The Impact of Randomization on Circular-Complex Extreme Learning Machine for Real Valued Classification Problems

Ram Govind Singh
Asst. Professor
Deptt. Of CSE & IT
UIT Allahabad

Akhil Pandey
PG Student
Deptt. Of CSE & IT
UIT Allahabad

ABSTRACT
Extreme Learning Machine (ELM) has recently emerged as a fast classifier giving good performance. Circular–Complex extreme learning machine (CC-ELM) is recently proposed complex variant of ELM which has fully complex activation function. It has been shown that CC-ELM outperforms real valued and other complex valued classifiers. In both CCELM & ELM parameters between input and hidden layer are initialized randomly and the weights between hidden and output layer are obtained analytically. Due to this randomization, the performance of both ELM & CC-ELM fluctuates. In this paper, performance fluctuation due to random parameter of CC-ELM and the circular transformation function have been analyzed first, then by using an Ensemble approach namely Bagging, a variants Bagging.C1 is proposed to bring the stability in the performance of CC-ELM. In Bagging.C1 various data samples are generated by using random parameters of circular transformation function. Performance of proposed classifier ensemble is evaluated using a set of benchmark real-valued classification problems from the University of California, Irvine machine learning repository.

Keywords
classification; complex-valued neural networks; extreme learning machine

1. INTRODUCTION
Extreme learning machine is a fast classifier with a good prediction ability which was recently proposed by Huang et al [1]. It is a single layer feed-forward neural network in which the input weights and hidden layer biases are initialized randomly and the weights between hidden layer and output layer are determined analytically using Moore-Penrose generalized inverse H† of the output matrix of hidden layer. Unlike other traditional gradient descent learning algorithm such a backpropagation ELM does not require tuning of the parameters like learning rate, learning epoch etc and can provide better performance in terms of learning speed, reliability and generalization Tian et. al [2]. It does not have to encounter problems like stopping criteria and local minima Wang et. al. [3].

With the evolution of technologies that include the processing of complex-valued signals like, signal processing, adaptive array processing [4, 5], image processing [6] it has been a necessity to create and develop complex-valued neural networks. R.Savitha et. al [7] proposed a Fast learning circular complex valued extreme learning machine (CC-ELM) for the classification of real valued data sets in complex domain.CC-ELM uses sin(z) as circular transformation function and sech() as fully complex Gaussian type activation function. Circular transformation function, an orthogonal decision boundary of CC-ELM at hidden layer and output layer shows better performance and prediction ability of real valued data than ELM and various complex valued classifier [7]. Various variants of ELM also have been proposed in literature to enhance the performance like OS-ELM [8], I-ELM [9], W-ELM [10], C-ELM [11] etc. However these variants could not address some drawbacks of ELM [12] which are mentioned below:

1. The ELM and its variants randomly select the input weights and the hidden layer bias. This causes instability in the predictions of ELM..
2. The ELMs suffer from the overfitting problem. This is due to more number of hidden nodes on larger datasets and the complexity of the input instances distribution.
3. When the size of the dataset increases the order of the matrix H also increases. This implies that when larger datasets are used large memory is required to calculate the Moore-Penrose inverse.

Hansen and Salmon have proved in [13] that the above mention problems can be solved by using ensemble learning. The ensemble learning aims at reducing the risk of modeling error by combining several base learners. Each base learner in the ensemble can be generated either by creating diversity in the dataset used for each classifier by using subsets of a larger dataset or the whole dataset by creating diversity in terms of settings in the learning algorithm [14]. Yu Liu [15] proved that variation among components input weights and initial parameter forces those components to have diverse output space which increases the diversity and generalization ability of an ensemble model. Jiuwen Cao et. al. [16] also have proved that large number of multiple realization of ELM reduces the misclassified sample, have the lower variance and is able to correctly classify test sample with probability of one.

