



Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Regular Paper

An improved cuckoo search based extreme learning machine for medical data classification

P. Mohapatra^a, S. Chakravarty^b, P.K. Dash^{c,*}

^a IIIT, Bhubaneswar, India

^b Orissa Engineering College, Bhubaneswar, India

^c Siksha 'O' Anusandhan University, Bhubaneswar, India

ARTICLE INFO

Article history:

Received 5 October 2014

Received in revised form

29 May 2015

Accepted 31 May 2015

Keywords:

Extreme learning machine (ELM)

Online sequential extreme learning machine (OSELM)

Cuckoo search (CS)

Cuckoo search based extreme learning machine (CSELM)

Improved cuckoo search based extreme learning machine (ICSELM)

ABSTRACT

Machine learning techniques are being increasingly used for detection and diagnosis of diseases for its accuracy and efficiency in pattern classification. In this paper, improved cuckoo search based extreme learning machine (ICSELM) is proposed to classify binary medical datasets. Extreme learning machine (ELM) is widely used as a learning algorithm for training single layer feed forward neural networks (SLFN) in the field of classification. However, to make the model more stable, an evolutionary algorithm improved cuckoo search (ICS) is used to pre-train ELM by selecting the input weights and hidden biases. Like ELM, Moore-Penrose (MP) generalized inverse is used in ICSELM to analytically determines the output weights. To evaluate the effectiveness of the proposed model, four benchmark datasets, i.e. Breast Cancer, Diabetes, Bupa and Hepatitis from the UCI Repository of Machine Learning are used. A number of useful performance evaluation measures including accuracy, sensitivity, specificity, confusion matrix, Gmean, F-score and norm of the output weights as well as the area under the receiver operating characteristic (ROC) curve are computed. The results are analyzed and compared with both ELM based models like ELM, on-line sequential extreme learning algorithm (OSELM), CSELM and other artificial neural networks i.e. multi-layered perceptron (MLP), MLPICS, MLPNNCS and radial basis function neural network (RBFNN), RBFNNCS, RBFNNICS. The experimental results demonstrate that the ICSELM model outperforms other models.

© 2015 Published by Elsevier B.V.

1. Introduction

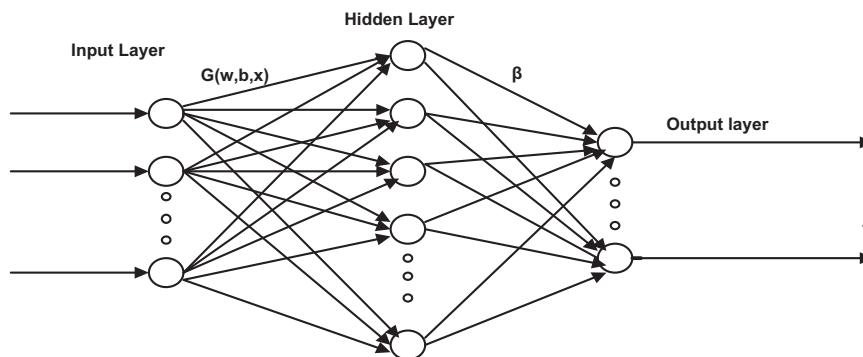
Classification of the exponentially growing complex, continuous data with large number of records and features has become the most challenging data mining tasks of human activity. In the last 20 years, it is being applied in the field of pattern classification like optical character recognition [1], text and image classification [2], machine vision [3], fraud detection [4], natural language processing [5], market segmentation [6,7], bioinformatics [8], protein sequence classification [9], biomedical image classification [10] and real world data classification [11]. The research community has given increasing attention in developing fast and accurate classifiers with good generalization capability. Even though a lot of classifiers [12] are already available, there is ample scope for improving the performance of the classifiers or to design good classifiers to handle more complex datasets and to gain more accuracy.

As of now various statistical and soft computing based classifiers have been proposed in the literature. The traditional statistical techniques like Euclidean minimum distance (EMD), quadratic minimum distance (QMD) and k-nearest neighbor (KNN) classifiers [1] and Bayesian decision theory [13] are used to build different classifiers. One of the drawbacks of the statistical method [13,14] is that it depends on the correctness of the underlying assumptions for its successful application. Unlike soft computing methods, in statistical method user needs to have thorough grasp over the properties of dataset for successful application of the model. However, it is not always possible. As a result statistical technique based classifiers generally give less accuracy as compared to the soft computing based classifiers.

Multilayer perceptron (MLP) [15], radial basis function neural networks (RBFNNs) [16], fuzzy rule based systems [17], adaptive neuro-fuzzy systems (ANFIS) [18], support vector machine (SVM) [19,20], k-nearest neighbor classifier (KNNs) [21], Naive Bayes classifier [22], polynomial classifiers [23], CART [24], decision tree classifier [25] and random forest [26] have been used as classifiers in many applications. However, amongst all the classifiers, ANN has been chosen by the researchers most frequently and extensively. While considering ANN as a classifier, the major things that

* Corresponding author. Tel.: +91 674 2727336.

E-mail address: pkdash.india@gmail.com (P.K. Dash).

**Fig. 1.** Architecture of SLFN model.

are taken into consideration are the number of hidden layers, the values of the weights between input to hidden layer, hidden to output layer and the learning algorithms [27] i.e. backpropagation (BP), recursive least square (RLS) and different evolutionary learning algorithms. So the performance of ANN as a classifier mostly depends on the right combination of the structures and learning algorithms. However, the major disadvantage of ANN is considered to be its association with gradient descent learning algorithm that makes the performance of the model time consuming and increases the computational overhead [16]. In the gradient descent learning algorithm, due to the initial random choice of parameters, the convergence rate becomes very slow and most often it gets trapped in the local minima. To avoid the above said limitations, Huang et al. proposed a novel learning algorithm i.e. extreme learning machine (ELM) [11]. Researchers have shown that ELM owes its origin to random vector functional link (RVFL) [28–35]. However Huang [36] has shown its newness in Ref. [36]. The two major advantages of ELM are faster learning speed and good generalization ability. Literature survey reveals that ELM has been extensively used in many applications [37–43]. Although several variants of extreme learning machines [28–43] are now available for multiclass classification there remains several problems like the optimal choice of network size requiring a large number of hidden nodes for better generalization and choice of activation functions. Besides the randomness of ELM causes an additional uncertainty in regression and classification problems with regards to universal approximation and learning.

In the last seven years, a lot of research has already been done using ELM in the field of designing filters in image applications [44], sales forecasting [45], time series prediction [46], power system economic dispatch [47], electricity load forecasting [48,49], target recognition, aircraft recognition, clustering [50], real time fault diagnosis, end point prediction model, neural architecture design [51], disease diagnosis [52], mobility prediction in mobile ad-hoc networks, corporate life cycle prediction, system identification, breast tumor detection, etc. Especially for classification problems, ELM has been successfully employed in the various fields of classification like gene expression classification [53], binary class and multiclass data classification [11,54].

In ELM, the input weights and hidden biases are chosen randomly from which the output weights are computed. During this process, ELM tries to minimize the training error and determine the smallest norm of the output weights. Due to the random choice of the input weights and biases in ELM, many a times the output matrix does not show full column rank and this leads to ill-conditioning [55] of the system that generates non-optimal solutions. So, to improve the conditioning of ELM and to ensure the optimal solutions, evolutionary ELM [30,56–62] is used. The evolutionary ELM not only gives better accuracy but also ensures system stability. Amongst the evolutionary learning algorithms,

1	67
2	68
3	69
4	70
5	71
6	72
7	73
8	74
9	75
10	76
11	77
12	78
13	79
14	80
15	81
16	82
17	83
18	84
19	85
20	86
21	87
22	88
23	89
24	90
25	91
26	92
27	93
28	94
29	95
30	96
31	97
32	98
33	99
34	100
35	101
36	102
37	103
38	104
39	105
40	106
41	107
42	108
43	109
44	110
45	111
46	112
47	113
48	114
49	115
50	116
51	117
52	118
53	119
54	120
55	121
56	122
57	123
58	124
59	125
60	126
61	127
62	128
63	129
64	130
65	131
66	132

Table 1
Specification of the datasets used for classification.

Dataset name	#Trg samples	#Tst samples	#Features	#Classes	Random permutation
Breast Cancer	499	200	9	2	Yes
Diabetes	576	192	8	2	Yes
Bupa	199	146	6	2	Yes
Hepatitis	80	75	19	2	Yes

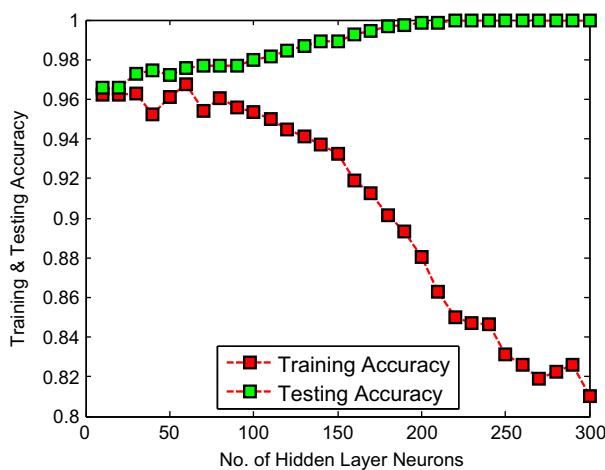
GA based ELM [27], PSO based ELM [57,58,60], DE based ELM [61], group search based ELM [62] have already been used in the literature. In recent years meta-heuristic algorithms have been successfully applied for solving practical and difficult optimization problems.

Thus in this paper, the recently developed meta-heuristic algorithm i.e. cuckoo search (CS) proposed by Yang and Deb [63–68] is used to pre-train the ELM ensuring optimal solutions. This algorithm is based on the combination of breeding behavior of cuckoos with the Lévy flight seen in some species of birds [66]. The CS algorithm has been developed to provide better performance than other established meta-heuristic algorithms. Civicioglu [68] proves that CSA yields better and robust solution as compared to particle swarm optimization (PSO), differential evolution (DE) and artificial bee colony (ABC). Again, to further improve the accuracy and stability of cuckoo search based extreme learning algorithm (CSELM), the improved cuckoo search algorithm [69,70] is combined with ELM i.e. improved cuckoo search extreme learning machine (ICSELM) model is proposed and experimented to classify four binary class datasets, Breast Cancer, Diabetes, Bupa and Hepatitis. Both CSELM and ICSELM choose the input weights and biases before calculating the output weights and they ensure the full column rank of the hidden layer output matrix. A fast batch learning algorithm i.e. online sequential ELM (OSELM) [71–73] and other two ANN based models, MLP and RBFNN with BP, CS and ICS learning algorithms are also taken to classify the above said binary datasets and compared with the proposed model. The performance of all the classifiers discussed in this paper is measured using various performance evaluation measures like Overall Accuracy, sensitivity, specificity, confusion matrix, Gmean, F-score and ROC analysis [54,74]. The system complexity for CSELM and ICSELM are found to be $O(sj^2)$ where 's' represents the number of samples and 'f' represents the number of features in the dataset.

The paper is organized as follows: The ELM and OSELM algorithms are briefly introduced in Section 2. The cuckoo search (CS), improved cuckoo search (ICS) and cuckoo search based algorithms such as CSELM and ICSELM are explained in Section 3. Section 4 introduces all the benchmark datasets. All the performance evaluation measures used in this study are discussed

1
2
3
4
Table 2
5 Confusion matrix.
6
7

	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN



27 Fig. 2. Training and testing accuracy (ELM).
28
29

30 in Section 5. Experimental outputs and results are given in Section
31 6 whereas analysis of the results is presented in Section 7. Finally,
32 conclusions are given in Section 8. An outline of MLP and RBFNN
33 are given in Appendix A and B respectively.

35 36 2. The outline of extreme learning machine

37 This section discusses both extreme learning machine (ELM)
38 and online sequential extreme learning machine (OSELML) used in
39 this study.

43 44 2.1. Extreme learning machine

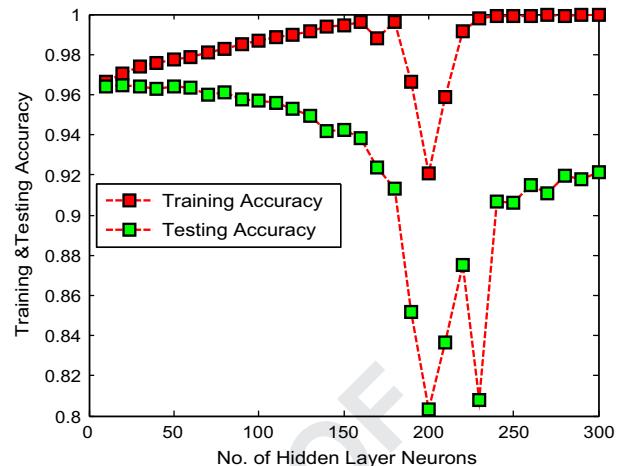
45 46 47 48 49 50 51 52 53 54 55 Extreme learning machine proposed by G. B. Huang [28,29] is
56 designed as the most advanced learning algorithm to train SLFN i.e.
57 single hidden layer feed forward network. It has become one of the
58 important conceptual and computational paradigms for its thousand
59 times faster learning speed in comparison to simple neural networks
60 and better generalization performance than the traditional gradient
61 based learning algorithms. ELM is computationally free from iterations
62 that make ELM extremely fast by significantly reducing the
63 computational time required to train the SLFNs. The SLFN model
64 often requires high number of hidden neurons while generating
65 optimal solutions. The architecture of SLFN is given in Fig. 1.

