Intelligent Decision Technologies 12 (2018) 163–171
DOI 10.3233/IDT-180325
IOS Press

An ELM based multi-agent system and its applications to power generation

Chong Tak Yaw[a], Shen Yuong Wong[a], Keem Siah Yap[a], Hwa Jen Yap[b], Ungku Anisa Ungku Amirulddin[c] and Shing Chiang Tan[d]
[a]*Department of Electronics and Communication Engineering, Universiti Tenaga Nasional, Malaysia*
[b]*Faculty of Engineering, University of Malaya, Malaysia*
[c]*Department of Electrical Power Engineering, Universiti Tenaga Nasional, Malaysia*
[d]*Faculty of Information Science and technology, Multimedia University, Malaysia*

Abstract. This paper presents an implementation of Extreme Learning Machine (ELM) in the Multi-Agent System (MAS). The proposed method is a trust measurement approach namely Certified Belief in Strength (CBS) for Extreme Learning Machine in Multi-Agent Systems (ELM-MAS-CBS). The CBS is applied on the individual agents of MAS, i.e., ELM neural network. The trust measurement is introduced to compute reputation and strength of the individual agents. Strong elements that are related to the ELM agents are assembled to form the trust management in which will be letting the CBS method to improve the performance in MAS. The efficacy of the ELM-MAS-CBS model is verified with several activation functions using benchmark datasets (i.e., Pima Indians Diabetes, Iris and Wine) and real world applications (i.e., circulating water systems and governor). The results show that the proposed ELM-MAS-CBS model is able to achieve better accuracy as compared with other approaches.

Keywords: Certified belief in strength, extreme learning machine, neural network, multi-agent system, pattern classification, power generation

1. Introduction

The Extreme Learning Machine (ELM) has proven to be an efficient learning algorithm over the years as compared to the traditional learning methods in the aspect of generalization and learning speed [1–6]. ELM is capable of making universal approximation with random input weights and biases [7]. In other words, the hidden neurons are not required in neuron alike and the weights are the parameters that need to learn the connection between the hidden layer and the output layer.

Based on Huang et al. [8], ELM is extraordinarily efficient and lean towards to global optimum as compared with the traditional feedforward neural network (FNN) [8]. In addition, ELM can reach the best generalization bound of the traditional FNN where all the parameters are learned with commonly used activation functions [9]. In terms of efficiency and generalization, the performance of ELM is far better than traditional FNN and has been experimented in different kind of problems. The application of ELM has also been exemplified in different fields such as biomedical analysis [10,11], chemical process [12], system modeling [13,14], power systems [15], action recognition [16], hyperspectral images [17], etc.

There is some research works focusing on ensemble model to combine individual prediction of multiple ELMs to give a final output [18–22]. This strategy is also adopted in a Multi-Agent System (MAS). MAS allows the subproblems of a constraint satisfaction problem to be subcontracted to different problem solving agents with their own interests and goals. Thus, MAS has been applied to tackle problems in different fields successfully in the past decade. This is evidenced by a widespread application of MAS to

*Corresponding author: Chong Tak Yaw, Department of Electronics and Communication Engineering, Universiti Tenaga Nasional, Malaysia. E-mail: takhehe@yahoo.com.

ISSN 1872-4981/18/$35.00

different domains including e-Commerce [23], healthcare [24], military support [25], decision support [26], knowledge management [27], as well as control systems [28]. The general structure of a MAS is shown in the Fig. 1 where the base platform is built by a group of ELMs that are deemed as individual agents. In this structure, the outcome of an individual ELM (individual agent) is sent to a parent agent, which is the decision combination module to make the final decision.

Generally, the average output in the decision combination module is based on the methods such as exact average [20], weighted average [7], confusion matrix [29], and voting [30]. Unfortunately, those approaches required additional algorithms to generate outcome in the decision combination module.

Recognition and rejection accuracy rates based trust measurement has been proposed [21]. In the model, two teams were used where the first team consists of three modified Fuzzy min-max (FMM) agents and the second team consists of three modified Fuzzy ARTMAP (FAM) agents. The model was presented with better performances as compared with other approaches mentioned in [21]. Another trust measurement strategy based on Bayesian formalism with FMM MAS was proposed in [31]. In this model, the FMM is used as a learning agent in MAS and followed by combining with Bayesian formalism to obtain a trust measurement. The results show that the model is able to yield the better performances as compared with other approaches mentioned in [31].

