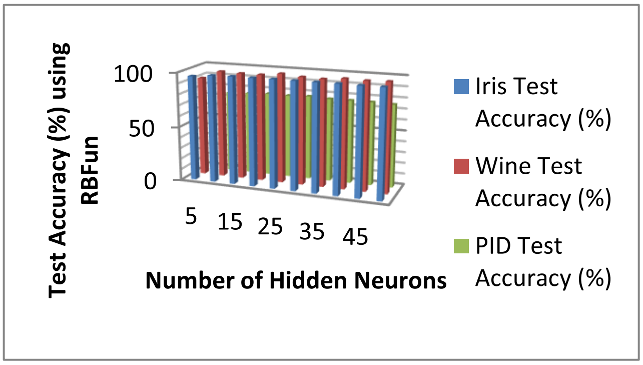


FIGURE 10. Accuracy rates (test) for ELM-MAS-CBS using RBFun.

TABLE 20. Summary of Accuracy Rates (Test) for ELM-MAS-CBS.

Activation Function	*Iris*		*PID*		*Wine*	
	# Hidden Neurons, L	**Test Accuracy (%)**	**# Hidden Neurons, L**	**Test Accuracy (%)**	**# Hidden Neurons, L**	**Test Accuracy (%)**
RBFun	23	**98.87**	27	**77.52**	25	**100**
SigAct	34	**98.87**	20	77.19	22	99.44

in Table 21 with the highest test accuracy of 96.92% using Radial Basis activation function for training. Comparison to other classifiers showed that ELM-MAS-CBS was comparable to SVM [94] and FAM [95], as shown in Table 21. Due to the complexity of hidden neurons in ELM, however, the test accuracy of ELM-MAS-CBS is lesser than ELM-MAS.

2) REAL-WORLD APPLICATION: GAST GOVERNOR

Table 22 summarized the outcomes for using ELM-MAS-CBS in the relation of the test accuracy and the number of hidden neurons for different kind of activation function in GAST. The finest test accuracy rate in Sigmoid activation function was 83.04%. In addition, the comparison between ELM-MAS and ELM-MAS-CBS in Table 17 and Table 22, the test accuracy was better for ELM-MAS-CBS as compared to ELM-MAS.

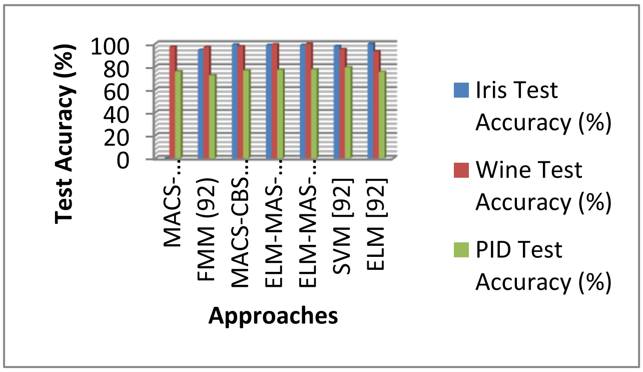


FIGURE 11. The comparison ELM-MAS-CBS with other approaches.

TABLE 21. Comparison ELM-MAS-CBS With Other Approaches Using CWS Dataset.

Approaches	*CWS (Circulating Water Systems)*	
	# Hidden Neurons	**Test Accuracy (%)**
ELM-MAS (Laplace Act.)	100	96.96
ELM-MAS-CBS (SigAct)	55	96.81
ELM-MAS-CBS (RBFun)	65	96.92
SVM [94]	**124**	**97.10**
FAM [95]	18	95.70

TABLE 22. Test Accuracy Rates in GAST.

Activation Function	*GAST*	
	# Hidden Neurons	**Test Accuracy (%)**
RBFun	40	79.57
SigAct	**50**	**83.04**

Lastly, an improved version of ELM-MAS model with certified belief in strength was proven. The proposed model was validated using Wine, Pima Indians Diabetes (PID) and Iris. The results of ELM-MAS-CBS were comparable to ELM (Sigmoid) and ELM (RBF). Moreover, the ELM-MAS-CBS was applied to the governor (GAST) and circulating water systems (CWS) for the power generation system. The results showed that ELM-MAS for CWS was comparable (if not superior) to other approaches.