66 The output function of the single hidden layer feed forward
67 networks (SLFN) with L hidden nodes can be represented by the
68 following equations:

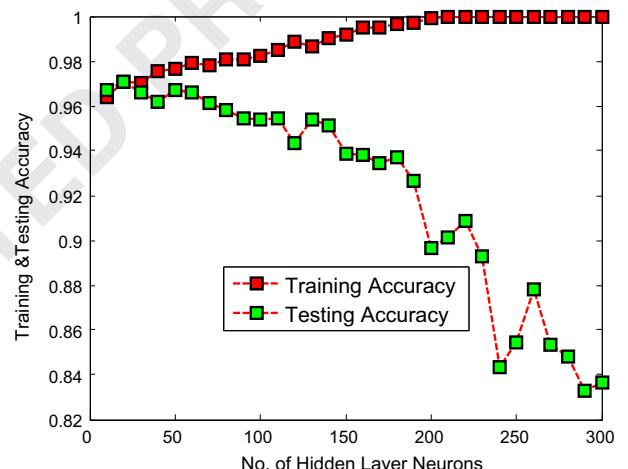
$$f_L(x) = \sum_{i=1}^L \beta_i g_i x = \sum_{i=1}^L \beta_i G(a_i, b_i, x), x \in R^d, \beta_i \in R^m \quad (1)$$

69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 For additive nodes with activation function g , g is defined as

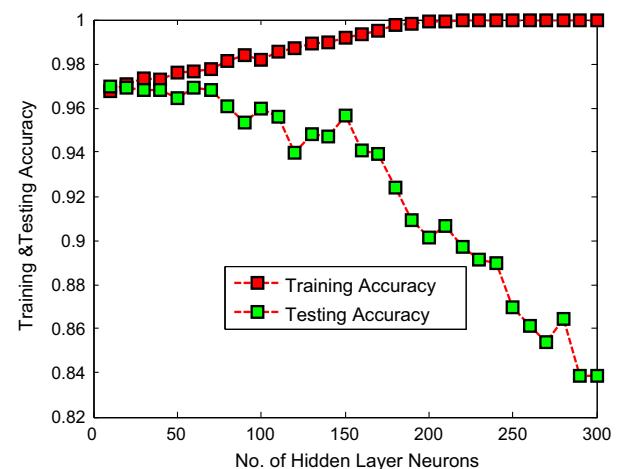
$$g_i = G(a_i, b_i, x) = g(a_i x + b_i), a_i \in R^d, b_i \in R \quad (2)$$



33 Fig. 3. Training and testing accuracy (OSELML).



34 Fig. 4. Training and testing accuracy (CSELM).



35 Fig. 5. Training and testing accuracy (ICSELM).

$$\sum_{i=1}^L \beta_i G(a_i, b_i, x) = t_j, j = 1, \dots, N \quad (3)$$

36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 The above equation can be rewritten as

$$H\beta = T \quad (4)$$

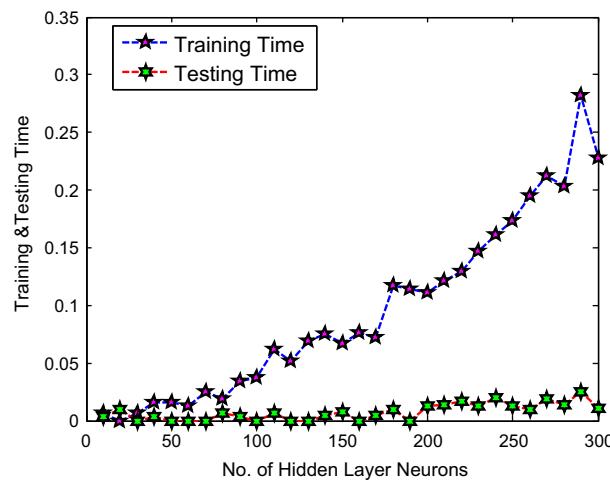


Fig. 6. Training and testing time (ELM).

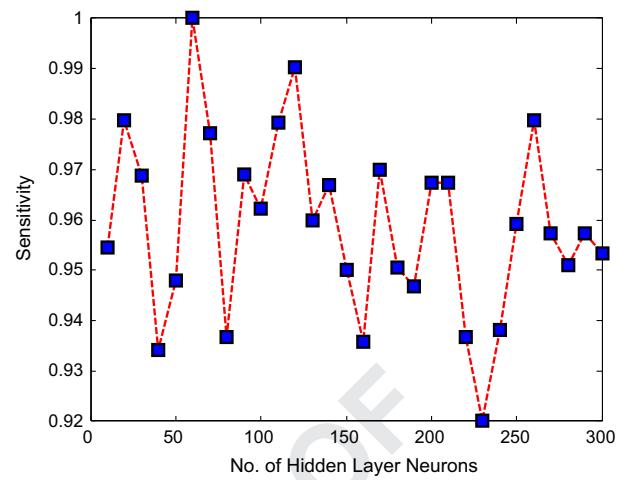


Fig. 9. Sensitivity (ICSELM).

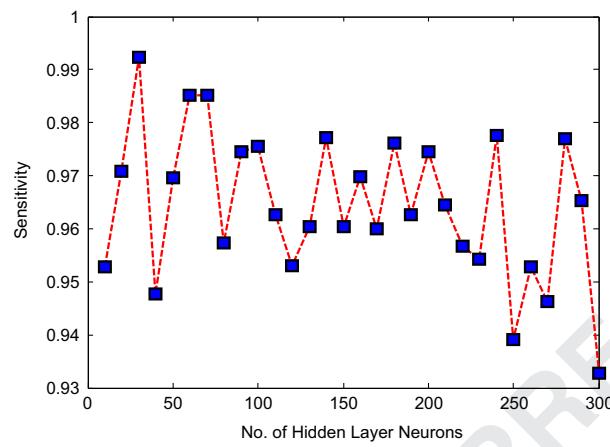


Fig. 7. Sensitivity (ELM).

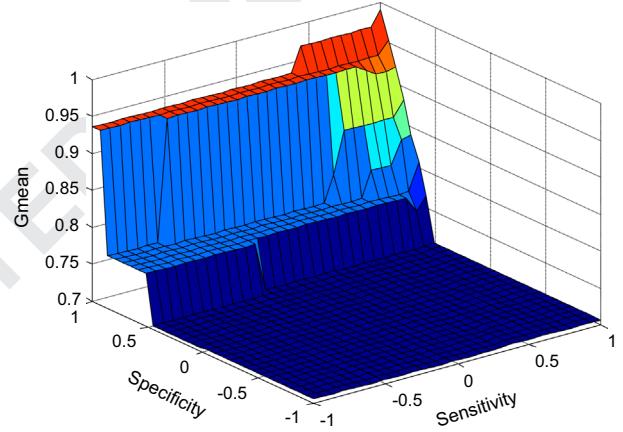


Fig. 10. Sensitivity vs Specificity vs Gmean (ELM).

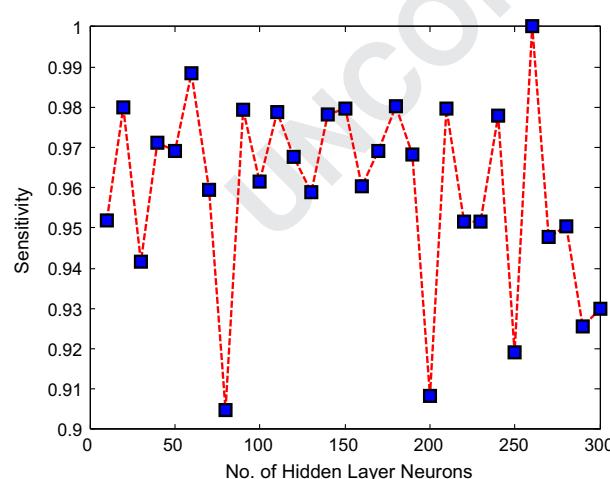


Fig. 8. Sensitivity (CSELM).

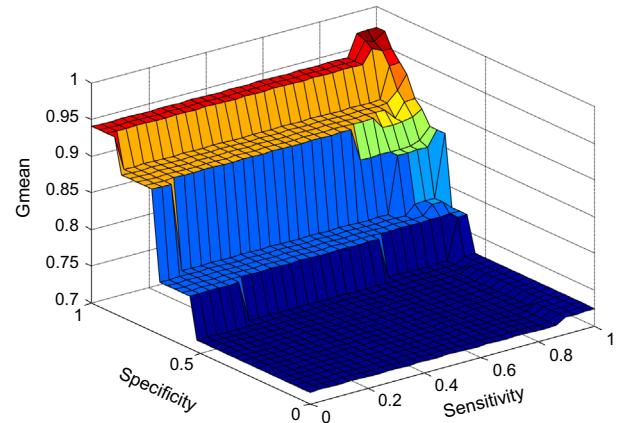


Fig. 11. Sensitivity vs Specificity vs Gmean (CSELM).

here

$$H(w_1 \dots w_L, b_1 \dots b_L, x_1 \dots x_N)$$

$$= \begin{pmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_L \cdot x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_L \cdot x_N + b_L) \end{pmatrix} \quad (5)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix} \quad (6)$$

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix} \quad (7)$$

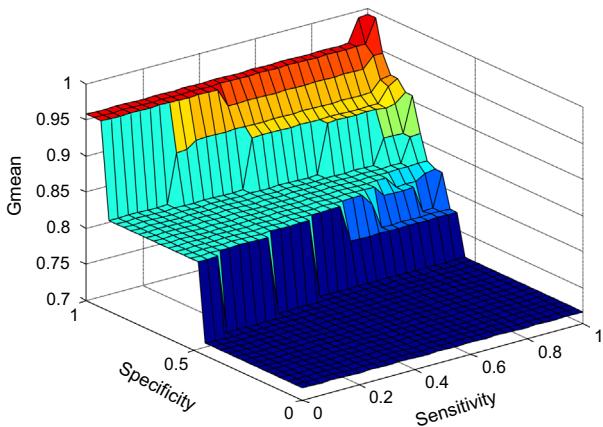


Fig. 12. Sensitivity vs Specificity vs Gmean (ICSELM).

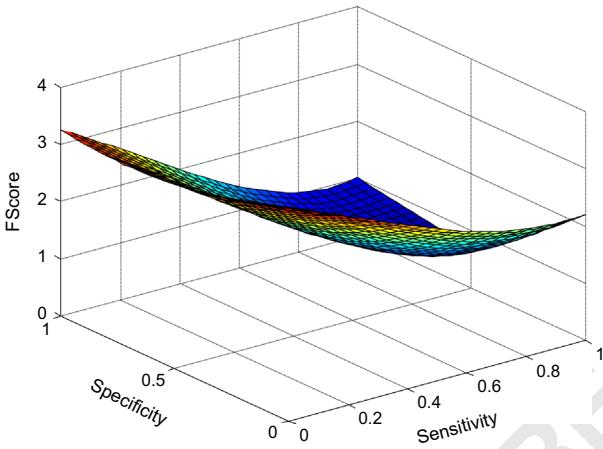


Fig. 13. Sensitivity vs Specificity vs Fscore (ICSELM).

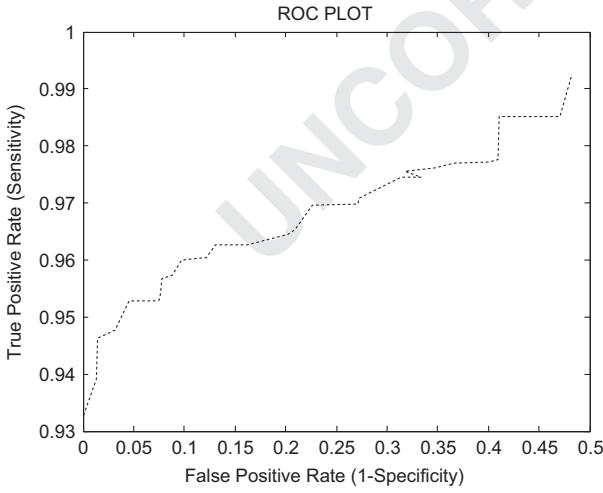


Fig. 14. ROC Plot (ELM).

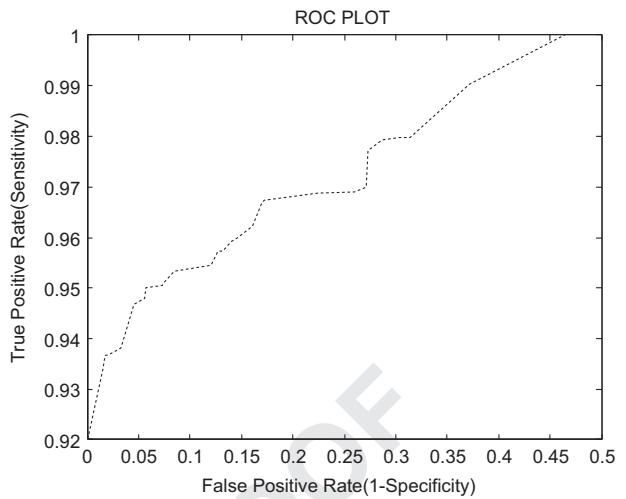


Fig. 15. ROC Plot (CSELM).

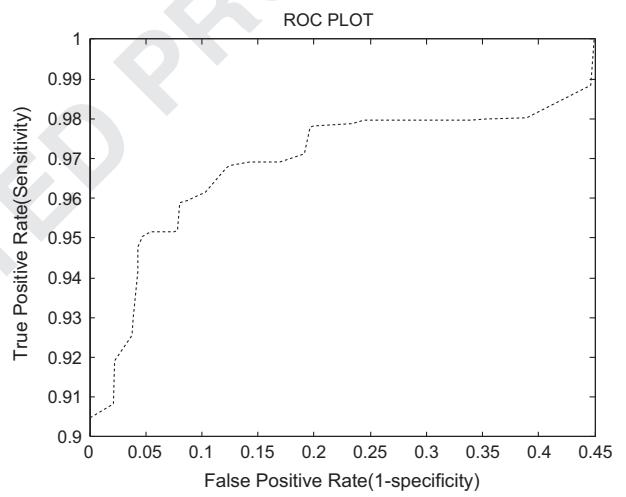


Fig. 16. ROC Plot (ICSELM).