Ensemble of weak learner model may differ in terms of 1).the base classifier used for prediction 2). Approach used to create various model either by selecting different realization of model or by using different samples of data set. 3).ensembling approach is used and 4). The ensemble pruning algorithm is used. The two popular methods for an ensemble are Bagging [17], Boosting [18] and its variants. In the literature variants of ELM based on Bagging are, V-ELM [16],in this it is proved that ability to correctly classify a test data with a large number of training model is probability of one and the final prediction is done by majority voting . Chen et. al. [19] proposed an ensemble based on simple averaging that select ELM models on the basis of product index for each model by the correlation between the MSE value and ambiguity from a population of models. Liu et. al. [20] proposed an EN-ELM that generate RxM model from R fold cross validation over M iteration and on the basis of output norms weights first half of the model are elected. Heeswijk et al [21] introduced an adaptive ensemble model in which average weights of the model are updated when new example are arrived. Yuan Lan et. al. [22] introduced EOS-ELM which uses ensemble method for sequential learning when data arise with fixed or varying size or chunk by chunk. Zhai et. al [12] introduced DE-ELM which uses entropy of an instance for model pruning and Gaitang Wang et. al [23] proposed ELM ensemble uses fuzzy activation function for output and dynamic AdaBoost in which weights of ELM components are modified based upon optimum threshold .

However very less research work has been conducted about the performance ability of ensemble classifier in a complex domain. In this paper for CVNN variants of Bagging is evaluated. Two complex-valued ensemble based classifiers namely, V-CELM.C1 (Voting based-Complex Extreme Learning Machine.C1) and V-CELM.C2 (Voting based-Complex Extreme Learning Machine.C2) are implemented using a variants of Bagging in the complex domain named as Bagging.C1.

For evaluating the performance of the ensemble based classifiers the simulations has run on datasets retrieved from the open UCI repository. Eight multi-class and six binary-class datasets have chosen for the experiments. Along with the various experiments presented in this paper to study the effectiveness of our classifiers, we have also used the Wilcoxon ranksum method to strengthen our claims of building an efficient complex based ensemble classifier.

The rest of the paper is organized as follows. Section 2 gives a brief description of the CC-ELM and ensemble methods. Section 3, presents the work and Section 4 presents the experimental study. Finally conclusion is done in Section 5.

2. CIRCULAR COMPLEX-VALUED EXTREME LEARNING MACHINE

CC-ELM has chosen as base classifier among other complex-valued extreme learning machines due to it efficient transformation mapping and better accuracy in prediction. For creating a complex-valued extreme learning machine the basic assumption is that a set of N observations namely O, can be represented as : O = {(xi,ci)|xi ∈ Rn, i = 1,2,…..,N}, where xi is the n-dimensional real-valued input vector and ci ∈ CV, where CV = {1,2,…,C} is the class vector containing coded class labels with C number of classes. The following function is used to obtain the coded-class label, ccrt :

$$\text{ccrt} = \begin{cases} 1+i, & \text{if } c_t = r, \\ -1-i, & \text{otherwise,} \end{cases} \qquad r=1,2,\ldots,C \qquad (1)$$

The transformation function used in CC-ELM to map

the real-valued input features into complex domain is the sin () function.

$$z_l = \sin(ax_l + ibx_l + \propto_l) \qquad (2)$$

The sine function is analytic and almost bounded everywhere [16] which makes it a suitable function to be used in the transformation part of the CC-ELM classifier. a and b(0 < a,b ≤ 1) are real-valued non-zero transformation constant and $\propto (0 < \propto < 2\pi)$ is the non-zero translational/rotational bias.

The CC-ELMs use a fully complex valued activation function in the hidden layer of the type of hyperbolic secant function [14]. The responses of the sech activation are given as in (3):

$$hj = \text{sech}(ujT(zt - vj));\ j=1,\ldots, K \qquad (3)$$

where uj and vj ∈ Cn are complex-valued scaling factor and complex-valued center of the j-th hidden neuron respectively.

The output layer neurons employ a linear activation function in the CELM. The output $\hat{y}$ of the CELM network with K hidden neurons is:

$$\widehat{y_n} = \sum_{j=1}^{K} w_{nj} h_j \,, \quad (4)$$

where wnj are the weights connecting the n-th output neuron with the j-th hidden neuron. Equation (4) can also be written as in Equation (5).

$$\hat{Y} = WH \quad (5)$$

where W is the matrix of all output weights connecting the hidden and output layer neurons. H is the response matrix of the hidden layer and is given as in Equation (6).

H(V,U,Z) =

$$\begin{bmatrix} \text{sech}\,({u_1}^T \|z_1 - v_1\|) & . & . & \text{sech}\,({u_1}^T \|z_N - v_1\|) \\ \vdots & \ddots & \vdots & \vdots \\ \text{sech}\,({u_K}^T \|z_1 - v_K\|) & . & . & \text{sech}\,({u_K}^T \|z_N - v_K\|) \end{bmatrix} \quad (6)$$

Here H is a K × N matrix, where K is the number of hidden neurons and N is the number of samples to be trained. The parameters (uj,vj) chosen randomly and The output weights W are calculated by the least squares method according to Equation (7):

$$W = YH^{\dagger}, \qquad (7)$$

Where H† is the Moore-Penrose pseudo-inverse of the hidden layer output matrix and Y is the complex-valued coded class label.