Even though results were reassuring, further research with the application on other fields are crucial to further validate ELM-MAS-CBS.

IV. CONCLUSION

This paper presented a framework of ELM (Extreme Learning Machine) with Single Input Rule Module (SIRM-ELM), a fresh model (ELM-MAS) with two levels of ELMs and an improved version of ELM-MAS model with certified belief in strength (ELM-MAS-CBS) was established. All those proposed models were validated by utilizing benchmark datasets. The number of hidden neuron is based on trial and error method [79], [98]. It required a tuning process to find a reasonable number of hidden neurons. The experimental outcomes demonstrated that the SIRM-ELM was better than with OS-ELM [19], SVM [19] and ELM [1], as shown in Table 3. For CWS (circulating water systems), the test accuracy rates of ELM-MAS was comparable (if not superior) to other algorithms. Lastly, the comparison between ELM-MAS and ELM-MAS-CBS in Table 17 and Table 22, the test accuracy is better for ELM-MAS-CBS as compared to ELM-MAS. Most importantly, a new development of the hybrid ELM with MAS and SIRM is comparable (if not superior) to other algorithms.

Even though outcomes attained from the applications in power generation as well as benchmark studies were encouraging, further research should be conducted with application in other fields for further validation of SIRM-ELM, ELM-MAS, and ELM-MAS-CBS.

REFERENCES

[1] M. A. W. Saduf, "Comparative study of back propagation learning algorithms for neural networks," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 12, pp. 1151–1156, 2013.

[2] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, "Improving the convergence of the backpropagation algorithm using learning rate adaptation methods," *Neural Comput.*, vol. 11, no. 7, pp. 1769–1796, Oct. 1999.

[3] Y. Nancy Jane, H. Khanna Nehemiah, and K. Arputharaj, "A Q-backpropagated time delay neural network for diagnosing severity of gait disturbances in Parkinson's disease," *J. Biomed. Informat.*, vol. 60, pp. 169–176, Apr. 2016.

[4] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford Univ. Press, 1995.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," La Jolla Inst. Cogn. Sci., California Univ., San Diego, CA, USA, Tech. Rep. ICS-8506, 1985.

[6] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

[7] J. Platt, *A Resource-Allocating Network for Function Interpolation*. Cambridge, MA, USA: MIT Press, 1991.

[8] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks," *IEEE Trans. Syst., Man Cybern., B (Cybern.)*, vol. 34, no. 6, pp. 2284–2292, Dec. 2004.

[9] L. Yingwei, N. Sundararajan, and P. Saratchandran, "Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm," *IEEE Trans. Neural Netw.*, vol. 9, no. 2, pp. 308–318, Mar. 1998.

[10] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 57–67, Jan. 2005.

[11] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Comput.*, vol. 5, no. 6, pp. 954–975, Nov. 1993.

[12] D. J. Swanston, R. J. Mitchell, and J. M. Bishop, "Simple adaptive momentum: New algorithm for training multilayer perceptrons," *Electron. Lett.*, vol. 30, no. 18, pp. 1498–1500, Sep. 1994.

[13] M. Rehman and N. Nawi, "Improving the accuracy of gradient descent back propagation algorithm (GDAM) on classification problems," *Int. J. New Comput. Archit. Their Appl.*, vol. 1, no. 4, pp. 838–847, 2011.

[14] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, Apr. 1994.

[15] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, 'Feed forward neural networks with random weights," in *Proc. Int. Conf. Pattern Recognit.*, 1992, p. 1.

[16] D. Broomhead and D. Lowe, "Feed-forward neural networks and topographic mappings for exploratory data analysis," *Complex Syst.*, vol. 2, pp. 321–355, 1988.

[17] H.-F. Yang and Y.-P.-P. Chen, "Representation learning with extreme learning machines and empirical mode decomposition for wind speed forecasting methods," *Artif. Intell.*, vol. 277, Dec. 2019, Art. no. 103176.