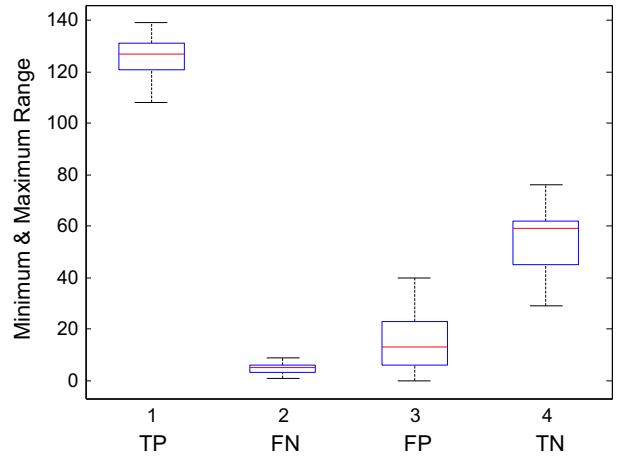


Fig. 17. Box Plot (ELM).

H represents the hidden layer output matrix of the neural network. Huang et al. [28] have proved that given an arbitrary small value $\epsilon > 0$, if the hidden layer activation function is infinitely differentiable, the number of hidden neurons $L \leq N$, the input weights and hidden layer biases can be evaluated, then SLFN can be trained to solve a linear system optimization problem using the following equation:

$$\|H\hat{\beta} - T\| = \min_{\beta} \|H\beta - T\| \quad (8)$$

here $\hat{\beta}$ can be denoted as

$$\hat{\beta} = H^T T = (H^T H)^{-1} H^T T \quad (9)$$

is the smallest norm least square solution of $w_i = T$ and H^T represents the Moore Penrose generalized inverse of H .

The procedure of ELM can be summarized in the following three steps:

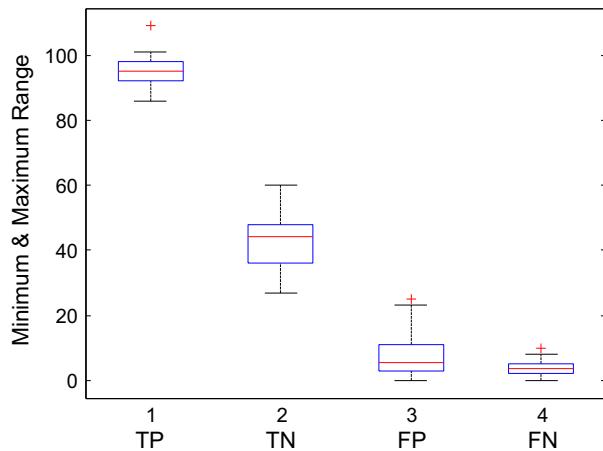


Fig. 18. Box Plot (CSELM).

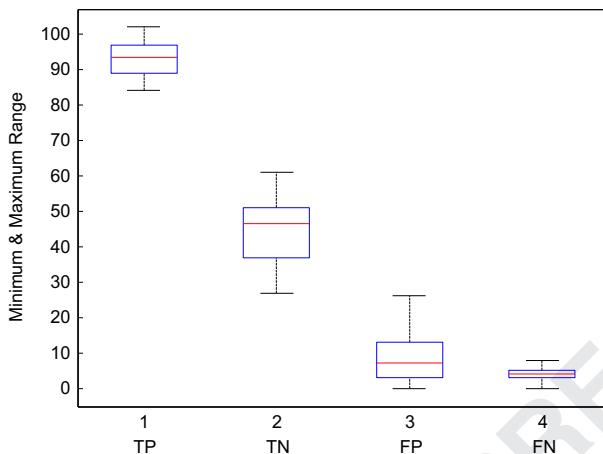


Fig. 19. Box Plot (ICSELM).

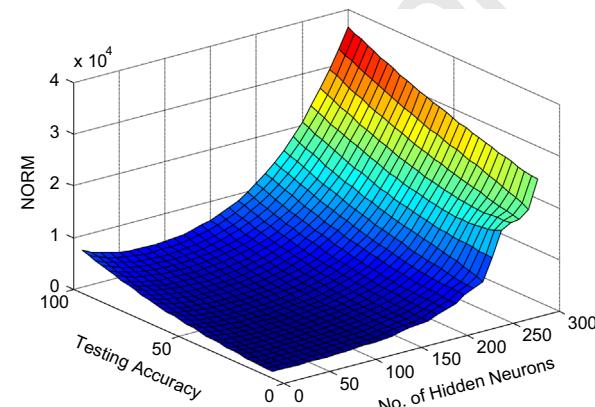


Fig. 20. No. of hidden neurons vs testing accuracy vs norm (ELM).

Step 1 Choose the input weights w_i and hidden layer biases b_i randomly.

Step 2 Compute the hidden layer output matrix H.

Step 3 Obtain the output weight $\hat{\beta}$ using the equation $\hat{\beta} = H^T T$.

2.2. Online sequential extreme learning machine (OSELM)

The online sequential extreme learning machine (OSELM) is an extremely fast batch learning algorithm and can provide better generalization performance [71]. Here the OSELML can learn the training data one-by-one or chunk-by-chunk. The data that have

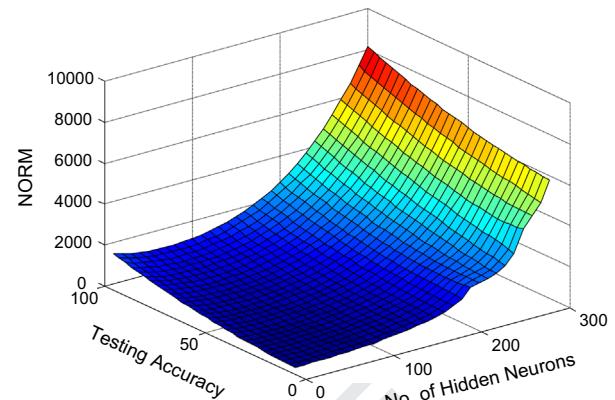


Fig. 21. No. of hidden neurons vs testing accuracy vs norm (CSELM).

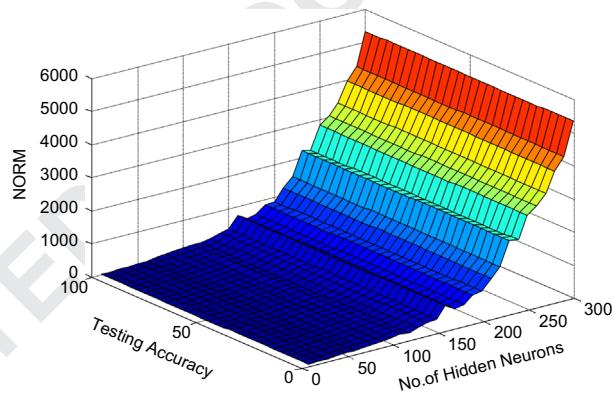


Fig. 22. No. of hidden neurons vs testing accuracy vs norm (ICSELM).

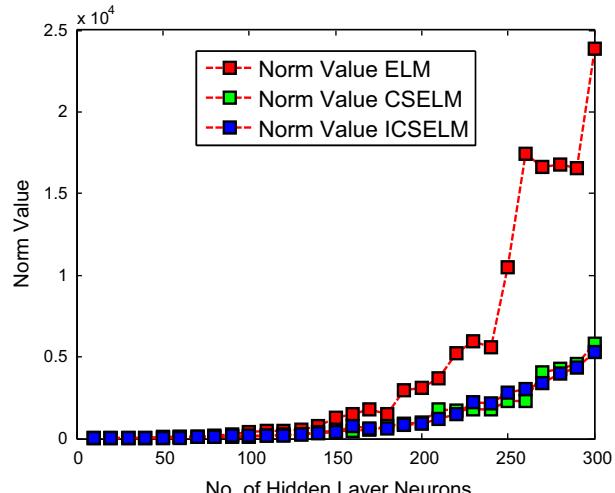


Fig. 23. Comparison of norm value (ELM, CSELM, ICSELM).

already been used in the training can be discarded to save more memory and computational load to process the newly incoming data. Though a lot of variations of ELM have been used extensively in many applications, it is being used here as a classifier in order to exploit the batch learning method of OSELML.

The online sequential learning in OS-ELM consists of two steps.

Step 1 Initialization

In the step 1, a small portion of training data $n_0 = (x_i, y_i), i=1, \dots, N_0$ with $N_0 \in N$ is considered for initializing the learning process. The initial output weight matrix is calculated

Table 3

Training time (Trg Time), training accuracy (Trg Acc), testing accuracy (Tst Acc) and standard deviation of testing accuracy (Tst Dev) for ELM, OSELM, CSELM and ICSELM (Breast Cancer dataset).

L	ELM				OSELM				CSELM				ICSELM		
	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	
10	.0062	.9625	.9657	.0069	.0106	.9664	.9638	.0108	.9641	.9673	.0135	.9675	.9700	.0079	
20	0	.9625	.9659	.0039	.0147	.9707	.9649	.0107	.9707	.9707	.0114	.9709	.9693	.0123	
30	.0062	.9630	.9727	.0060	.0223	.9738	.9641	.0103	.9705	.9660	.0152	.9733	.9680	.0133	
40	.0156	.9525	.9745	.0067	.0267	.9758	.9628	.0114	.9756	.9620	.0148	.9731	.9680	.0103	
50	.0156	.9610	.9721	.0039	.0363	.9776	.9641	.0118	.9766	.9673	.0192	.9758	.9647	.0114	
60	.0125	.9675	.9758	.0046	.0415	.9789	.9636	.0110	.9790	.9660	.0097	.9766	.9693	.0202	
70	.0250	.9540	.9770	.0045	.0466	.9812	.9600	.0137	.9784	.9613	.0166	.9778	.9680	.0157	
80	.0187	.9605	.9768	.0038	.0593	.9829	.9611	.0122	.9806	.9580	.0178	.9812	.9607	.0162	
90	.0343	.9560	.9768	.0019	.0710	.9853	.9575	.0117	.9810	.9547	.0180	.9842	.9533	.0189	
100	.0374	.9535	.9798	.0038	.0800	.9867	.9569	.0117	.9826	.9540	.0159	.9816	.9600	.0181	
110	.0624	.9500	.9816	.0052	.0952	.9884	.9558	.0127	.9850	.9547	.0217	.9854	.9560	.0126	
120	.0515	.9450	.9844	.0031	.1200	.9899	.9529	.0139	.9888	.9433	.0155	.9870	.9400	.0213	
130	.0686	.9415	.9866	.0027	.1323	.9916	.9494	.0132	.9868	.9540	.0182	.9890	.9480	.0177	
140	.0749	.9370	.9890	.0025	.1494	.9940	.9421	.0147	.9902	.9513	.0137	.9898	.9473	.0200	
150	.0671	.9325	.9892	.0034	.1697	.9947	.9422	.0147	.9918	.9387	.0093	.9918	.9567	.0123	
160	.0764	.9190	.9928	.0046	.1705	.9962	.9384	.0168	.9948	.9380	.0160	.9934	.9407	.0168	
170	.0718	.9125	.9942	.0027	.2039	.9880	.9237	.0667	.9952	.9347	.0177	.9948	.9393	.0173	
180	.1170	.9015	.9968	.0032	.2264	.9960	.9130	.0290	.9966	.9373	.0173	.9974	.9240	.0242	
190	.1139	.8930	.9976	.0018	.2407	.9663	.8519	.0778	.9970	.9267	.0183	.9980	.9093	.0240	
200	.1108	.8805	.9986	.0014	.2602	.9208	.8031	.0721	.9994	.8967	.0158	.9990	.9013	.0259	
210	.1217	.8630	.9988	.0014	.3162	.9587	.8363	.0535	.9998	.9013	.0213	.9992	.9067	.0163	
220	.1295	.8500	.9996	8.4496e–04	.3312	.9916	.8751	.0351	.9998	.9087	.0274	1	.8973	.0265	
230	.1466	.8470	1	0	.3671	.9979	.8079	.0258	1	.8927	.0378	1	.8913	.0351	
240	.1607	.8465	1	0	.3671	.9991	.9068	.0248	1	.8433	.0307	1	.8900	.0327	
250	.1732	.8315	1	0	.4204	.9992	.9061	.0228	1	.8547	.0347	1	.8700	.0216	
260	.1950	.8260	1	0	.4608	.9993	.9150	.0235	1	.8780	.0292	1	.8613	.0275	
270	.2122	.8190	1	0	.4833	.9995	.9108	.0182	1	.8533	.0322	1	.8540	.0378	
280	.2028	.8225	1	0	.5086	.9994	.9196	.0191	1	.8480	.0344	1	.8647	.0298	
290	.2824	.8260	1	0	.5569	.9995	.9181	.0201	1	.8327	.0258	1	.8387	.0243	
300	.2278	.8105	1	0	.5808	.9995	.9215	.0179	1	.8367	.0178	1	.8387	.0255	

Table 4

Sensitivity (Sen.), norm, Gmean and F-score (Fscore) for ELM, CSELM and ICSELM (Breast Cancer dataset).