In the recent development of MAS model for trust measurement, a method namely Certified Belief in Strength (CBS), which based on strength and reputation of individual FMM based agent [31]. During the training process, trust is the strong elements that are related with the FMM agents which let the CBS method to improve the performance of the MAS. As a result, the CBS improved the performance of the MAS model by improving the accuracy rates of the individual agents.

Nowadays, an element that plays an important role in daily life is trust and trustworthiness. Especially occur in our social environment. Basically the element allows the consignment of duties and decisions to applicable persons, who can execute the duties [32]. The element had been developed in few areas, such as in e-business filed by Mui et al. [33] and in wireless sensor networks by Boukerche and Li [34].

These papers propose an extended CBS method using Extreme Learning Machine based Multi-Agent System (hereafter denoted as ELM-MAS-CBS). The

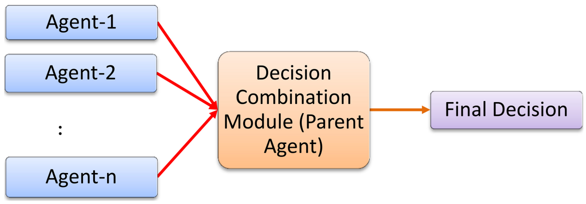

Fig. 1. A general structure of MAS.

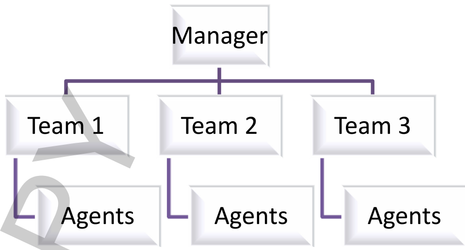

Fig. 2. Overview of ELM-MAS-CBS model.

difference is that MACS CBS used FMM which consists of multiple hyperboxes while the proposed approach employs a “team” concept which involves a group of individual ELM-based agents.

This paper is organized as follows. The algorithms of ELM-MAS-CBS are explained in Section 2. The flow chart of ELM-MAS-CBS is showed in Section 3. Section 4 showed the results and discussion of the benchmarking datasets. The applications of ELM-MAS-CBS in power generation are presented in Section 5. Lastly, Section 6 is the conclusion.

2. The algorithms of ELM-MAS-CBS

In this paper, the ELM-MAS-CBS model consists of three layers as shown in Fig. 2, i.e., the first layer consists of several individual ELM-based agents; the second layer consists of several teams of ELM-based and The CBS is implemented in the individual ELM-based agents. Then, the Manager will select the team with the highest CBS as the final decision as the output. In this paper, the number of teams is set as 3 ($T = 3$). The number of agent used in a team is set as 5 ($K = 5$). An ELM-based agent is denoted as ELM^{tk} (for $t = 1, \ldots, T$, for $k = 1, \ldots, K$).

The step-by-step training procedure is given as follows.

Step 1: The input weights $\mathbf{a}_i^{tk}$ and b_i^{tk} are assigned randomly. For all steps/equations of training process, variables run for $i = 1, \ldots, L$ (number of hidden neuron of ELM), for $t = 1, \ldots, T$, and $k = 1, 2, \ldots, K$.

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

$$\mathbf{H}^{tk} = \begin{bmatrix} G(\mathbf{a}_1^{tk}, b_1^{tk}, \mathbf{x}_1) \ldots G(\mathbf{a}_L^{tk}, b_L^{tk}, \mathbf{x}_1) \\ \vdots \quad \ldots \quad \vdots \\ G(\mathbf{a}_1^{tk}, b_1^{tk}, \mathbf{x}_N) \ldots G(\mathbf{a}_L^{tk}, b_L^{tk}, \mathbf{x}_N) \end{bmatrix}_{N \times L} \quad (1)$$

$$G(\mathbf{a}_i^{tk}, b_i^{tk}, \mathbf{x}_j) = \frac{1}{1 + \exp\{-(\mathbf{a}_i^{tk} \cdot \mathbf{x}_j + b_i^{tk})\}} \quad \text{(Sigmoid)} \quad (2)$$

$$G(\mathbf{a}_i^{tk}, b_i^{tk}, \mathbf{x}_j) = \exp\left\{-b_i^{tk} \left\|\mathbf{x}_j - \mathbf{a}_i^{tk}\right\|^2\right\} \quad \text{(RBF)} \quad (3)$$