[18] V. Cocco Mariani, S. Hennings Och, L. dos Santos Coelho, and E. Domingues, "Pressure prediction of a spark ignition single cylinder engine using optimized extreme learning machine models," *Appl. Energy*, vol. 249, pp. 204–221, Sep. 2019.

[19] X. Wang, K. Yang, and J. H. Kalivas, "Comparison of extreme learning machine models for gasoline octane number forecasting by near-infrared spectra analysis," *Optik*, vol. 200, Jan. 2020, Art. no. 163325.

[20] I. Markic, M. Stula, and J. Maras, "Intelligent multi agent systems for decision support in insurance industry," in *Proc. 37th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2014, pp. 1118–1123.

[21] S. G. Loizou, "The multi-agent navigation transformation: Tuning-free multi-robot navigation," in *Robotics: Science and Systems*, vol. 6. Berkeley, CA, USA: Univ. of California, Berkeley, 2014, pp. 1516–1523.

[22] S. Srinivasan, D. Kumar, and V. Jaglan, "Multi-agent system supply chain management in steel pipe manufacturing," *IJCSI Int. J. Comput. Sci. Issues*, vol. 7, no. 4, pp. 1694–1814, 2010.

[23] A. Sujil, J. Verma, and R. Kumar, "Multi agent system: Concepts, platforms and applications in power systems," *Artif. Intell. Rev.*, vol. 49, no. 2, pp. 153–182, Feb. 2018.

[24] H. Seki, F. Mizuguchi, S. Watanabe, H. Ishii, and M. Mizumoto, "SIRMs connected fuzzy inference method using kernel method," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2008, pp. 1776–1781.

[25] J. Yi, N. Yubazaki, and K. Hirota, "Anti-swing and positioning control of overhead traveling crane," *Inf. Sci.*, vol. 155, nos. 1–2, pp. 19–42, Oct. 2003.

[26] J. Yi, N. Yubazaki, and K. Hirota, "A proposal of SIRMs dynamically connected fuzzy inference model for plural input fuzzy control," *Fuzzy Sets Syst.*, vol. 125, no. 1, pp. 79–92, Jan. 2002.

[27] J. Yi, N. Yubazaki, and K. Hirota, "A new fuzzy controller for stabilization of parallel-type double inverted pendulum system," *Fuzzy Sets Syst.*, vol. 126, no. 1, pp. 105–119, Feb. 2002.

[28] J. Yi, N. Yubazaki, and K. Hirota, "Upswing and stabilization control of inverted pendulum system based on the SIRMs dynamically connected fuzzy inference model," *Fuzzy Sets Syst.*, vol. 122, no. 1, pp. 139–152, Aug. 2001.

[29] J. Yi, N. Yubazaki, and K. Hirota, "Stabilization control of series-type double inverted pendulum systems using the SIRMs dynamically connected fuzzy inference model," *Artif. Intell. Eng.*, vol. 15, no. 3, pp. 297–308, Jul. 2001.

[30] N. Yubazaki, J. Yi, M. Otani, and K. Hirota, "SIRM's connected fuzzy inference model and its applications to first-order lag systems and second-order lag systems," in *Proc. Soft Comput. Intell. Syst. Inf. Process. Proc. Asian Fuzzy Syst. Symp.*, Dec. 1996, pp. 545–550.

[31] N. Yubazaki, "SIRMs (Single input rule Modules) connected fuzzy inference model," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 1, no. 1, pp. 23–30, Oct. 1997.

[32] B. Saiful Idzwan, C. C. Phing, and T. S. Kiong, "Prediction of Nox using support vector machine for gas turbine emission at Putrajaya power station," *J. Adv. Sci. Eng. Res.*, vol. 4, pp. 37–46, 2014.

[33] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[34] G.-B. Huang, Q.-Y. Zhu, K. Z. Mao, C.-K. Siew, P. Saratchandran, and N. Sundararajan, "Can threshold networks be trained directly?" *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 3, pp. 187–191, Mar. 2006.

[35] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 2004, pp. 985–990.