From the outputs, the class labels are estimated as below:

$$\hat{c} = \max_{l=1,2,\ldots,C} \text{real}\,(\widehat{y_l}) \qquad (8)$$

3. PROPOSED WORK: CC-ELM BASED ENSEMBLE METHODS (*BAGGING.C1)*

In this paper it has been tried to significantly improve the classification performance of CC-ELM by incorporating variants of bagging using the random parameter of circular transformation function. Major focus has been rendered to the method by which datasets are chosen for training by each base learner in the ensemble. The description of the variants of bagging namely: 'Bagging.C1' is given below:

V-CELM.C1 using Bagging.C1: To bring diversity in order to implement an ensemble (i) data can be diverse and the classifier constant, (ii) data is kept constant and classifier varied or (iii) bring both data and classifier diversity together. In the algorithm Bagging.C1 we follow the latter method, i.e., to incorporate both data and classifier diversity. In each iteration of Bagging. C1, a new dataset is provided to the base classifier.

Table 1: Bagging.C1 Algorithm

Algorithm 1: Bagging.C1

Input: O_{DS}**:** Original training set = $\{(x_i , c_i)| \ x_i \in R^n, c_i \in CV, i = 1,2,\ldots,N\}$; *J*: Number of iterations; n: size of O_{DS} and Bootstrap, *CC-ELM:* Base classifier

Output: *final_CLP* : Final class prediction using majority voting

Algorithm

1. **for** $j = 1$ to J
2. L_j = *RandomTransformationSample* (O_{DS})
3. Generate a new CC-ELM_j model by randomly choosing U and V
4. Calculate the class prediction, CLP_j for CC-ELM_j $CLP_j = \text{output}(\text{CC-ELM}_j (L_j))$
5. **End**
6. $\text{final\_CLP} = \max_{j=1,\ldots,J} (CLP)$

function RandomTransformationSample(O_{DS}*)*

1. Select random numbers $a\in [0,1]$,$b \in [0,1]$,$\alpha \in [0,2\pi]$
2. **for** $i = 1,\ldots, N$
3. $z_i = \sin (ax_i + bx_i + \alpha)$
4. $cc_r^i = \begin{cases} \mathbf{1 + i}, & \boldsymbol{c_i = r} \\ \mathbf{-1 - i}, & \textbf{otherwise} \end{cases}$ $r=1,2,\ldots,C$
5. **end**
6. **return** $L= \{(z_i,cc_i)|z_i \in C^n,cc_i \in CC, i=1,2,\ldots,N\}$, CC is the complex coded class label matrix

The original dataset ODs which is in real-valued format is transformed to complex-valued dataset by the transformation function: zi = sin (axi + bxi + α) where a and b ($0 < a, b \leq 1$) is real-valued non-zero transformation constant and $\propto (0 < \propto < 2\pi)$ is the non-zero translational/rotational bias. The values of a, b and $\propto$ are randomly chosen for each classifier in the ensemble thereby generating random samples of the original dataset. The classifier specifications itself are changed for each classifier in the ensemble by varying the parameters u and v of the CC-ELM classifier. We use random values of a, b and $\propto$ for changing the dataset for each classifier. The impact of transformation function on the classification ability of complex valued neural networks has been stressed on [14], [15] and [16]. Thus creating diverse datasets from the original dataset by randomizing the parameters a, b and $\propto$ are supposed to bring good performance results in bagging. Subsequent to training, in the testing phase when a new instance is provided to the ensemble, the class prediction is performed by a majority voting of the class predictions by the individual CC-ELM classifiers in the ensemble. We generate an ensemble based classifier termed as V-CELM.C1 which implements Bagging.C1 algorithm. The entire algorithm is presented in detail in Table no. 1.

V-CELM.C2: In V-CELM.C2, instead of providing random samples of the original dataset OD for each classifier in the ensemble, we produce only one sample and use the same across all the CC-ELM classifiers. The diversity is only in terms of the parameters u and v of the CC-ELM classifier in each iteration of the ensemble. The remaining steps are the same as for Bagging.C1.