L	ELM				CS ELM				ICS ELM			
	Sens.	Norm	Gmean	Fscore	Sens.	Norm	Gmean	Fscore	Sens.	Norm	Gmean	Fscore
10	.9528	24.2016	.9694	.9692	.9519	14.6714	.9650	.9649	.9545	13.8719	.9691	.9690
20	.9708	22.7002	.9695	.9695	.9800	24.5555	.9495	.9491	.9798	25.3162	.9898	.9898
30	.9924	37.5520	.9889	.9888	.9417	34.2265	.9704	.9700	.9688	32.2676	.9471	.9469
40	.9478	59.7537	.9359	.9359	.9712	43.8248	.9638	.9638	.9341	44.0108	.9582	.9579
50	.9697	114.4018	.9403	.9398	.9691	46.4784	.9368	.9363	.9479	72.4240	.9462	.9462
60	.9851	118.5545	.9697	.9696	.9885	75.5334	.9703	.9701	1	67.4016	.9909	.9908
70	.9852	126.4956	.9536	.9531	.9596	89.1168	.9602	.9602	.9773	105.2625	.9725	.9725
80	.9574	176.8664	.9785	.9783	.9048	111.0089	.9406	.9399	.9368	134.4422	.9320	.9320
90	.9746	222.5733	.9504	.9501	.9794	134.8042	.9516	.9512	.9691	150.2590	.9561	.9561
100	.9756	392.6240	.9485	.9481	.9615	140.8422	.9590	.9590	.9623	182.0169	.9584	.9584
110	.9626	439.0646	.9206	.9197	.9789	205.7366	.9621	.9619	.9794	219.1995	.9220	.9203
120	.9531	459.9965	.9276	.9273	.9677	290.4036	.9396	.9392	.9903	192.2514	.9065	.9030
130	.9603	550.0730	.8970	.8949	.9588	268.4945	.9605	.9605	.9600	270.0195	.9191	.9183
140	.9771	787.6442	.9218	.9202	.9783	378.1279	.9365	.9356	.9670	362.8081	.9408	.9404
150	.9603	1.2502e+003	.9184	.9175	.9798	448.2985	.9502	.9498	.9500	401.5255	.9649	.9648
160	.9699	1.4859e+003	.8759	.8714	.9604	438.9843	.9073	.9058	.9358	734.9133	.8938	.8928
170	.9600	1.7676e+003	.8616	.8566	.9691	565.9789	.8969	.8943	.9700	586.1422	.9133	.9117
180	.9762	1.5195e+003	.8822	.8777	.9804	761.1709	.9262	.9247	.9505	636.4936	.8586	.8541
190	.9627	2.9908e+003	.8367	.8286	.9684	923.0175	.8802	.8762	.9468	855.1685	.8914	.8898
200	.9746	3.1204e+003	.7861	.7683	.9083	993.7104	.8930	.8929	.9674	889.5885	.8948	.8920
210	.9645	3.6957e+003	.8384	.8303	.9798	1.8205e+003	.8875	.8832	.9674	1.2182e+003	.8948	.8920
220	.9568	5.2132e+003	.7515	.7300	.9515	1.6820e+003	.8771	.8742	.9368	1.4992e+003	.9042	.9036
230	.9542	5.9874e+003	.7976	.7849	.9515	1.7668e+003	.8537	.8487	.9200	2.2166e+003	.8251	.8202
240	.9776	5.6028e+003	.8073	.7927	.9780	1.8171e+003	.7725	.7515	.9381	2.1607e+003	.8093	.8005
250	.9391	1.0477e+004	.7051	.6771	.9192	2.2677e+003	.7712	.7595	.9592	2.8334e+003	.8261	.8170
260	.9528	1.7462e+004	.7491	.7280	1	2.2738e+003	.8695	.8611	.9798	3.0565e+003	.8200	.8072
270	.9462	1.6661e+004	.8055	.7952	.9479	4.0650e+003	.8484	.8432	.9574	3.3746e+003	.7162	.6870
280	.9769	1.6780e+004	.7656	.7434	.9505	4.2619e+003	.7237	.6976	.9510	3.9715e+003	.8327	.8254
290	.9653	1.6573e+004	.7070	.6741	.9255	4.5507e+003	.7158	.6928	.9574	4.3354e+003	.8166	.8063
300	.9328	2.3875e+004	.7705	.7566	.9300	5.8035e+003	.7835	.7721	.9533	5.3302e+003	.7737	.7571

Table 5

Specificity (Spe.), TP, FN, FP, TN for ELM, CSELM and ICSELM (Breast Cancer dataset).

L	ELM					CSELM					ICSELM				
	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN
10	.9863	121	6	1	72	.9783	99	5	1	45	.9839	84	4	1	61
20	.9683	133	4	2	61	.9200	98	2	4	46	1	97	2	0	51
30	.9853	131	1	1	67	1	97	6	0	47	.9259	93	3	4	50
40	.9242	127	7	5	61	.9565	101	3	2	44	.9831	85	6	1	58
50	.9118	128	4	6	62	.9057	94	3	5	48	.9444	91	5	3	51
60	.9545	132	2	3	63	.9524	86	1	3	60	.9818	95	0	1	54
70	.9231	133	2	5	60	.9608	95	4	2	49	.9677	86	2	2	60
80	1	135	6	0	59	.9778	95	10	1	44	.9273	89	6	4	51
90	.9268	115	3	6	76	.9245	95	2	4	49	.9434	94	3	3	50
100	.9221	120	3	6	71	.9565	100	4	2	44	.9545	102	4	2	42
110	.8806	128	5	8	59	.9455	93	2	3	52	.8679	95	2	7	46
120	.9028	122	6	7	65	.9123	90	3	5	52	.8298	102	1	8	39
130	.8378	121	5	12	62	.9623	93	4	2	51	.8800	96	4	6	44
140	.8696	128	3	9	60	.8966	90	2	6	52	.9153	88	3	5	54
150	.8784	121	5	9	65	.9216	97	2	4	47	.9800	95	5	1	49
160	.7910	129	4	14	53	.8571	97	4	7	42	.8537	102	7	6	35
170	.7733	120	5	17	58	.8302	94	3	9	44	.8600	97	3	7	43
180	.7973	123	3	15	59	.8750	100	2	6	42	.7755	96	5	11	38
190	.7273	129	5	18	48	.8000	92	3	11	44	.8393	89	5	9	47
200	.6341	115	3	30	52	.8780	99	10	5	36	.8276	89	3	10	48
210	.7288	136	5	16	43	.8039	97	2	10	41	.8276	89	3	10	48
220	.5902	133	6	25	36	.8085	98	5	9	38	.8727	89	6	7	48
230	.6667	125	6	23	46	.7660	98	5	11	36	.7400	92	8	13	37
240	.6667	131	3	22	44	.6102	89	2	23	36	.6981	91	6	16	37
250	.5294	108	7	40	45	.6471	91	8	18	33	.7115	94	4	15	37
260	.5890	121	6	30	43	.7561	109	0	10	31	.6863	97	2	16	35
270	.6857	123	7	22	48	.7593	91	5	13	41	.5357	90	4	26	30
280	.6000	127	3	28	42	.5510	96	5	22	27	.7292	97	5	13	35
290	.5179	139	5	27	29	.5536	87	7	25	31	.6964	90	4	17	39
300	.6364	125	9	24	42	.6600	93	7	17	33	.6279	102	5	16	27

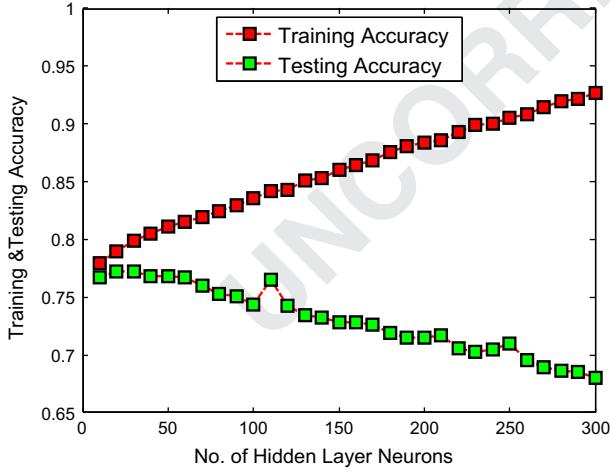


Fig. 24. Training and testing accuracy (ELM).

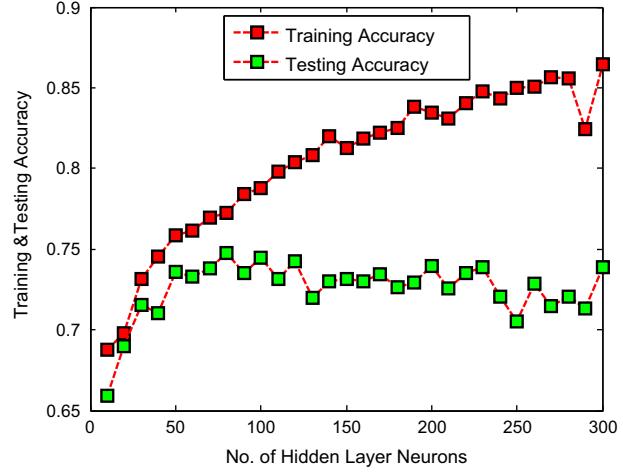


Fig. 25. Training and testing accuracy (OSELM).

according to the ELM algorithm by randomly assigning the input weights w_j and bias b_j , $j=1, 2, \dots, L$ as follows:

$$\beta^{(0)} = Q_0 H_0^T T^0 \quad (10)$$

where

$$Q_0 = (H_0^T H_0)^{-1} \quad (11)$$

and H^0 is represented as the initial hidden layer output matrix.

Step 2 Sequential learning

When the new set of observations arrive $n_{k+1} = (x_i, y_i)$,

$$Y_{k+1} = [y(\sum_{l=0}^k N_l) + 1, \dots, y(\sum_{l=0}^{k+1} N_l)]^T \quad (12)$$

i.e. $(k+1)$ th chunk of data, first compute the partial hidden layer output matrix H_{k+1} , N_{k+1} is the number of samples in the $(k+1)$ th chunk. Then by using the output weight update equation as shown below, the output weight matrix $\beta^{(k+1)}$ will be computed. $Y_{k+1} = [y(\sum_{l=0}^k N_l) + 1, \dots, y(\sum_{l=0}^{k+1} N_l)]^T$ can be calculated

$$Q_{k+1} = Q_k - Q_k H_{k+1}^T (1 + H_{k+1} Q_k H_{k+1}^T)^{-1} H_{k+1} Q_k \quad (13)$$

$$\beta^{(k+1)} = \beta^k + Q_{k+1} H_{k+1} (T_{k+1} - H_{k+1} \beta^k) \quad (14)$$

The above equations are analogous to the recursive least square algorithm. Each time a new chunk of data arrives, the output weight matrix is updated according to (13) and (14).

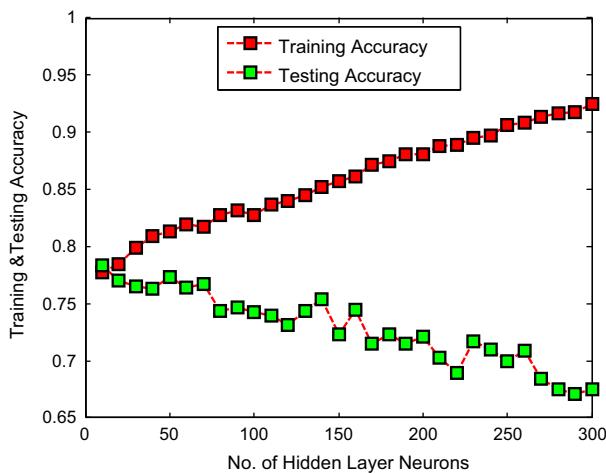


Fig. 26. Training and testing accuracy (CSELM).

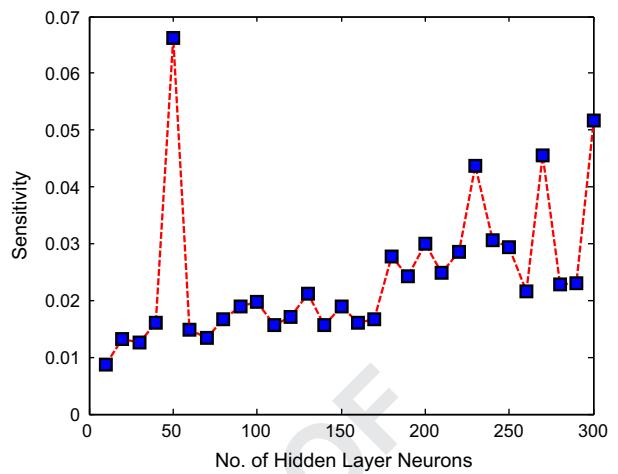


Fig. 29. Sensitivity (ELM).

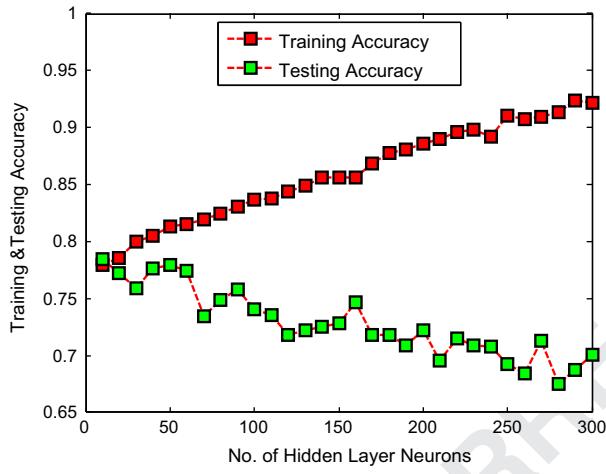


Fig. 27. Training and testing accuracy (ICSELM).

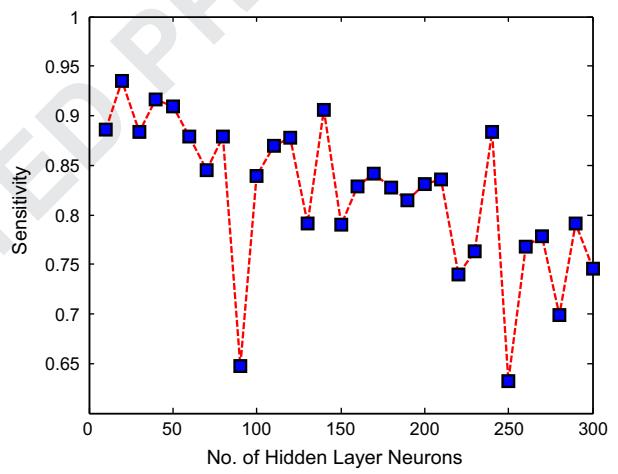


Fig. 30. Sensitivity (CSELM).