[36] K. Siah Yap and H. Jen Yap, "Daily maximum load forecasting of consecutive national holidays using OSELM-based multi-agents system with weighted average strategy," *Neurocomputing*, vol. 81, pp. 108–112, Apr. 2012.

[37] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3460–3468, Oct. 2008.

[38] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3056–3062, Oct. 2007.

[39] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[40] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.

[41] S. Lin, X. Liu, J. Fang, and Z. Xu, "Is extreme learning machine feasible? A theoretical assessment (Part II)," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 21–34, Jan. 2015.

[42] X. Liu, S. Lin, J. Fang, and Z. Xu, "Is extreme learning machine feasible? A theoretical assessment (Part I)," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 7–20, Jan. 2015.

[43] Y. Yang, Y. Wang, and X. Yuan, "Bidirectional extreme learning machine for regression problem and its learning effectiveness," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 9, pp. 1498–1505, Sep. 2012.

[44] G. Zhao, Z. Shen, C. Miao, and R. Gay, "Enhanced extreme learning machine with stacked generalization," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1191–1198.

[45] Y. Zhou, J. Peng, and C. L. P. Chen, "Extreme learning machine with composite kernels for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2351–2360, Jun. 2015.

[46] Y. Song, J. Crowcroft, and J. Zhang, "Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine," *J. Neurosci. Methods*, vol. 210, no. 2, pp. 132–146, Sep. 2012.

[47] Y. Zhang and P. Zhang, "Optimization of nonlinear process based on sequential extreme learning machine," *Chem. Eng. Sci.*, vol. 66, no. 20, pp. 4702–4710, Oct. 2011.

[48] R. Minhas, A. A. Mohammed, and Q. M. J. Wu, "Incremental learning in human action recognition based on snippets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 11, pp. 1529–1541, Nov. 2012.

[49] A. H. Nizar, Z. Y. Dong, and Y. Wang, "Power utility nontechnical loss analysis with extreme learning machine method," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 946–955, Aug. 2008.

[50] Z. Yan and J. Wang, "Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 457–469, Mar. 2014.

[51] M. Van Heeswijk, Y. Miche, T. Lindh-Knuutila, P. A. J. Hilbers, T. Honkela, E. Oja, and A. Lendasse, "Adaptive ensemble models of extreme learning machines for time series prediction," in *Proc. Int. Conf. Artif. Neural Netw.* Limassol, Cyprus: Springer, 2009, pp. 305–314.

[52] M. van Heeswijk, Y. Miche, E. Oja, and A. Lendasse, "GPU-accelerated and parallelized ELM ensembles for large-scale regression," *Neurocomputing*, vol. 74, no. 16, pp. 2430–2437, Sep. 2011.

[53] A. Quteishat, C. Peng Lim, J. Tweedale, and L. C. Jain, "A neural network-based multi-agent classifier system," *Neurocomputing*, vol. 72, nos. 7–9, pp. 1639–1647, Mar. 2009.

[54] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3391–3395, Aug. 2009.

[55] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decis. Support Syst.*, vol. 46, no. 1, pp. 411–419, Dec. 2008.

[56] K. Gwebu, J. Wang, and M. D. Troutt, "Constructing a multi-agent system: An architecture for a virtual marketplace," in *Intelligent Decision Support Systems in Agent-Mediated Environments*. Amsterdam, The Netherlands: IOS Press, 2005.

[57] D. L. Hudson and M. E. Cohen, "Use of intelligent agents in the diagnosis of cardiac disorders," in *Proc. Comput. Cardiol.*, Sep. 2002, pp. 633–636.

[58] S. Das, S. Biswas, A. Paul, and A. Dey, "AI doctor: An intelligent approach for medical diagnosis," in *Industry Interactive Innovations in Science, Engineering and Technology*. Kalyani, India: Springer, 2018, pp. 173–183.

[59] H. Salem, G. Attiya, and N. El-Fishawy, "A survey of multi-agent based intelligent decision support system for medical classification problems," *Int. J. Comput. Appl.*, vol. 123, no. 10, pp. 20–25, Aug. 2015.