Step 3: The output weights of *ELM*^{tk},β^{tk} are computed by using the following equation,

$$\beta^{tk} = \left((\mathbf{H}^{tk})^T(\mathbf{H}^{tk})\right)^{-1} (\mathbf{H}^{tk})^T\mathbf{T}, \quad (4)$$

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

Step 4: Once output weights of *ELM*^{tk} are computed, the training samples are used to compute the outputs of *ELM*^{tk}, i.e.,

$$\mathbf{y}^{tk} = \mathit{ELM}^{tk}(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i^{tk} G(\mathbf{a}_i^{tk}, b_i^{tk}, \mathbf{x}_j) \quad \text{for } j = 1, \ldots, N \quad (5)$$

Step 5: The accuracy rates of the *ELM* ^{tk} are calculated.

$$A^{tk} = \frac{N^{tk}}{N} \times 100\% \quad (6)$$

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

Step 6: Given the validation samples, the output of *ELM*^{tk} is calculated based on Eq. (5).

Step 7: Given an initial strength of CBS for all team is 100 (**S** = [100 100 100]) [21] and initial bid coefficient (C_{bid}) is 0.01 [21]. The initial team bid is in proportion to strength as follows [23],

$$B^t = C_{bid}S^t \quad (7)$$

Step 8: With the validation samples, the trust element, C^t , is calculated as shown in Eq. (8). Use the Eq. (6) to determine C^k in order to look for the accuracy rate of the agents in each team. Then, the ELM with the highest accuracy rate is chosen (denoted as *ELM*^{tw} where w is the winner of the team) and will be representing its team and will be applying into Eq. (8) and will be submitting to the manager.

$$C^t = C_{bid}\left(S^t + A^{tw}\right) \quad (8)$$

Step 9: According to [21], the Eq. (7) is further modified as the reward and penalty to update the strength shown in Eq. (9), where P is penalty and$R$ is reward. When an agent generates a correct prediction, $P = 0$ while $R = B^t$; otherwise, $P = B^t$ while $R = 0$.

$$S^t(new) = S^t(now) - P + R \quad (9)$$

Step 10: Since S^t is updated, hence both the A^{tk} and the B^t are updated based on Eqs (6) and (7) respectively.

Once all the samples are trained using Step 1 to Step 10, the ELM-MAS-CBS can be used for prediction of a newly arrived and unknown input vector **z**. The step-by-step prediction procedure is given as follows.

Step 11: All the $\mathbf{a}_i^{tk}$, b_i^{tk}, β^{tk}, A^{tk}, S^t, and C^t are loaded from completed training process. For all steps/equations of predictions procedures, variables are run for $i = 1, \ldots, L$, for $t = 1, \ldots, T$, and $k = 1$, 2,$\ldots, K$.

Step 12: The hidden layer output matrix for *ELM*^{tk}, $\mathbf{h}^{tk}$, is calculated as follows.

$$\mathbf{h}^{tk} = [G(\mathbf{a}_1^{tk}, b_1^{tk}, \mathbf{z}) \ldots G(\mathbf{a}_L^{tk}, b_L^{tk}, \mathbf{z})]_{1\times L} \quad (10)$$

Step 13: The outputs of *ELM*^{tk} are calculated,

$$\mathbf{y}^{tk} = \mathbf{h}^{tk}\boldsymbol{\beta}^{tk} \quad (11)$$

Step 14: The highest value of accuracy rates is selected from each team (denoted as A^{tU}), and are applied in Eq. (13) to calculate the trust elements of teams.

$$A^{tU} = \arg\max_k \left(A^{tk}\right) \quad (12)$$

$$C^t = C_{bid}\left(S^t + A^{tU}\right) \quad (13)$$

Step 15: The highest value of the C^t from all teams (denoted as C^V) is determined, where V is the winner from all teams, i.e.,

$$C^V = \arg\max_t \left(C^t\right) \quad (14)$$

Step 16: The final output of ELM-MAS-CBS can be found based on the Eq. (11), where $t = V$ (winning team) and $k = U$ (winning agent of the winning team).

The proposed sequence of ELM-MAS-CBS model is summarized in Fig. 3.