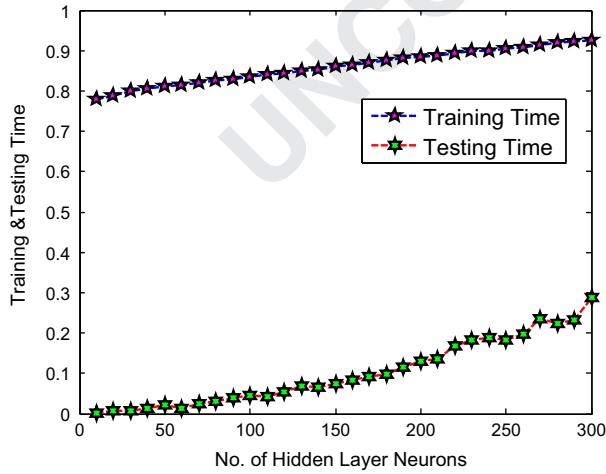


Fig. 28. Training and testing time (ELM).

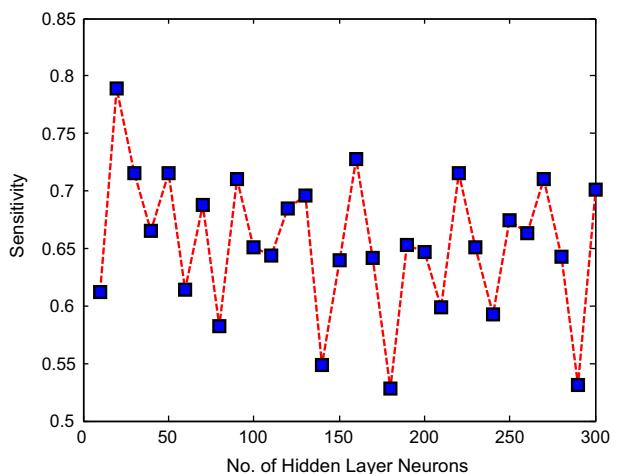


Fig. 31. Sensitivity (ICSELM).

The updated Eqs. (13) and (14) are for both chunk-by-chunk and one-by-one learning as one-by-one can be considered as a special case of chunk-by-chunk where the chunk size=1.

Other ANN based models i.e., MLP [15] and RBFNN [16] are implemented using both backpropagation and evolutionary learning algorithm such as CS and ICS for all four datasets. Both models are briefly discussed in Appendix A and B respectively.

3. Features of cuckoo search and cuckoo search based algorithms

In this section, standard cuckoo search, improved cuckoo search, cuckoo search extreme learning machine and improved cuckoo search extreme learning machine are discussed.

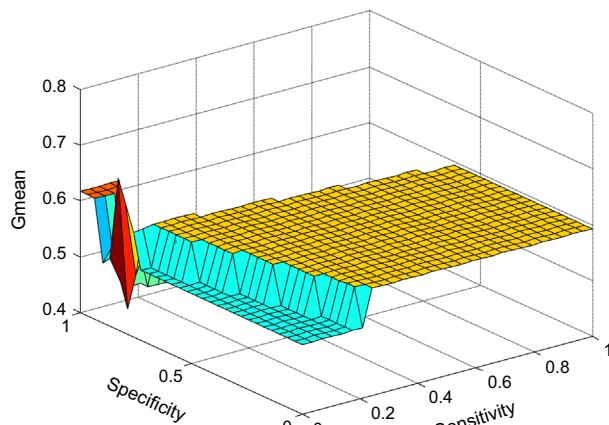


Fig. 32. Sensitivity vs Specificity vs Gmean (ELM).

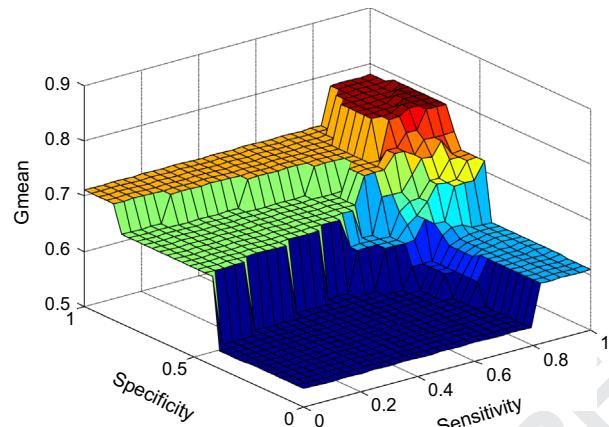


Fig. 33. Sensitivity vs Specificity vs Gmean (CSELM).

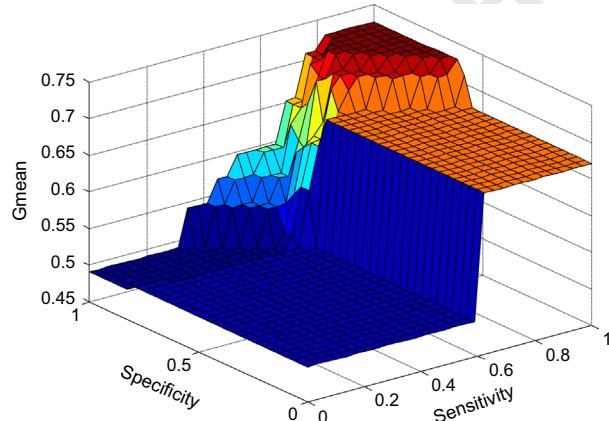


Fig. 34. Sensitivity vs Specificity vs Gmean (ICSELM).

3.1. Cuckoo search algorithm

Evolutionary algorithms such as genetic algorithm, genetic programming, differential evolution, particle swarm optimization, ant colony optimization algorithm are seen to be more successful than the conventional methods in handling complex engineering optimization problems. However, the meta-heuristic algorithm, cuckoo search (CS) proposed by Yang and Deb in 2010 [64] has been proved to give far better solutions than the best solutions obtained from PSO, GA, DE and ABC [68]. The CS algorithm is inspired by the behavior of parasitic cuckoo that chooses a nest where the host bird has just laid its own

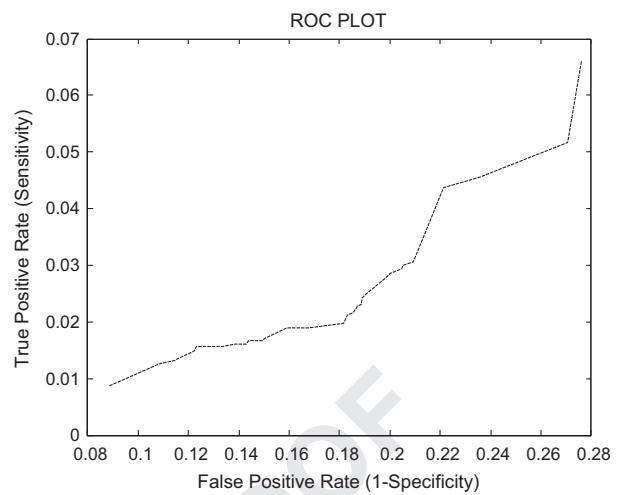


Fig. 35. ROC (ELM).

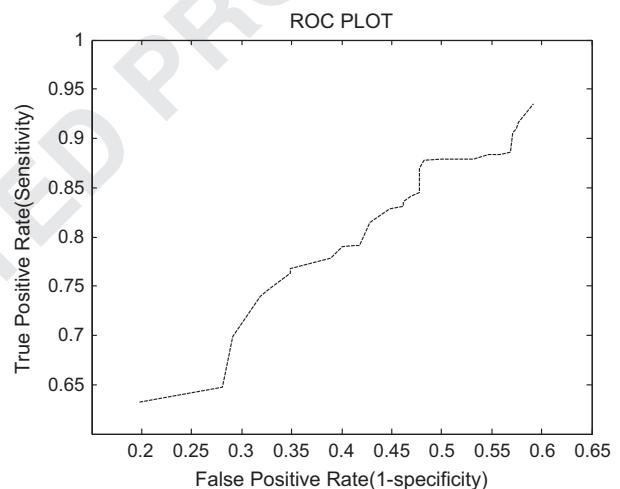


Fig. 36. ROC (CSELM).

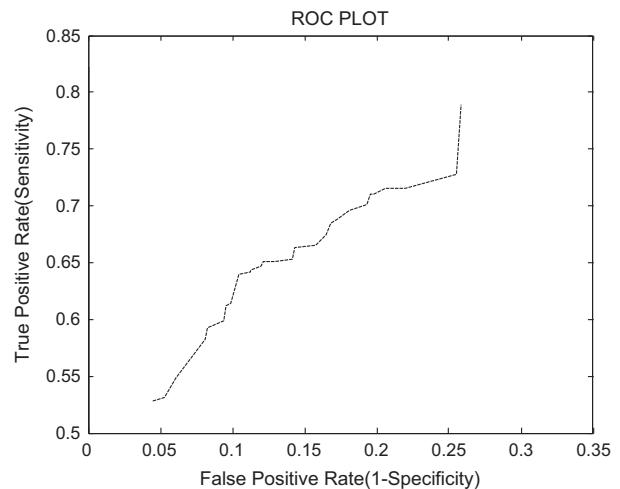


Fig. 37. ROC (ICSELM).

eggs. Generally eggs of cuckoo hatch slightly earlier than their host eggs. As soon as the first cuckoo chick is hatched, the foremost action it takes is to evict the host eggs by blindly propelling the eggs out of the nest in order to increase the cuckoo chick's share of food provided by its host bird. Apart from propelling the host bird's eggs from the nest, the cuckoo chick also mimics the call of the host chicks to access

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

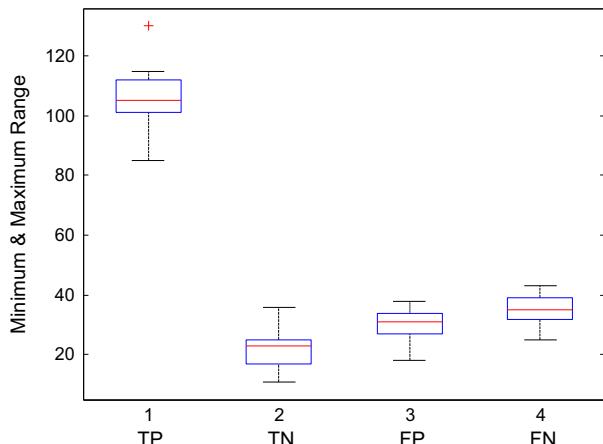


Fig. 38. Box Plot (ELM).

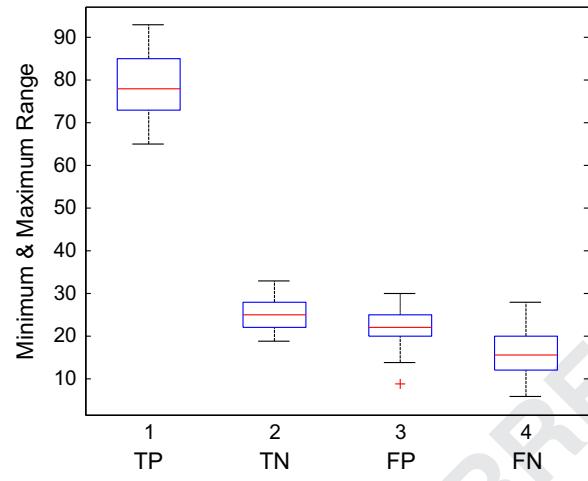


Fig. 39. Box Plot (CSELM).

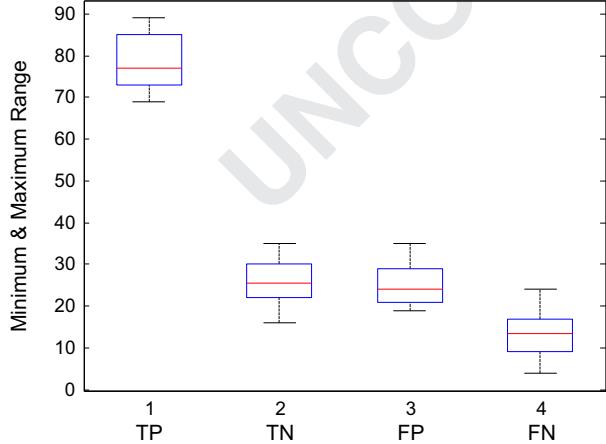


Fig. 40. Box Plot (ICSELM).

more feeding opportunity. Yang and Deb further introduced the concept of Lévy flight [65] to improve the CS algorithm. Lévy flight follows a random walk and foraging behavior of animals and the next movement always depends on the current step and the transition probability to the next step. As proposed by Yang [64] the three main strategies on which the CS is based are as follows:

- Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.

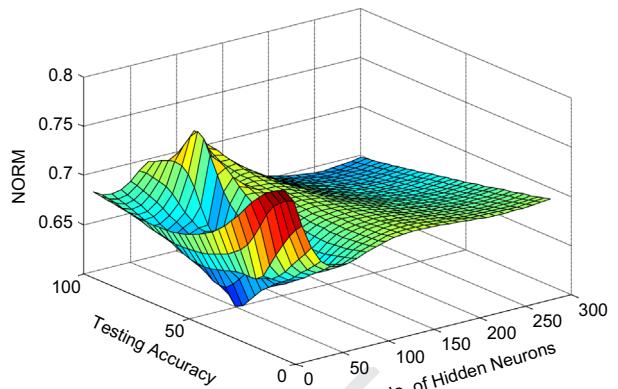


Fig. 41. No. of hidden neurons vs testing accuracy vs norm (ELM).

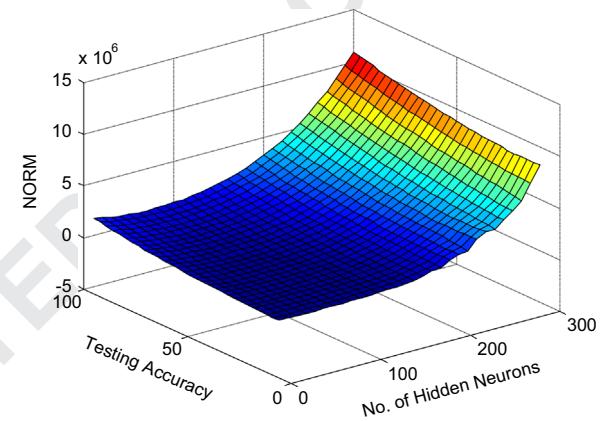


Fig. 42. No. of hidden neurons vs testing accuracy vs norm (CSELM).