[60] M. L. Borrajo and J. M. Corchado, "An agent-based virtual organization for risk control in large enterprises," in *Proc. Int. Conf. Knowl. Manage. Organizations*. Žilina, Slovakia: Springer, 2018, pp. 277–287.

[61] M. Oprea, "ABVE-frame: An agent-based virtual enterprise development framework," *AI Commun.*, vol. 30, no. 2, pp. 117–140, May 2017.

[62] H. O. Nyongesa, G. W. Musumba, and N. Chileshe, "Partner selection and performance evaluation framework for a construction-related virtual enterprise: A multi-agent systems approach," *Architectural Eng. Des. Manage.*, vol. 13, no. 5, pp. 344–364, Sep. 2017.

[63] A. Tolk, "An agent-based decision support system architecture for the military domain," in *Intelligent Decision Support Systems in Agent-Mediated Environments*, vol. 115. Amsterdam, The Netherlands: IOS Press, 2005, pp. 187–205.

[64] A. Naseem, S. T. H. Shah, S. A. Khan, and A. W. Malik, "Decision support system for optimum decision making process in threat evaluation and weapon assignment: Current status, challenges and future directions," *Annu. Rev. Control*, vol. 43, pp. 169–187, 2017.

[65] R. Singh, A. Salam, and L. Iyer, "Using agents and XML for knowledge representation and exchange: An intelligent distributed decision support architecture (IDDSA)," in *Proc. AMCIS*, 2003, p. 239.

[66] R. F. Sampaio, L. S. Melo, R. P. S. Leão, G. C. Barroso, and J. R. Bezerra, "Automatic restoration system for power distribution networks based on multi-agent systems," *IET Gener., Transmiss. Distrib.*, vol. 11, no. 2, pp. 475–484, Jan. 2017.

[67] M. Liu, J. Ma, L. Lin, M. Ge, Q. Wang, and C. Liu, "Intelligent assembly system for mechanical products and key technology based on Internet of Things," *J. Intell. Manuf.*, vol. 28, no. 2, pp. 271–299, Feb. 2017.

[68] D. Kortenkamp, R. Simmons, and D. Brugali, "Robotic systems architectures and programming," in *Springer Handbook of Robotics*. Berlin, Germany: Springer-Verlag, 2016, pp. 283–306.

[69] S. Ossowski, "Designing multiagent decision support system the case of transportation management," in *Proc. Null*, 2004, pp. 1470–1471.

[70] M. Le Pira, E. Marcucci, V. Gatta, M. Ignaccolo, G. Inturri, and A. Pluchino, "Towards a decision-support procedure to foster stakeholder involvement and acceptability of urban freight transport policies," *Eur. Transp. Res. Rev.*, vol. 9, no. 4, p. 54, Dec. 2017.

[71] B. Singh and A. Gupta, "Recent trends in intelligent transportation systems: A review," *J. Transp. Literature*, vol. 9, no. 2, pp. 30–34, Apr. 2015.

[72] C.-S. Karavas, G. Kyriakarakos, K. G. Arvanitis, and G. Papadakis, "A multi-agent decentralized energy management system based on distributed intelligence for the design and control of autonomous polygeneration microgrids," *Energy Convers. Manage.*, vol. 103, pp. 166–179, Oct. 2015.

[73] P. Ghadimi, F. Ghassemi Toosi, and C. Heavey, "A multi-agent systems approach for sustainable supplier selection and order allocation in a partnership supply chain," *Eur. J. Oper. Res.*, vol. 269, no. 1, pp. 286–301, Aug. 2018.

[74] A. J. C. Trappey, C. V. Trappey, L. Ma, and J. C. M. Chang, "Intelligent engineering asset management system for power transformer maintenance decision supports under various operating conditions," *Comput. Ind. Eng.*, vol. 84, pp. 3–11, Jun. 2015.

[75] A. A. Lopez-Lorca, G. Beydoun, R. Valencia-Garcia, and R. Martinez-Bejar, "Supporting agent oriented requirement analysis with ontologies," *Int. J. Hum.-Comput. Stud.*, vol. 87, pp. 20–37, Mar. 2016.