3. The Flow Chart of ELM-MAS-CBS

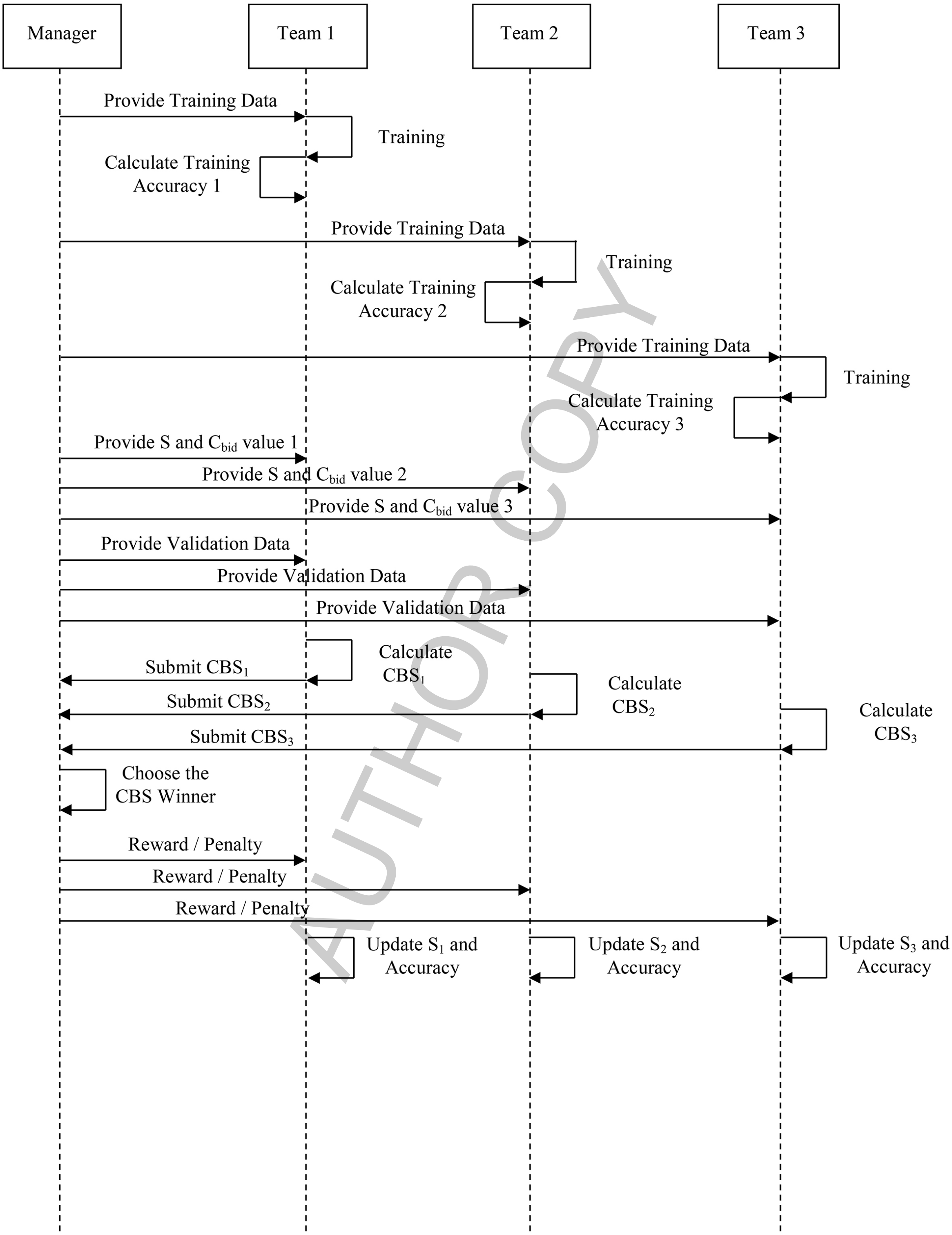

Fig. 3. Sequence of algorithms of ELM-MAS-CBS.

Table 1
Specification of benchmark datasets [20]

Dataset	# Attributes	# Classes	# Data samples
PID	8	2	768
Iris	4	3	150
Wine	13	3	178

Table 2
Testing accuracy rates of ELM-MAS-CBS using sigmoid activation function

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

4. Results on the benchmarking data

Throughout this paper, there were three benchmark datasets (e.g. Pima Indians Diabetes (PID), Iris and Wine) were used to test the performance of ELM-MACS-CBS. For all experiments, the number of teams had set as 3 ($T = 3$). Each team had 5 agents (N_1) based on ELM. In the experiment, both activation functions (Sigmoid, SigAct and RBFun) are used. The specifications of the datasets are shown in Table 1 [20]. All experiments were ran in MATLAB (ver.2010) on a personal computer equipped with Intel(R) Core(TM) i7 2.9 GHz CPU and 8 G RAM.