4. EXPERIMENTAL STUDY AND RESULT ANALYSIS

To evaluate the efficacy of the proposed ensemble techniques Four experiments has conducted in this study and for the proposed method the simulation is done on The MATLAB 7.10.0 (R2010a) running on Core 2 DUO PC. The experiments have used a total 14 datasets obtained from the UCI repository [23] of which 8 and 6 are multi-class datasets and binary class datasets respectively. The information regarding the datasets is provided in Table No.2.

Table 2: Datasets used and their description

Data-set	Category	Instances	Features	Classes	Training Instances	Testing Instances
Pen digits (DS1)	Multi	10992	16	10	7494	3498
Optical digits(DS2)	Multi	5620	64	10	3823	1797
Waveform(DS3)	Multi	5000	40	3	3000	2000
Image(DS4)	Multi	2310	19	7	210	2100
Vehicle (DS5)	Multi	846	18	4	424	422
Balance(DS6)	Multi	625	4	3	400	225
Ecoli(DS7)	Multi	336	7	8	168	168
Wine(DS8)	Multi	178	13	3	100	78
Spambase(DS9)	Binary	4601	57	2	2300	2301
Pima(DS10)	Binary	768	8	2	400	368
Cancer(DS11)	Binary	683	9	2	300	383
Cancer1(DS12)	Binary	569	30	2	300	269
Ionosphere(DS13)	Binary	351	34	2	100	251
Heart(DS14)	Binary	270	13	2	100	170

Here the datasets of various sizes have been used across our experimental study. The number of training instances and testing instances used for each dataset has been mentioned in the Table. We use twenty base CC-ELM classifiers throughout our experiments (*J=20*). The number of hidden nodes generated for each CC-ELM classifier in the ensemble is as per given in [16]. In order to Prove the efficiency of the ensemble methods over the single classifier models; two complex-valued ensemble based classifiers *V-CELM.C1, V-CELM.C2* are compared with the CC-ELM classifier in four major aspects:

1. The impact of building ensemble methods over a single neural network based classifier
2. Average testing accuracy
3. The impact of stability in the performance of an ensemble classifier over a single classifier model.
4. Statistical test namely *Wilicoxon test* to ensure the performance overhand of the ensemble techniques.

We mention each experiment, their respective results and a detailed study of the proposed ensemble methods based on these results.

4.1 The impact of building CC-ELM base classifier over the proposed ensemble methods

To enhance the performance of CC-ELM Bagging.C1 has been used to generate the base classifier with diversity. In this section we study the power of ensemble method of CC-ELM base classifier with data set Waveform (DS3).

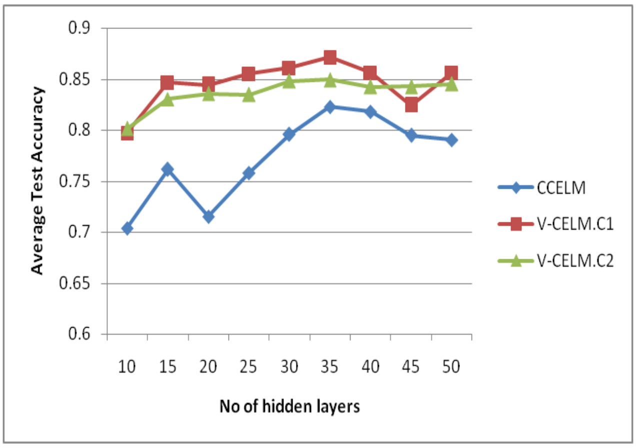

Fig.1: The impact of construction of CC-ELM sub classifier on Ensemble Waveform data set (DS3)

The experiment is performed by varying the hidden nodes of CC-ELM from 10 to 50 and the results are shown for the testing accuracy in Fig. 1. The CC-ELM is evaluated our proposed algorithms and From the results it is vivid that the testing accuracy of the ensemble methods are far better than that of a single CC-ELM model. It implies that while building an ensemble rather than using a single model better prediction capability can be achieved with less number of computational resources.

In Fig. 2 performance of dataset Spambase (DS9) is evaluated with varying no. of training numbers, to analyze the performance fluctuation due to the random parameter of circular transformation function. The result clearly depicts that performance vary too much when we use a single model.