[76] S. Ossowski, J. Z. Hernández, C. A. Iglesias, and A. Fernández, "Engineering agent systems for decision support," in *Proc. Int. Workshop Eng. Societies Agents World*. Madrid, Spain: Springer, 2002, pp. 184–198.

[77] S. Liao and C. Feng, "Meta-ELM: ELM with ELM hidden nodes," *Neurocomputing*, vol. 128, pp. 81–87, Mar. 2014.

[78] Y. Guo, S. Rueger, J. Sutiwaraphun, and J. Forbes-Millott, "Meta-learning for parallel data mining," in *Proc. 7th Parallel Comput. Workshop*, 1997, pp. 1–2.

[79] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

[80] S. C. Tan, C. P. Lim, and M. V. C. Rao, "A hybrid neural network model for rule generation and its application to process fault detection and diagnosis," *Eng. Appl. Artif. Intell.*, vol. 20, no. 2, pp. 203–213, Mar. 2007.

[81] N. L. A. A. Aziz, K. S. Yap, and M. A. Bunyamin, "A hybrid fuzzy logic and extreme learning machine for improving efficiency of circulating water systems in power generation plant," in *Proc. IOP Conf., Earth Environ. Sci.*, vol. 16, no. 1, Bristol, U.K.: IOP Publishing, 2013, Art. no. 012102.

[82] S. C. Tan and C. P. Lim, "Application of an adaptive neural network with symbolic rule extraction to fault detection and diagnosis in a power generation plant," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 369–377, Jun. 2004.

[83] K. P. Wong, "Artificial intelligence and neural network applications in power systems," in *Proc. 2nd Int. Conf. Adv. Power Syst. Control, Operation Manage. (APSCOM)*, 1993, pp. 37–46.

[84] K. Baba and N. Kakimoto, "Dynamic behavior of a combined cycle power plant in the presence of a frequency drop," *Electr. Eng. Jpn.*, vol. 143, no. 3, pp. 9–19, May 2003.

[85] H. Shalan, M. M. Hassan, and A. Bahgat, "Comparative study on modeling of gas turbines in combined cycle power plants," in *Proc. 14th Int. Middle East Power Syst. Conf. (MEPCON)*, Giza, Egypt: Cairo Univ., 2010.

[86] T. Inoue, Y. Sudo, A. Takeuchi, Y. Mitani, and Y. Nakachi, "Development of a combined cycle plant model for power system dynamic simulation studies," *IEEJ Trans. Power Energy*, vol. 119, no. 7, pp. 788–797, 1999.

[87] K. Kunitomi, A. Kurita, H. Okamoto, Y. Tada, S. Ihara, P. Pourbeik, W. W. Price, A. B. Leirbukt, and J. J. Sanchez-Gasca, "Modeling frequency dependency of gas turbine output," in *Proc. IEEE Power Eng. Soc. Winter Meeting. Conf.*, Jan./Feb. 2001, pp. 678–683.

[88] S. Suzuki, K. Kawata, M. Sekoguchi, and M. Goto, "Combined cycle plant model for power system dynamic simulation study," *IEEJ Trans. Power Energy*, vol. 120, nos. 8–9, pp. 1146–1152, 2000.

[89] W. I. Rowen, "Simplified mathematical representations of heavy-duty gas turbines," *J. Eng. for Power*, vol. 105, no. 4, pp. 865–869, Oct. 1983.

[90] F. De Mello and D. Ahner, "Dynamic models for combined cycle plants in power system studies," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1698–1708, 1994

[91] A. Quteishat, C. P. Lim, J. M. Saleh, J. Tweedale, and L. C. Jain, "A neural network-based multi-agent classifier system with a Bayesian formalism for trust measurement," *Soft Comput.*, vol. 15, no. 2, pp. 221–231, Feb. 2011.

[92] D. E. Golberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, no. 102. Reading, MA, USA: Addison-Wesley,1989, p. 36, 1989.

[93] J. M. Fossaceca, T. A. Mazzuchi, and S. Sarkani, "MARK-ELM: Application of a novel multiple kernel learning framework for improving the robustness of network intrusion detection," *Expert Syst. Appl.*, vol. 42, no. 8, pp. 4062–4080, May 2015.