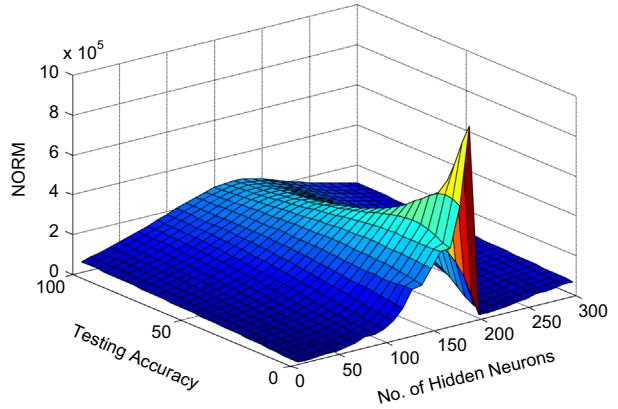


Fig. 43. No. of hidden neurons vs testing accuracy vs norm (ICSELM).

- The best nests with high quality of eggs (solutions) will be forwarded to the next generations.
- The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a complete new nest in a new location. The last strategy is approximated by a fraction p_a of the n nests being replaced by new nests (with new random solutions at new locations).

For generating the new solutions $x_i^{(t+1)}$ for, say cuckoo i , a Lévy flight is performed

$$x_i^{(t+1)} = x_i^t + (\alpha \oplus \text{Levy}(\lambda)) \quad (15)$$

where $\alpha > 0$ is the step size which should be related to the scales

of the problem of interest. The product \oplus means entry-wise multiplication. Lévy flights essentially provide a random walk while their random steps are drawn from a levy distribution for large steps and $(1 < \lambda \leq 3)$.

Cuckoo search algorithm

Begin

 Objective function $f(x)$, $x = [x_1, x_2, \dots, x_d]^T$

 Generate an Initial Population of 'n' number of host nests or different solutions $x_i (i=1, 2, \dots, n)$

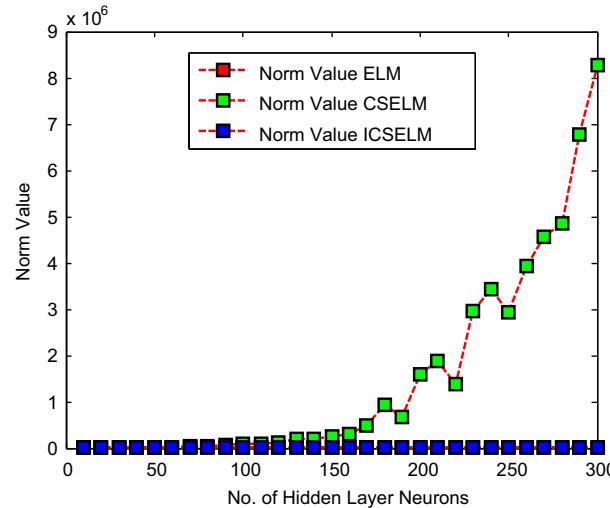


Fig. 44. Comparison of norm value (ELM, CSELM, ICSELM).

while ($t < \text{Max_Iteration}$)

do

 Get a Cuckoo (say i) randomly by Levy flights

 Evaluate it's fitness F_i

 Choose a nest among n (say j) randomly

if ($F_i > F_j$) **then**

 Replace j by the new solution

end if

 Abandon a fraction (p_a) of worst nests (and build new ones at new locations)

 Keep the best solutions (or nests with quality solutions)

 Rank the solutions and find the current best

end while

Process results

end

3.2. Improved cuckoo search

However, to enhance the system stability, the standard cuckoo search algorithm is slightly modified with respect to the p_a , (probability factor) α (step size) and thus improved cuckoo search ELM algorithm (ICSELM) is designed. In the traditional CS algorithm, the values of p_a probability and α are kept fixed and following two conclusions or problems are found from the simulation:

- for smaller p_a and larger α , the performance of algorithm is poor [60] and required more number of iterations to get optimal solutions.
- for smaller α and larger p_a , the speed of convergence becomes high but is unable to find the optimal solution.

Table 6

Training time (Trg Time), training accuracy (Trg Acc), testing accuracy (Tst Acc) and standard deviation of testing accuracy (Tst Dev) for ELM, OSELM, CSELM and ICSELM (Diabetes Dataset).

L	ELM				OSELM				CSELM				ICSELM			
	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc
10	.0025	.7794	.7668	.0270	.0920	.6873	.6589	.0377	.7771	.7838	.0301	.7790	.7850	.0298	100	101
20	.0072	.7892	.7721	.0262	.0671	.6979	.6901	.0353	.7847	.7704	.0233	.7852	.7718	.0367	102	103
30	.0066	.7990	.7719	.0265	.0671	.7316	.7156	.0325	.7993	.7655	.0397	.7998	.7592	.0413	104	105
40	.0128	.8049	.7683	.0250	.1451	.7455	.7104	.0210	.8089	.7634	.0424	.8049	.7768	.0432	106	107
50	.0222	.8107	.7680	.0295	.0874	.7582	.7359	.0276	.8135	.7732	.0333	.8137	.7796	.0433	108	109
60	.0125	.8148	.7676	.0303	.0749	.7613	.7333	.0381	.8193	.7641	.0264	.8156	.7739	.0224	110	111
70	.0246	.8198	.7596	.0348	.1404	.7698	.7380	.0413	.8170	.7669	.0211	.8198	.7345	.0191	112	113
80	.0318	.8248	.7526	.0272	.0967	.7726	.7479	.0434	.8274	.7437	.0306	.8243	.7486	.0444	114	115
90	.0406	.8297	.7506	.0276	.1045	.7842	.7354	.0363	.8318	.7472	.0341	.8306	.7577	.0352	116	117
100	.0452	.8354	.7440	.0301	.1186	.7877	.7448	.0279	.8280	.7423	.0337	.8368	.7408	.0375	118	119
110	.0427	.8418	.7647	.0324	.1498	.7977	.7313	.0489	.8366	.7394	.0318	.8377	.7352	.0368	120	121
120	.0534	.8434	.7431	.0278	.1482	.8042	.7422	.0349	.8396	.7317	.0298	.8444	.7183	.0324	122	123
130	.0690	.8507	.7340	.0280	.1856	.8083	.7198	.0298	.8446	.7437	.0332	.8486	.7218	.0396	124	125
140	.0649	.8532	.7325	.0251	.1794	.8196	.7302	.0349	.8517	.7542	.0314	.8561	.7254	.0439	126	127
150	.0755	.8608	.7280	.0250	.2044	.8125	.7313	.0343	.8569	.7232	.0304	.8559	.7282	.0240	128	129
160	.0836	.8647	.7278	.0287	.2340	.8188	.7297	.0239	.8611	.7451	.0365	.8566	.7465	.0489	130	131
170	.0924	.8686	.7260	.0316	.2683	.8222	.7344	.0385	.8715	.7148	.0424	.8682	.7176	.0331	132	133
180	.0980	.8761	.7186	.0293	.2527	.8248	.7266	.0270	.8747	.7232	.0470	.8776	.7183	.0428	134	135
190	.1158	.8805	.7148	.0273	.3073	.8385	.7292	.0387	.8807	.7148	.0297	.8807	.7092	.0262	136	137
200	.1295	.8838	.7145	.0298	.6131	.8349	.7396	.0384	.8809	.7211	.0408	.8859	.7225	.0305	138	139
210	.1348	.8864	.7168	.0289	.3432	.8307	.7260	.0247	.8878	.7028	.0362	.8896	.6951	.0290	140	141
220	.1679	.8934	.7061	.0310	.6989	.8401	.7354	.0353	.8885	.6894	.0379	.8964	.7148	.0372	142	143
230	.1838	.8993	.7031	.0268	.4072	.8476	.7391	.0284	.8953	.7169	.0401	.8979	.7092	.0312	144	145
240	.1878	.9002	.7049	.0364	.4493	.8432	.7208	.0354	.8969	.7099	.0412	.8925	.7077	.0260	146	147
250	.1838	.9049	.7096	.0282	.6552	.8502	.7052	.0264	.9066	.7000	.0303	.9102	.6923	.0351	148	149
260	.1962	.9089	.6954	.0346	.5538	.8509	.7286	.0374	.9087	.7092	.0257	.9078	.6845	.0257	150	151
270	.2368	.9143	.6891	.0300	.5834	.8563	.7146	.0281	.9139	.6838	.0329	.9099	.7127	.0366	152	153
280	.2237	.9198	.6860	.0316	.6287	.8561	.7208	.0283	.9163	.6754	.0443	.9139	.6754	.0186	154	155
290	.2324	.9213	.6858	.0297	.7363	.8243	.7135	.0642	.9174	.6711	.0322	.9241	.6871	.0300	156	157
300	.2883	.9265	.6807	.0340	.7613	.8649	.7385	.0282	.9245	.6754	.0357	.9219	.7007	.0299	158	159

Table 7

Sensitivity, norm, Gmean and F-score for ELM, CSELM and ICSELM (Diabetes dataset).

L	ELM				CS ELM				ICS ELM			
	Sen.	Norm	Gmean	Fscore	Sen.	Norm	Gmean	Fscore	Sen.	Norm	Gmean	Fscore
10	.8769	49.0963	.7707	.7644	.8854	268.3557	.7209	.7059	.8866	34.2009	.6118	.5720
20	.9113	146.2313	.7502	.7363	.9355	1.3444e+03	.7180	.6935	.8804	894.0870	.7887	.7276
30	.8661	279.4997	.6731	.6522	.8835	2.2224e+03	.6897	.6691	.8953	2.9285e+03	.7153	.6976
40	.8915	402.2186	.6729	.6471	.9158	5.2904e+03	.7896	.7810	.9186	7.3479e+03	.6655	.6324
50	.8560	546.1790	.6293	.6007	.9091	1.1180e+04	.7454	.7309	.9063	1.0379e+04	.7157	.6962
60	.8780	804.1926	.6862	.6658	.8788	1.6975e+04	.7565	.7480	.9043	1.8309e+04	.6138	.5705
70	.8852	1.1768e+003	.6747	.6506	.8455	2.6718e+04	.7795	.7770	.9012	2.6138e+004	.6876	.6632
80	.8321	1.6010e+003	.7482	.7440	.8788	3.7629e+04	.6231	.5880	.8586	4.7402e+004	.5826	.5414
90	.8110	2.1653e+003	.6886	.6795	.6469	5.7695e+04	.7113	.6322	.8706	4.5216e+004	.7099	.6954
100	.8846	2.7454e+003	.6862	.6646	.8391	9.7285e+04	.6987	.6872	.8315	6.3194e+004	.6508	.6318
110	.8615	3.5718e+003	.7548	.7482	.8696	9.0575e+04	.7223	.7101	.9551	9.1032e+004	.6438	.5968
120	.7907	4.9750e+003	.7085	.7043	.8778	1.2847e+05	.6094	.5710	.9390	1.4239e+005	.6852	.6525
130	.7953	6.4744e+003	.6450	.6311	.7917	1.9457e+05	.6427	.6290	.8878	2.2888e+005	.6959	.6757
140	.8409	7.8712e+003	.6697	.6527	.9063	2.0051e+05	.6876	.6622	.8023	2.3725e+005	.5485	.5111
150	.8571	9.6841e+003	.6680	.6477	.7895	2.5035e+05	.6609	.6505	.8788	3.0810e+005	.6393	.6083
160	.8130	1.1941e+004	.6779	.6688	.8283	3.0138e+05	.7344	.7291	.9175	3.8298e+005	.7281	.7091
170	.7786	1.4785e+004	.6084	.5904	.8421	4.8941e+05	.6278	.6017	.8421	3.8634e+005	.6419	.6190
180	.7951	1.8299e+004	.6305	.6139	.8280	9.4649e+05	.6628	.6467	.7447	5.8442e+005	.5284	.4988
190	.8145	2.2162e+004	.6922	.6831	.8144	6.5931e+05	.6591	.6446	.8191	9.8051e+005	.6532	.6368
200	.8496	2.7174e+004	.6350	.6090	.8315	1.5867e+06	.6136	.5863	.8876	1.4880e+04	.6471	.6160
210	.8095	3.2410e+004	.5860	.5567	.8352	1.8790e+06	.6002	.5689	.8068	1.8308e+04	.5988	.5732
220	.8016	3.8765e+004	.6331	.6159	.7396	1.3705e+06	.6081	.5966	.8046	2.0096e+04	.7156	.7107
230	.8000	4.4051e+004	.6236	.6048	.7634	2.9571e+06	.5720	.5490	.9469	2.5847e+04	.6507	.6288
240	.8182	5.3058e+004	.5954	.5666	.8842	3.4444e+06	.6134	.5746	.8353	3.5456e+04	.5930	.5599
250	.8168	6.1688e+004	.6846	.5676	.6327	2.9258e+06	.6620	.6287	.7419	4.0217e+04	.6740	.6709
260	.8507	6.9967e+004	.6056	.5722	.7674	3.9421e+06	.6304	.6184	.8280	5.2390e+04	.6628	.6467
270	.7293	8.3957e+004	.6578	.6543	.7778	4.5681e+06	.6472	.6364	.8571	5.1695e+04	.7101	.6977
280	.7647	9.4786e+004	.6712	.6655	.6989	4.8502e+06	.5341	.5154	.7802	5.6513e+04	.6427	.6308
290	.7238	1.1059e+005	.6210	.6126	.7907	6.7869e+06	.6722	.6634	.7938	6.4570e+04	.5313	.4911
300	.8115	1.2717e+005	.6637	.6505	.7454	8.2680e+06	.7100	.7091	.7935	6.5687e+04	.7014	.6961

Table 8

Specificity (Spe.), TP, FN, FP and TN for ELM, CSELM and ICSELM (Diabetes dataset).