In the experiment, three benchmark datasets were evaluated using the adopted train-validation-test method. The 60% of the PID samples were used for training while the 20% were used to determine the most appropriate number of neurons (i.e., L) through a validation process. In the case of Iris, 100% of the data samples were used for training (90% for training and 10% for validation) and 100% for testing. All the experiments were repeated for 10 times. The tenfold cross-validation method was used to evaluate the Wine. Each Wine data set was divided into 10 subsets where 8 for training and 1 for validation and the remaining for testing.

There are two types of activation functions, i.e., Radial Basis Function (RBFun) and Sigmoid activation function (SigAct) are used in each benchmark dataset. Table 2 showed the accuracy rates based on sigmoid activation function for PID, Iris and Wine. In addition, Table 3 also shows the accuracy rates based on radial basis activation function for the three benchmark datasets. Among the Tables 2 and 3, the number of hidden neurons, L with the best test accuracy rate is selected for evaluating the performance of ELM-MAS-CBS.

Table 3
Testing accuracy rates of ELM-MAS-CBS using radial basis activation function

# Hidden Neurons, L	PID Test accuracy (%)	Iris Test accuracy (%)	Wine Test accuracy (%)
5	71.70	96.27	91.11
10	75.03	98.07	98.33
15	75.88	98.53	97.78
20	76.67	98.53	97.78
25	76.34	**98.67**	**100**
30	**77.12**	98.60	98.33
35	76.14	98.60	97.78
40	76.21	98.67	99.44
45	76.14	98.40	98.89
50	75.49	98.40	99.44

In the Tables 2 and 3, the increasing number of hidden neuron is not improved the accuracy rate. This is because that this situation in called overfitting, where the neural networks overestimate the complexity of the targeted problem. On the other hand, it also greatly degrades generalization capability, which leads to significant deviation in predictions. By doing this, allocating the proper number of hidden neurons to prevent overfitting is critical in function approximation using feedforward neural network.

Table 4 summarizes the results for using ELM-MAS-CBS in terms of the test accuracy and the number of hidden neurons for two types of activation function in the benchmark datasets. The results showed that the RBFun has the highest test accuracy rate as compared with the SigAct.

The proposed ELM-MAS-CBS is compared with other ELM. Table 5 showed that the test accuracy rates of ELM-MAS-CBS are comparable (if not superior) with MACS-TNC [13] and MACS-CBS [20].

5. Application in power generation

The ELM-MAS-CBS is applied on the power generation in following section.

5.1. Circulation water systems

An overview of Circulating Water System (CWS) is shown in Fig. 4 [36,37]. This system consists of pip-

Table 4
Summary for test accuracy rates of ELM-MAS-CBS

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

Table 5
Comparison with other approaches

Algorithm	PID	Iris	Wine
	Test accuracy (%)	Test accuracy (%)	Test accuracy (%)
FMM [20]	72.46	94.65	97.15
MACS-TNC [13]	75.82	–	97.32
MACS-CBS [20]	76.58	**99.33**	97.67
ELM-MAS-CBS (RBFun)	**77.52**	98.87	**100**
ELM-MAS-CBS (SigAct)	77.19	98.87	99.44

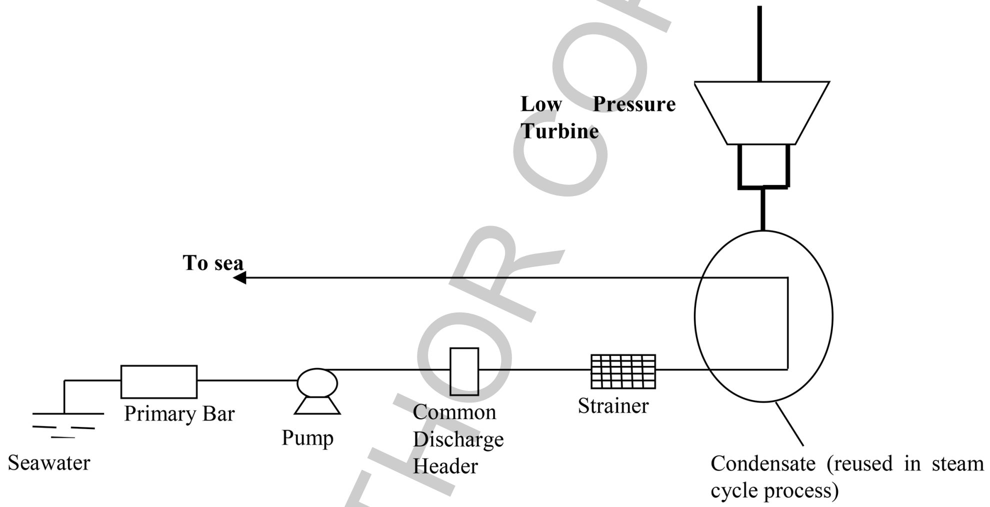

Fig. 4. Circulating water systems.