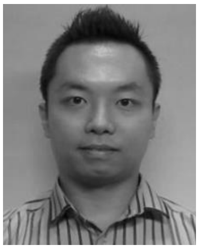
[94] K. Siah Yap, C. Peng Lim, and M. Teng Au, "Improved GART neural network model for pattern classification and rule extraction with application to power systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2310–2323, Dec. 2011.

[95] A. Quteishat and C. P. Lim, "A modified fuzzy min–max neural network with rule extraction and its application to fault detection and classification," *Appl. Soft Comput.*, vol. 8, no. 2, pp. 985–995, 2008.

[96] C. Yaw, N. Namas Khan, U. Ungku Amirulddin, A. Hashim, and M. Harun, "Development of gas turbines model in MATLAB to investigate response in the event of major system contingencies," Power Tech, Gurugram, Haryana, Tech. Rep., 2009, doi: 10.1016/j.asoc.2007.07.013.

[97] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, "AI techniques in induction machines diagnosis including the speed ripple effect," in *Proc. Conf. Rec. IEEE Ind. Appl. Conf. 31st IAS Annu. Meeting*, vol. 1, Oct. 1996, pp. 655–662.

[98] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," *Math. Problems Eng.*, vol. 2013, pp. 1–11, 2013.



CHONG TAK YAW received the bachelor's degree (Hons.) in electrical and electronics engineering, the master's degree in electrical and electronics engineering, and the Ph.D. degree in artificial neural network from Universiti Tenaga Nasional (UNITEN), Malaysia, in 2008, 2012, and 2019, respectively. He is currently working as a Postdoctoral Researcher with the Institute of Sustainable Energy, UNITEN. His research interests include artificial neural networks and renewable energy.

KEEM SIAH YAP received the B.Eng. degree (Hons.) in electrical and the M.Sc. degree in electrical engineering from Universiti Teknologi Malaysia, in 1998 and 2000, respectively, and the Ph.D. degree in electronics engineering from Universiti Sains Malaysia, in 2010. He is currently a Professor with the College of Engineering, Universiti Tenaga Nasional, Malaysia. He is also a Professional Engineer registered to the Board of Engineers Malaysia. His research interest includes theory and applications of artificial intelligence.

SHEN YUONG WONG (Senior Member, IEEE) received the bachelor's and master's degrees (Hons.) in electrical and electronic engineering and the Ph.D. degree in engineering from Universiti Tenaga National (National Energy University), Malaysia, in 2010, 2012, and 2015, respectively. She is currently with the Department of Electrical & Electronics Engineering, Xiamen University Malaysia. She is also a Professional Engineer registered to the Board of Engineers Malaysia, accredited under the Washington Accord. Her research interests include computer vision and image processing, extreme learning machine, deep learning, classification, regression, and other applications of artificial intelligence. She also serves as the Editorial Board member for *Applied Soft Computing* journal.

HWA JEN YAP received the bachelor's degree in mechanical engineering, the master of engineering science degree, and the Ph.D. degree from the University of Malaya, Kuala Lumpur, Malaysia, in 2000, 2005, and 2012, respectively. He was the Head of the Centre of Product Design & Manufacturing (CPDM), from 2013 to 2015. He was appointed as the Deputy Dean (Development) with the Faculty of Engineering, University of Malaya, from 2017 to 2018. He currently serves as an Associate Professor with the Department of Mechanical Engineering, Faculty of Engineering, University of Malaya. His research interests include virtual reality (VR), augmented reality (AR), product design, robotics and automation, artificial intelligence (AI), and industrial revolution 4.0 (IR4.0). He is a member of The Institute of Engineering and Technology (IET) and a Chartered Engineer of the Engineering Council, U.K.

JOHNNY KOH SIEW PAW received the bachelor's degree (Hons.) in electrical & electronic engineering and the M.Sc. and Ph.D. degrees from Universiti Putra Malaysia, in 2000, 2002, and 2008, respectively. He is currently an Associate Professor with the Department of Electronics and Communication Engineering, Universiti Tenaga Nasional. His research interests include machine intelligence, automation technology, and renewable energy.