L	ELM					CSELM					ICSELM				
	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN
10	.6774	114	16	20	42	.5870	85	11	19	27	.4222	86	11	26	19
20	.6176	113	11	26	42	.5510	87	6	22	27	.6200	81	11	19	31
30	.5231	110	17	31	34	.5385	91	12	18	21	.5714	77	9	24	32
40	.5079	115	14	31	32	.6809	87	8	15	32	.4821	79	7	29	27
50	.4627	107	18	36	31	.6111	80	8	21	33	.5652	87	9	20	26
60	.5362	108	15	32	37	.6512	87	12	15	28	.4167	85	9	28	20
70	.5143	108	14	34	36	.7188	93	17	9	23	.5246	73	8	29	32
80	.6727	114	23	18	37	.4419	87	12	24	19	.3953	85	14	26	17
90	.5846	103	24	27	38	.8021	77	19	22	24	.5789	74	11	24	33
100	.5323	115	15	29	33	.5818	73	14	23	32	.5094	74	15	26	27
110	.6613	112	18	21	41	.6000	80	12	20	30	.4340	85	4	30	23
120	.6349	102	27	23	40	.4231	79	11	30	22	.5000	77	5	30	30
130	.5231	101	26	31	34	.5217	76	20	22	24	.5455	87	11	20	24
140	.5333	111	21	28	32	.5217	87	9	22	24	.3750	69	17	35	21
150	.5205	102	17	35	38	.5532	75	20	21	26	.4651	87	12	23	20
160	.5652	100	23	30	39	.6512	82	17	15	28	.5778	89	8	19	26
170	.4754	102	29	32	29	.4681	80	15	25	22	.4894	80	15	24	23
180	.5000	97	25	35	35	.5306	77	16	23	26	.3750	70	24	30	18
190	.5882	101	23	28	40	.5333	79	18	21	24	.5208	77	17	23	25
200	.4746	130	20	31	28	.4528	74	15	29	24	.4717	79	10	28	25
210	.4242	102	24	38	28	.4314	76	15	29	22	.4444	71	17	30	24
220	.5000	101	25	33	33	.5000	71	25	23	23	.6364	70	17	20	35
230	.4861	96	24	37	35	.4286	71	22	28	21	.5000	83	15	22	22
240	.4333	108	24	34	26	.4255	84	11	27	20	.4211	71	14	33	24
250	.5738	107	24	26	35	.7083	68	28	20	26	.6122	69	24	19	30
260	.4310	114	20	33	25	.5179	66	20	27	29	.5306	77	16	23	26
270	.5932	97	36	24	35	.5385	70	20	24	28	.5882	78	13	21	30
280	.5890	91	28	30	43	.4082	65	28	29	20	.5294	71	20	24	27
290	.5263	85	31	36	40	.5714	68	18	24	32	.3556	77	20	29	16
300	.5429	99	23	32	38	.6744	74	25	14	29	.6200	73	19	19	31

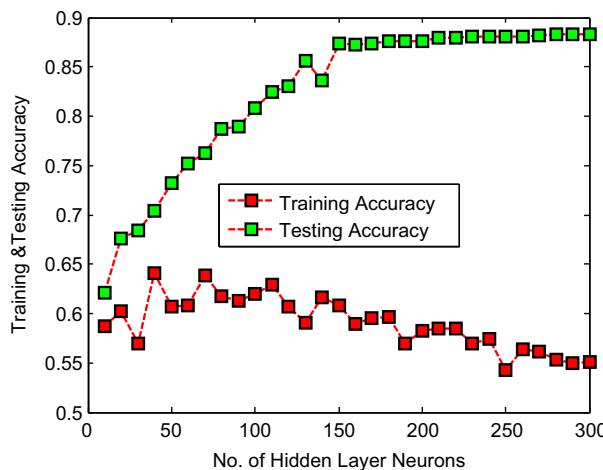


Fig. 45. Training and testing accuracy (ELM).

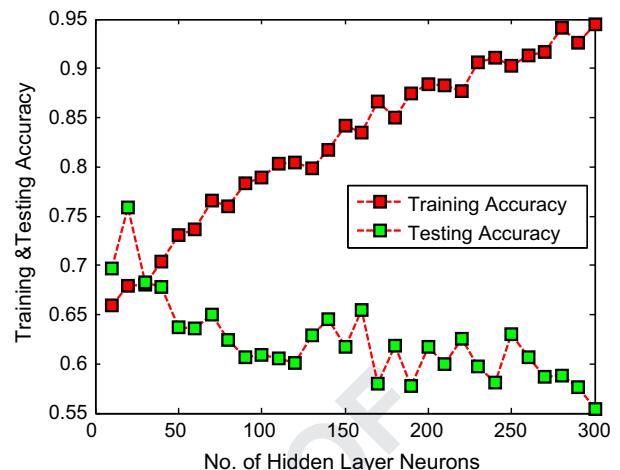


Fig. 48. Training and testing accuracy (ICSELM).

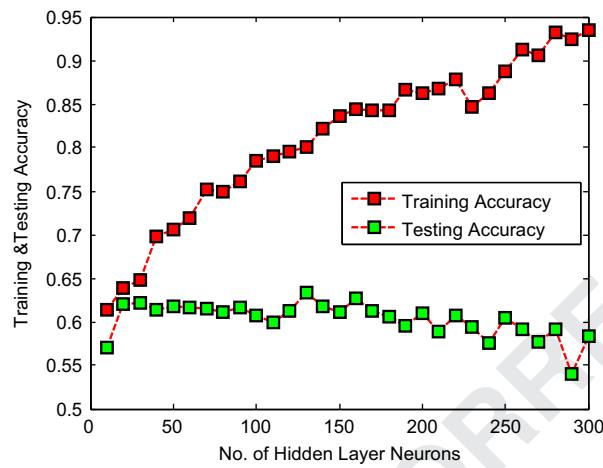


Fig. 46. Training and testing accuracy (OSELM).

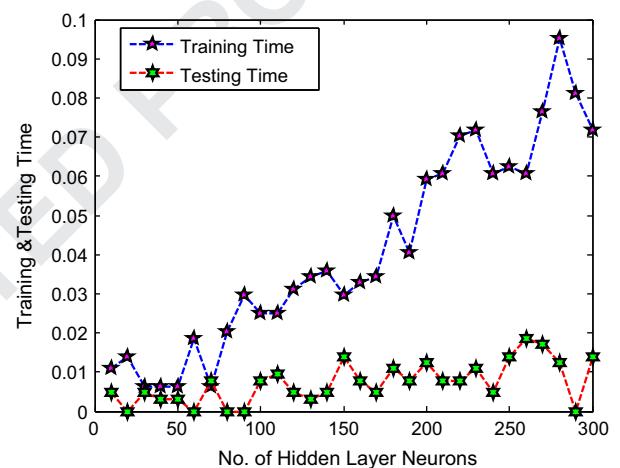


Fig. 49. Training and testing time (ELM).

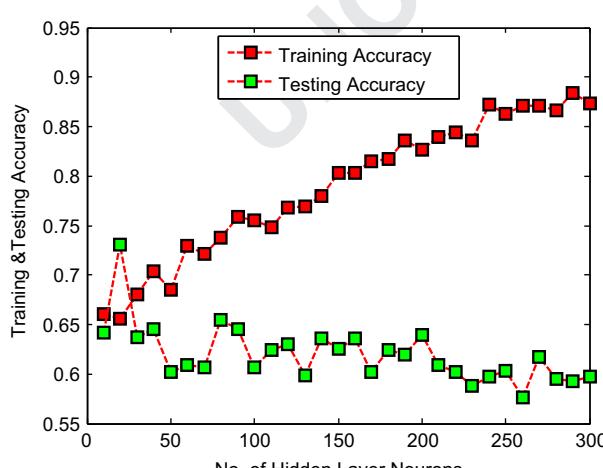


Fig. 47. Training and testing accuracy (CSELM).

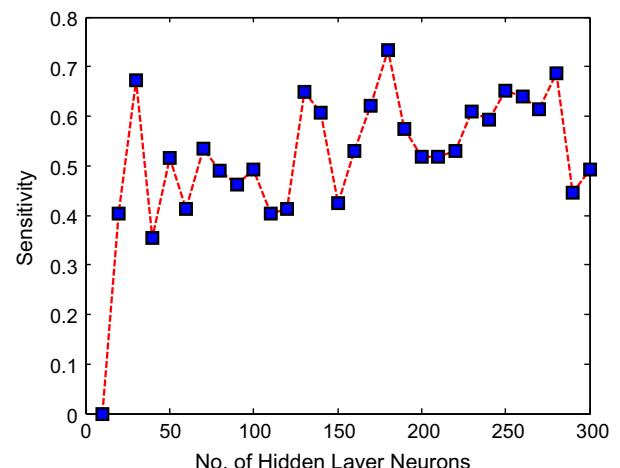


Fig. 50. Sensitivity (ELM).

The value of p_a and α is modified using the following equations:

$$p_a(it) = p_{amax} - \frac{it}{NI}(p_{amax} - p_{amin}) \quad (16)$$

$$\alpha(it) = \alpha_{max} \exp(k*it) \quad (17)$$

$$k = \frac{1}{NI} \ln \left(\frac{\alpha_{max}}{\alpha_{min}} \right) \quad (18)$$

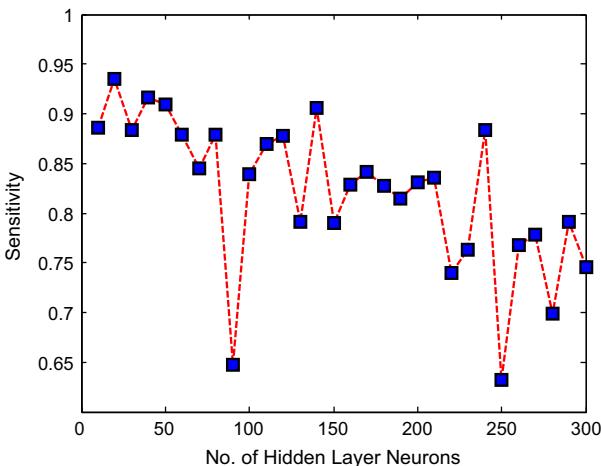


Fig. 51. Sensitivity (CSELM).

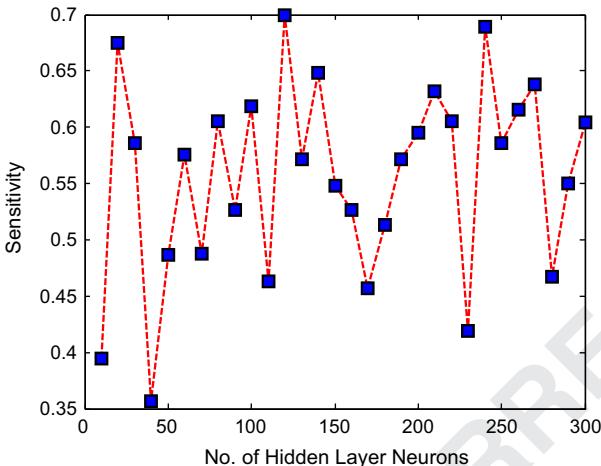


Fig. 52. Sensitivity (ICSELM).

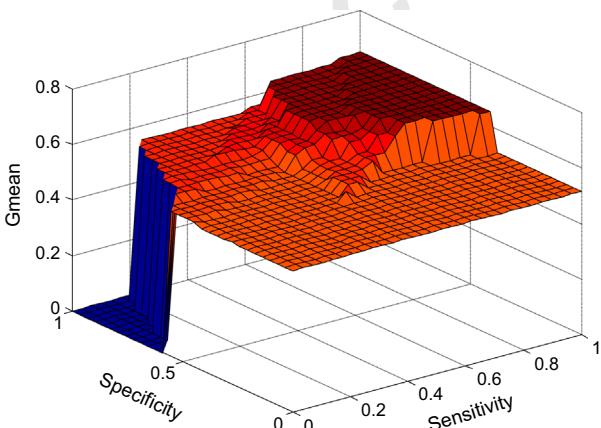


Fig. 53. Sensitivity vs Specificity vs Gmean (ELM).

where 'NI' and 'it' represent the total number of iterations and current iteration respectively.

The parameters of the improved cuckoo search considered in this study are taken in a hit and trial method. They are as follows:

$$p_{amin} = 0.05, p_{amax} = 0.5, \alpha_{min} = 0.01, \alpha_{max} = 0.5, \lambda = 1.5$$

Improved cuckoo search algorithm

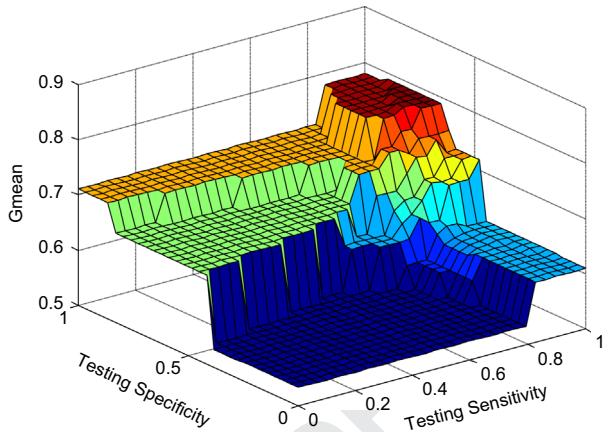


Fig. 54. Sensitivity vs Specificity vs Gmean (CSELM).

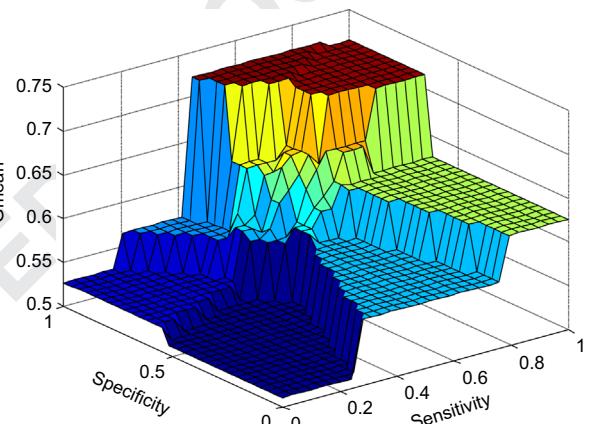


Fig. 55. Sensitivity vs Specificity vs Gmean (ICSELM).