ing, turbine condensers and drum strainer between the inlet of the sea water and the outfall where the water will be returned into the sea. The major component in the CWS is the turbine condenser where it is used to remove the heat from the low pressure steam while attempting to maintain turbine backpressure at the lowest possible yet constant level.

A total number of 2500 data samples were collected and they were categorized into training, validation, and testing sets, as shown in Table 6 [38]. The proposed ELM-MAS-CBS was trained and validated to determine the optimal number of hidden neurons before it was tested. The results of test accuracy are listed in Table 7. The highest test accuracy of ELM-MAS-CBS is 96.92% and it was achieved by training ELM-MAS-CBS with a Radial Basis activation function. The proposed ELM-MAS-CBS trained using a Radial Basis activation function is compared with other classifiers, which include FAM [37] and SVM [38]. From Table 7, the test accuracy rate of ELM-MAS-CBS is comparable (if not superior) with FAM [37] and SVM [38].

5.2. GAST governor

The GAST is one of the governor model [40]. It represents the principal dynamic characteristics of industrial gas turbines driving generators connected to electric power systems. Speed variations from nominal are expected to be small (approximately 5%). The model shown in Fig. 5 consists of a forward path with governor time constant, T_1, and a combustion chamber time constant, T_2, together with a load-limiting feedback

Table 6
Specification of Benchmark Dataset in 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

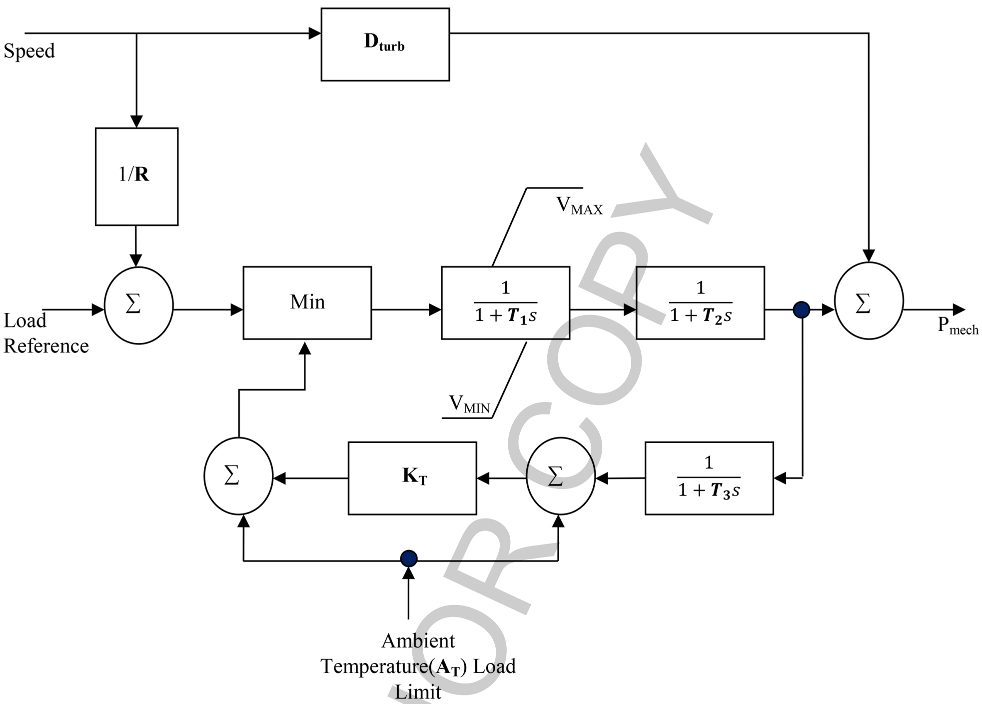

Fig. 5. Governor model, GAST [40].