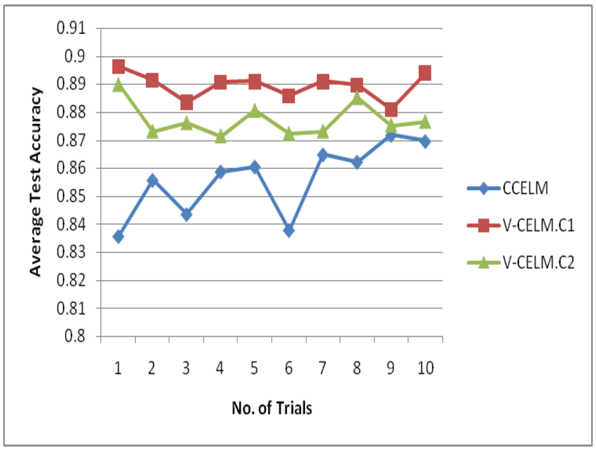

Fig.2: The Experimental result in average test accuracy with varying number of Training numbers of Spambase data set(DS9)

The result also describes that randomization of circular transformation function a, b and α creates more variation than the random parameter between input and hidden layer u and v. When ensemble of base classifier has used both performance and stability increases highly.

Table no. 3: Average test accuracy results of CC-ELM and ensemble methods for multi-class datasets

Data set	Hidden node	CC-ELM	V-CELM.C1	V-CELM.C2
balance	10	81.751	83.831	84.186
ecoli	10	83.726	86.678	84.678
image	60	90.843	93.934	92.495
vehicle	85	77.611	79.099	77.933
waveform	35	83.16	87.175	85.315
wine	10	89.769	92.81	92.20
optical	60	90.535	92.099	91.109
pen	50	79.081	85.975	84.033

The performance results for other datasets mentioned in Table No. 5 match with those that we have provided for the Waveform multi-class dataset. The results are not displayed due to the limitation in space to present them.

Table no. 4: Average test accuracy results of CC-ELM and ensemble methods for binary class datasets

Data set	Hidden node	CC-ELM	V-CELM.C1	V-CELM.C2
CANCER	15	97.68	98.791	97.702
CANCER1	15	86.46	88.208	87.096
HEART	15	85.36	86.988	85.658
IONO	15	81.74	83.932	82.924
PIMA	20	78.38	78.548	78.09
SPAMBASE	50	86.64	89.175	88.381

4.2 The impact of ensemble methods on the testing accuracy of base classifier CC-ELM

In this section, we intend to analyze the testing accuracy of the proposed ensemble methods over the original CC-ELM classifier. For this we have chosen to show the results of testing accuracy over the 14 datasets including multiclass and binary class datasets already described in Table no. 2. For each dataset, the experiment was run for 50 times and the average of the testing accuracy over the fifty results was retrieved and is presented in Table no. 3 for multi-class datasets and Table no. 4 for binary class datasets.

The tables clearly represent the efficacy of our methods. In almost all the datasets, the test accuracy of the ensemble methods are quite better than that of the basic CC-ELM classifier as a single model. Since we have used datasets of various sizes across our experiment, our results clearly indicate that the ensemble methods we proposed can outperform the single CC-ELM model in almost all the cases whether the data range be small, medium and large. We had intended to check the importance of using random values of *a*, *b* and α in producing random datasets for V-CELM.C1. The results show that this technique is effective. And our *Random Transformation Sample()* method is indeed efficient. The diversity in terms of classifier can bring significant changes to the classification ability for an ensemble.

4.3 Testing the stability of the ensemble methods over CC-ELM base classifier

In this experiment, we test the stability of the ensemble methods over various network size and data size. To demonstrate that our proposed methods have better stability over changing conditions of network and data ranges, we conduct experiments varying the datasets of large, medium and small sizes over different network sizes of 30, 40 and 50 hidden nodes.

The datasets chosen are Pen digits (large), waveform (medium) and heart (small). The experiment is run on each dataset for CC-ELM and two ensemble methods ten times each. The results are presented in Table no. 5. The observations from the table are briefly summarized below:

Table 5: stability test by varying the hidden node

	CC-ELM			V-CELM.C1			V-CELM.C2		
Data set	h_30	h_40	h_50	h_30	h_40	h_50	h_30	h_40	h_50
Pen digit	72.8	77.4	79.0	80.6	83.1	83.6	80.5	83.0	83.5
Spam base	83.5	85.1	86.2	86.0	86.8	87.1	86.6	87.1	87.3
Vehicle	76.1	78.0	78.4	77.3	78.6	79.3	78.5	79.0	80.2

1. As the number of hidden nodes increases, the testing accuracy of all the classifiers increases respectively in the case of any dataset irrespective of its size. The entire proposed ensemble methods can be seen to have higher testing accuracies than that of the original CC-ELM method.