Table 7
Comparison of CWS dataset

Algorithm	Circulating water systems	
	Test accuracy (%)	# Hidden neurons
ELM-MAS-CBS (RBFun)	96.92	65
ELM-MAS-CBS (SigAct)	96.81	55
FAM [37]	95.70	18
SVM [38]	97.10	124

Table 8
Summary of dataset in GAST

Class	Range	# Training
1	$0 < R < 1$	100
2	$0 < D_{turb} < 0.5$	50
3	$0 < A_T \leqslant 1$	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

path. The load limit is sensitive to turbine exhaust temperature, and T_3 represents the time constant of the exhaust gas measuring system. The constant, K_T, is used to adjust the gain of the load-limited (A_T) feedback path.

The training data is collected on the output of the GAST block which is the mechanical power, P_{mech} for a normal operating gas turbine [39]. The total data collected is 630 for all seven input attributes in the GAST shown in Table 8 These seven input attributes in the governor are varied within their operating range values [40]. The data are pre-divided into 60% training, 20% validation, and 20% testing sets, as shown in Table 9.

Table 10 summarizes the results for using ELM-MAS-CBS in terms of the training time (seconds), test accuracy, and the number of hidden neurons for each activation function in GAST.As the results, the best test accuracy rate is 83.04% in Sigmoid activation function.

Table 9
Details of the training, validation, and testing of the GAST dataset

Class	# Training	# Validation	# Testing	# Total
1	60	20	20	100
2	30	10	10	50
3	60	20	20	100
4	30	10	10	50
5	30	10	10	50
6	84	28	28	140
7	84	28	28	140
			Total	630

Table 10
Test accuracy rates for two activation function in GAST

Activation function	GAST	
	Test accuracy (%)	# Hidden neurons
RBFun	79.57	40
SigAct	**83.04**	**50**

6. Conclusion

In this research, a new ELM-MAS-CBS model with three layers of ELMs has been developed. These papers propose an extended CBS method using Extreme Learning Machine based Multi-Agent System (hereafter denoted as ELM-MAS-CBS). The difference is that MACS CBS used FMM which consists of multiple hyperboxes while the proposed approach employs a "team" concept which involves a group of individual ELM-based agents.

The developed model is validated by using benchmark datasets which are Pima Indians Diabetes (PID), Iris and Wine. In the Table 5, the test accuracy for PID and Wine are higher when compare with other approaches but is lower in Iris. Therefore based on the outcome, the test accuracy rates of ELM-MAS-CBS are comparable (if not superior) with MACS-TNC [13] and MACS-CBS [20]. Not only that, the developed model also applied its application on the power generation which are circulating water systems and governor (GAST). The experimental results showed that the test accuracy rates of ELM-MAS-CBS for circulating water systems is comparable (if not superior) with other algorithms in which the proposed model is higher than FAM [37] and lower than SVM [38].

Although the results obtained from the benchmark studies on Pima Indians Diabetes (PID), Iris and Wine and applications in power generation (circulating water systems and governor, GAST) are encouraging, more studies using datasets from various application domains are required to validate the applicability of ELM-MAS-CBS in real world application. In addition, investigation of proposed model in nonstationary applications by replacing ELM of ELM-MAS-CBS with OSELM (online sequence version of ELM [7]) could be another research for further work.

The generalized activation functions are added to 'future works'. For example, the generalized RBF (GRBF) activation function can continuously and smoothly reproduce different RBF by changing a real parameter. In addition, the generalized Gaussian distribution for GRBF can add a shape parameter to normal Gaussian distribution [41]. Therefore, a better corresponding between the shape of the kernel and the distribution of the distances.

References

[1] Huang GB, Bai Z, Kasun LLC, Vong CM. Local receptive based extreme learning machine, IEEE Computational Intelligence Magazine. May 2015; 10(2).

[2] Huang GB. An insight into extreme learning machines: random neurons, Features and Kernels, CognComput. 2014; 6: 376-390.

[3] Huang G-B, Ding X, Zhou H. Optimization method based extreme learning machine for classification, Neurocomputing. 2010; 74: 155-163.

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

[5] Huang GB, Chen L. Convex incremental extreme learning machine, Neurocomputing. 2007; 70(168): 3056-3062.

[6] Huang GB, Chen L. Enhanced random search based incremental Extreme Learning Machine, Neurocomputing. 2008; 71(16): 3460-3468.

[7] Yap KS, Yap HJ. Daily maximum load forecasting of consecutive national holidays using oselm-based multi-agents system with average strategy, Neurocomputing. 2012; 81: 108-112.

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

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

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

[11] Song Y, Crowcroft J, Zhang J, Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine, Journal of Neuroscience Methods. 2012; 210(2): 132-146.

[12] Zhang Y, Zhang P. Optimization of nonlinear process based on sequential extreme learning machine, Chemical Engineering Science. 2011; 66(20): 4702-4710.

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

[14] Yan Z, Wang J. 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. 2014; 25(3): 457-469.

[15] Nizar A, Dong Z, Wang Y. Power utility nontechnical loss analysis with extreme learning machine method, IEEE Transactions on Power Systems. 2008; 23(3): 946-955.

[16] Minhas R, Mohammed AA, Wu QMJ. Incremental learning in human action recognition based on snippets, IEEE Transactions on Circuits and Systems for Video Technology. 2012; 22(11): 1529-1541.

[17] Zhou Y, Peng J, Chen CLP. Extreme learning machine with composite kernels for hyperspectral image classification, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2014.

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

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

[20] Lan Y, Soh YC, Huang GB. Ensemble of online sequential Extreme Learning Machine, Neurocomputing. 2009; 72(135): 3391-3395.

[21] Anas Q, Lim CP, Jeffrey T, Lakhmi CJ. A Neural Network-based Multi-agent Classifier System, Neurocomputing. 2009; 72: 1639-1647.

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

[23] Gwebu K, Wang J, Troutt MD. 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, 2005.

[24] Hudson DL, Cohen ME. Use of Intelligent Agents in the Diagnosis of Cardiac Disorders, Computers in Cardiology. 2002; 633-636.

[25] Tolk A, 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, 2005.

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

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

[28] Ossowski S, Hernandez JZ, Iglesias CA. A. Fernandez, Engineering Agent Systems for Decision Support, The 3rd International Workshop Engineering Societies in the Agents World, 2002; 184-198.

[29] Marom ND, Rokach L. A. Shmilovici, Using the confusion matrix for improving ensemble classifiers, Electrical and Electronics Engineers in Israel (IEEEI), 2010.

[30] Cao JW, Lin ZP, Huang GB, Nan L. Voting based Extreme Learning Machine, Information Sciences. 2012; 185: 66-77.

[31] Anas Q, Lim CP, Saleh JM, Tweedale J, Jain LC. A Neural Network-based Multi-agent Classifier System with a Bayesian Formalism for Trust Measurement, Soft. Compt. 2011; 15(2); 221-231.

[32] Ries S, Kangasharju J, Mühlhäuser M. A Classification of Trust Systems, Lect. Notes Comput, Sci. Move Meaningful Intern Syst. 2006; 4277; 894-903.

[33] Mui L, Mohtashemi M, Halberstadt A. A computational model of trust and reputation, In: Hawaii international conference on system sciences-HICSS. 2002; 2431-2439.

[34] Boukerche A, Li X. An agent-based trust and reputation management scheme for wireless sensor networks, IEEE Glob. Telecommun. Conf. 2005; 3(5).

[35] Yaw CT, Yap KS, Yap HJ, Ungku Amirulddin UA. An ELM Based Multi Agent Systems Using Certified Belief in Strength, ICONIP 2014 (III) (LNCS 8836) Springer. 2014; 458-465.

[36] Wong SY, Yap KS, Yap HJ, Tan SC, Chang SW. On Equivalence of FIS and ELM for Interpretable Rule-Based Knowledge Representation, IEEE Trans. Neural Networks and Learning Systems. July 2015; 26(7).

[37] Tan SC, Lim CP, Rao MVC. A Hybrid Neural Network Model for Rule Generation and its Application to Process Fault Detection and Diagnosis, Engineering Applications of Artificial Intelligence. 2007; 20: 203-213.

[38] Yap KS, Lim CP, Au MT. Improved GART Neural Network Model for Pattern Classification and Rule Extraction with Application to Power Systems, IEEE Trans. Neural Networks. 2011; 22(12).

[39] Yaw CT, Namas Khan NR, Ungku Amirulddin UA, Hashim AH, Harun MN. Development of Gas Turbines Model in MATLAB to Investigate Response in the Event of Major System Contingencies, PowerTech 2009, 28 June–2 July 2009, Bucharest, Romania.

[40] PSS/E$^{\mathrm{TM}}$ 26, Program Application Guide: Volume II, December 1998, 1990–1998 PowerTechnologies, Inc.

[41] Navarro FF, Martínez CH, Ruiz R, Riquelme JC. Evolutionary Generalized Radial Basis Function neural networks for improving prediction accuracy in gene classification using feature selection, Applied Soft Computing. 2012; 12: 1787-1800.