2. It can be seen that all the ensemble methods presented here are better in stability for larger datasets when compared to the original-ELM which presents more fluctuations. The bagging methods which use stochastic replacement of the dataset for each base classifier provides more stable results for large data instances.

3. The medium size spam dataset exhibit far better results for the ensemble methods when compared to both large and small datasets. This is however obvious as Spambase as we have chosen is a binary class dataset and the performance overhand are quite reasonable. .

4.4 Statistical test: Wilcoxon test to verify the performance overhand of the proposed methods

Various statistical methods are used in literature to prove the efficiency of the neural network classifiers like Wilcoxon test, paired t-test etc. We use the Wilcoxon test in our experiments to analyze, evaluate and conclude that our methods are undeniably better than the original CC-ELM classifier. The experiments are conducted over 6 datasets where for each dataset the CC-ELM and one of the proposed methods are run for 10, 30 and 50 times. We thus generate six statistics for each ensemble method to ensure its efficacy over the CC-ELM. We denote the dataset for the original CC-ELM and our proposed method as O_i and P_i where $i \in \{1, 2, 3\}$. When $i = 1$, O_1 and P_1 are vectors of size 10, when i=2 O_2 and P_2 are vectors of size 30 and when i=3 O_1 and P_2 are size 50 vectors.

Table 6: CC-ELM and V-CELM.C1 rank test

	Wilicoxon test of CC-ELM and V-CELM.C2					
Dataset	**10 trial**		**30 trial**		**50 trial**	
	p_value 1	h_value 1	p_value 1	h_value 1	p_value 1	h_value 1
Heart	0.0518	0	1.45E-06	1	1.35E-08	1
Spam	0.0002	1	9.84E-11	1	2.10E-14	1
Balance	0.4034	0	0.471574	0	0.033979	1
Optical	0.0008	1	1.45E-10	1	8.78E-16	1
Wave	0.0002	1	6.00E-11	1	2.75E-17	1
Wine	0.0492	1	0.001836	1	0.004687	1

For conducting the Wilcoxon test we use the MATLAB *ranksum*() function. The ranksum was calculated three times for each dataset with each proposed method against the original CC-ELM as *ranksum*(O_1,P_1), *ranksum*(O_2,P_2) and *ranksum*(O_3,P_3). The Table no. 6 and Table no. 7, shows the p values and h values of Wilcoxon test tested between CC-ELM against *V-CELM.C1 and V-CELM.C2*. According to the Wilcoxon test when the value of p is too small, it shadows a doubt on the validity of the null hypothesis. Therefore the values of p and h clearly indicate the significant overhand of our proposed methods over the original CC-ELM. Thus we have proved statistically that building an ensemble using CC-ELM can bring noteworthy improvement in the classification ability of the complex-valued neural network classifier.

5. CONCLUSION

In this paper, to reduce the performance variation and to enhance the stability, an ensemble based method on CC-ELM base classifier has implemented which also improves the generalization ability and class prediction capability of the fast learning CC-ELM classifier. For introducing the ensemble methods into a complex domain a popular ensemble techniques namely Bagging has chosen and made improvements so that it may be used to solve real-valued classification problems using CC-ELM. The newly proposed algorithms are tested for their classification ability against a single CC-ELM classifier to show the better performance of an ensemble. We found that the classifier based on the algorithm Bagging.C1 does due to the diversity we brought in the ensemble by randomly changing the values of a, b and α in the dataset. The experimental study further demonstrates that our approach is strong and robust.

Table 7: CC-ELM and V-CELM.C2 rank test

	Wilicoxon test of CC-ELM and V-CELM.C1					
Dataset	**10 trial**		**30 trial**		**50 trial**	
	p_value 1	h_value 1	p_value 1	h_value 1	p_value 1	h_value 1
Heart	0.1314	0	1.92E-05	1	2.58E-07	1
Spam	0.0036	1	1.97E-07	1	3.37E-08	1
Balance	0.4488	0	0.588585	0	0.054447	0
Optical	0.0012	1	1.94E-10	1	8.74E-16	1
Wave	0.0002	1	3.65E-11	1	8.85E-18	1
Wine	0.0171	1	0.000322	1	0.00033	1

6. REFERENCES

[1] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501. 2.

[2] H.-X. Tian, Z.-Z. Mao, An ensemble ELM based on modified Adaboost.RT algorithm for predicting the temperature of molten steel in ladle furnace, IEEE Trans. Autom. Sci. Eng. 7 (1) (2010) 73–80.

[3] D. Wang, M. Alhamdoosh, Evolutionary extreme learning machine ensembles with size control, Neurocomputing 102 (2013) 98–110.

[4] J.C. Bregains, F. Ares, Analysis, synthesis, and diagnostics of antenna arrays through complex-value neural networks, Microwave Opt. Technol. Lett. 48 (8) (2006) 1512–1515.

[5] R. Savitha, S. Vigneshwaran, S. Suresh, N. Sundararajan, Adaptive beamform-ing using complex-valued radial basis function neural networks, in: Proceed-ings of the TENCON'09, IEEE Region 10 Annual International Conference, Singapore, November 23–26, 2009, pp. 1–6.

[6] N. Sinha, M. Saranathan, K.R. Ramakrishna, S. Suresh, Parallel magnetic resonance imaging using neural networks, Proceedings of ICIP'07 IEEE International Conference on Image Processing, vol. 3, 2007, pp. 149–152.

[7] R. Savitha S. Suresh, N. Sundararajan, Fast learning Circular Complex-valued Extreme Learning Machine(CC-ELM) for real-valued classification problems, Information Sciences, Vol. 187 (2012) pp. 277–290.

[8] G.B. Huang, N, Liang, H. Rong, P. Saratchandran and N. Sundararajan, "On-line sequential Extreme Learning Machine", IASTED International Conference on Computational Intelligence (CI 2005), Calgary, Canada, July 4-6, 2005.

[9] G. Huang, L. Chen, "Convex incremental extreme learning machine", Neurocomputing, Lttr. Vol. 70, pp. 3056-3062.

[10] W. Zong, G. Huang and Y. Chen, "Weighted extreme learning machine for imbalance learning", Neurocomputing, Vol. 101, pp. 229-242.

[11] M. Li, G Huang, P. Saratchandran and N. Sundararajan, "Fully complex extreme learning machine", Neurocomputing, Letters, Vol. 68, 2005, pp. 306-314.

[12] J Zhai, H Xu and X Wang, "Dynamic ensemble extreme learning machine based on sample entropy", Softcomputing, vol. 16, 2012, pp. 1493—1502.

[13] L.K. Hansen and P. Salmon, "Neural network ensembles", IEEE Transactions on Pattern analysis and machine intelligence, Vol. 12, no. 10, October 1990

[14] M.P. Perrone, L.N. Cooper, When Networks Disagree: Ensemble Methods for Hybrid Neural Networks, Technical Report A121062, Brown University, Institute for Brain and Neural Systems, January 1993

[15] Y.Liu, X. Xu, C. Wang , "Simple Ensemble of Extreme Learning Machine", 2009.

[16] J Cao, Z Lin and G Huang and N Liu, "Voting based extreme learning machine", Information Sciences Vol. 185 (2012) pp. 66–77.

[17] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, pp. 123–140, 1996.

[18] R. E. Schapire, "The strength of weak learnability," Mach. Learn.,vol.5, pp. 197–227, 1990.

[19] H. Chen, H. Chen, X. Nian, P. Liu, Ensembling extreme learning machines, in: Advances in Neural Networks, Lecture Notes in Computer Science, vol. 4491, Springer, Berlin, Heidelberg, 2007, pp. 1069–1076.

[20] N. Liu, H. Wang, Ensemble based extreme learning machine, IEEE Signal Process. Lett. 17 (8) (2010) 754–757.

[21] M. Heeswijk, Y. Miche, T. Lindh-Knuutila, P.A. Hilbers, T. Honkela, E. Oja, A. Lendasse, Adaptive ensemble models of extreme learning machines for time series prediction, in: Proceedings of the 19th International Conference on Artificial Neural Networks, Springer-Verlag, 2009, pp. 305–314.

[22] Y Lan, Y C Soh and G. Huang, "Ensemble of online sequential extreme learning machine", Neurocomputing Vol. 72 (2009) 3391–3395.

[23] G Wang and P Li, "Dynamic Adaboost Ensemble Extreme Learning Machine", 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE).

[24] Machine learning repository http://archive.ics.uci.edu/ml/.