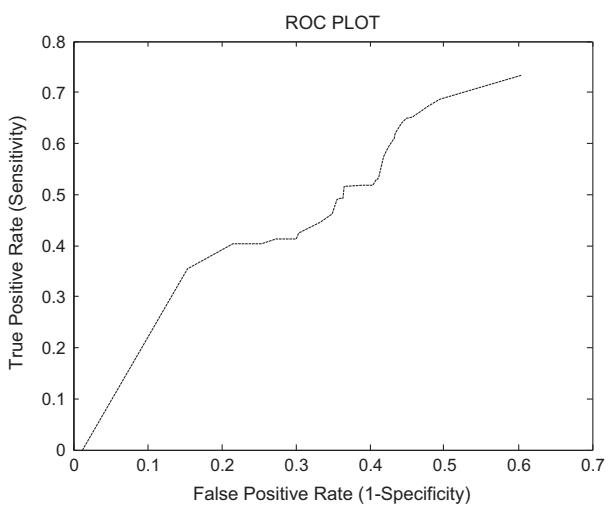


Fig. 56. ROC Plot (ELM).

Begin

Objective function $f(x)$, $x = [x_1, x_2, \dots, x_d]^T$

Generate an Initial Population of 'n' no. of host nests or different solutions $x_i (i=1, 2, \dots, n)$

Initialize p_{amin} , p_{amax} , α_{min} , α_{max} , λ , NI

while ($it < NI$)

do

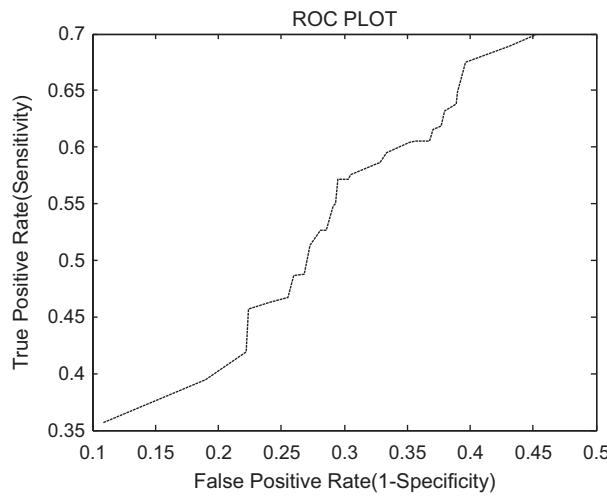


Fig. 57. ROC Plot (ICSELM).

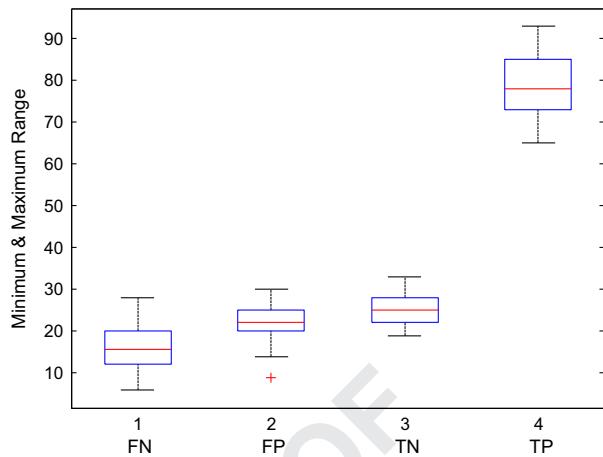


Fig. 60. Box Plot (ICSELM).

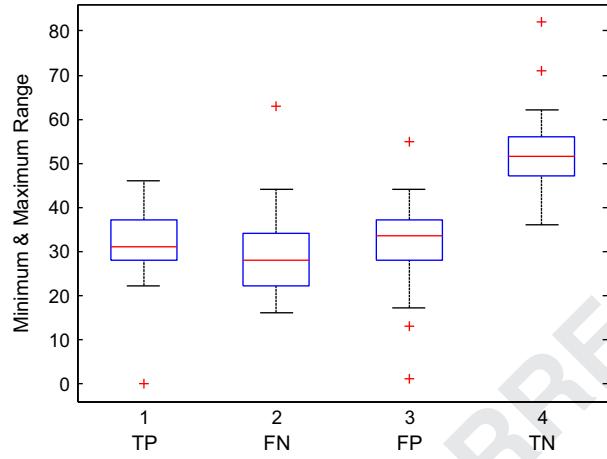


Fig. 58. Box Plot (ELM).

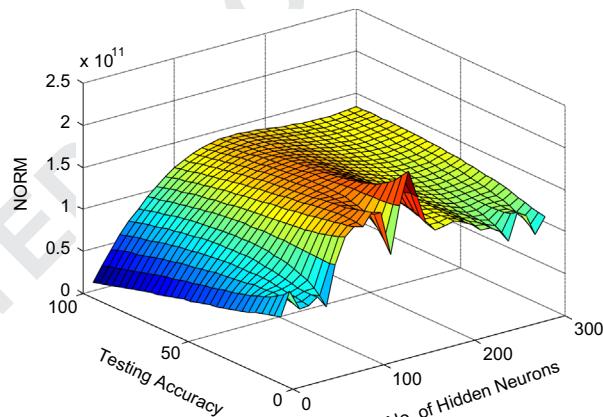


Fig. 61. No. of hidden neurons vs testing accuracy vs norm (ELM).

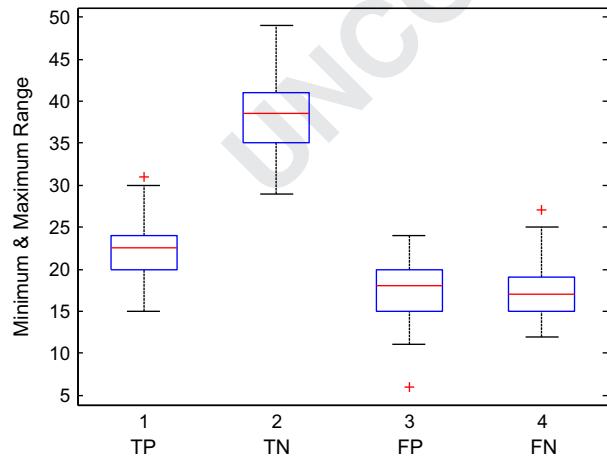


Fig. 59. Box Plot (CSELM).

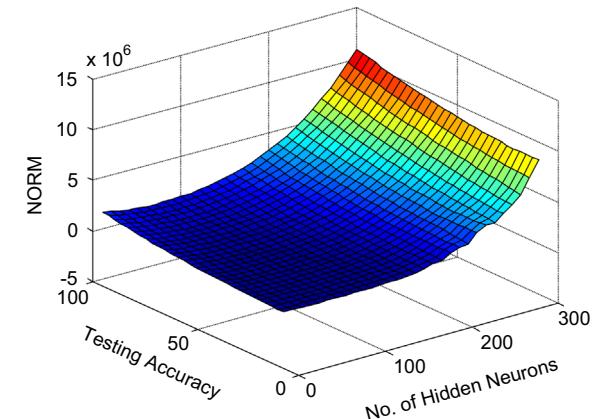


Fig. 62. No. of hidden neurons vs testing accuracy vs norm (CSELM).

Replace j by the new solution**end if**Abandon a fraction (p_a) of worst nests (and build new ones at new locations)

Keep the best solutions (or nests with quality solutions)

Rank the solutions and find the current best

end while**end while**

Process the results

end

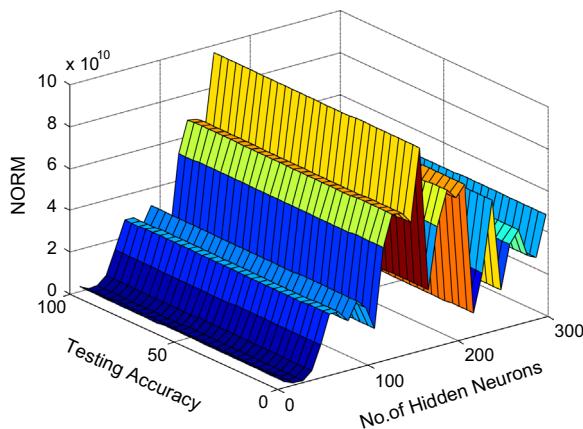


Fig. 63. No. of hidden neurons vs testing accuracy vs norm (ICSELM).

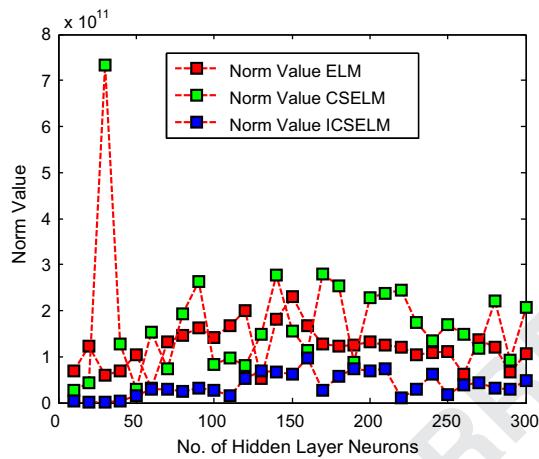


Fig. 64. Norm of ELM, CSELM and ICSELM.

3.3. Cuckoo search based extreme learning machine (CSELM)

This sub-section explains the steps of standard CSELM for classifying four bench mark datasets.

CSELM algorithm

Begin

 Prepare the training data and testing data

 Objective function $f(x)$, $x = [x_1, x_2, \dots, x_d]^T$

 Generate an Initial Population of 'n' no. of host nests or different solutions $x_i (i=1, 2, \dots, n)$

while ($t < \text{Max_Iteration}$)

do

 Get a Cuckoo (say i) randomly by Levy flights

 Compute the hidden layer output matrix H

 Obtain the output weight β^* using the equation

$$\beta^* = H^T T$$

 Evaluate it's fitness F_i

 Choose a nest among n (say j) randomly

if ($\text{norm}(F_i) > \text{norm}(F_j)$) **then**

 Replace j by the new solution

end if

 Abandon a fraction (p_a) of worst nests (and build new ones at new locations)

 Keep the best solutions (or nests with quality solutions)
 Rank the solutions and find the current best
 Calculate Output weight using $\text{pinv}(H^*) * T$
 Calculate $Y = (H^* * \text{bestweight})^*$ and determine actual label of the class
 Compare expected label and actual label of the class
end while
 Process the results
end

3.4. Improved cuckoo search based extreme learning machine (ICSELM)

In this section, the steps required for Improved cuckoo search based extreme learning machine (ICSELM) to classify all four data sets are discussed.

ICSELM algorithm

Begin

 Prepare the training data and testing data

 Objective function $f(x)$, $x = [x_1, x_2, \dots, x_d]^T$

 Generate an Initial Population of 'n' no. of host nests or different solutions $x_i (i=1, 2, \dots, n)$

 Initialize p_{amin} , p_{amax} , α_{min} , α_{max} , λ , NI

while ($t < NI$)

do

 update the value of P_a , α and K using Eqs. (16)–(18) respectively

while ($t < \text{Max_Generation}$)

do

 Get a Cuckoo (say i) randomly by Levy flights

 Compute the hidden layer output matrix H

 Obtain the output weight β^* using the equation

$$\beta^* = H^T T$$

 Evaluate it's fitness F_i

 Choose a nest among n (say j) randomly

if ($\text{norm}(F_i) > \text{norm}(F_j)$) **then**

 Replace j by the new solution

end if

 Abandon a fraction (p_a) of worst nests (and build new ones at new locations)

 Keep the best solutions (or nests with quality solutions)

 Rank the solutions and find the current best

 Calculate Output weight using $\text{pinv}(H^*) * T$

 Calculate $Y = (H^* * \text{bestweights})^*$ and determine actual label of the class

 Compare expected label and actual label of the class

end while

end while

 Process the results

end

4. The datasets used for different experiments

This section introduces the four benchmark datasets that are collected from UCI repository for classification. They are Wisconsin Breast Cancer, Diabetes, Bupa and Hepatitis. Each dataset is normalized between the range $[-1, 1]$ using Eq. (20) and shuffled using randperm (). Each dataset is divided into two data files i.e. training and testing. The number of samples taken for training and testing, the available features and the number of classes in each

Table 9

Training time (Trg Time), training accuracy (Trg Acc), testing accuracy (Tst Acc) and standard deviation of testing accuracy (Tst Dev) for ELM, OSELM, CSELM and ICSELM (Bupa dataset).

L	ELM				OSELM				CSELM				ICSELM				
	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev
10	.0109	.5870	.6216	.0368	.0577	.6136	.5705	.0493	.6603	.6417	.0507	.6598	.6965	.0561			
20	.0140	.6021	.6759	.0229	.0499	.6392	.6205	.0375	.6563	.7308	.0740	.6794	.7583	.0406			
30	.0062	.5692	.6839	.0342	.0265	.6482	.6219	.0398	.6799	.6375	.0704	.6799	.6827	.0820			
40	.0062	.6411	.7045	.0278	.0406	.6975	.6137	.0529	.7035	.6448	.0454	.7035	.6775	.0522			
50	.0062	.6075	.7322	.0258	.0577	.7055	.6178	.0418	.6854	.6021	.0668	.7312	.6375	.0403			
60	.0187	.6082	.7523	.0373	.0390	.7196	.6171	.0468	.7296	.6094	.0693	.7362	.6354	.0364			
70	.0062	.6390	.7623	.0242	.0359	.7523	.6151	.0352	.7211	.6063	.0453	.7663	.6500	.0562			
80	.0203	.6178	.7874	.0267	.0546	.7487	.6110	.0387	.7372	.6542	.0444	.7598	.6240	.0418			
90	.0296	.6130	.7894	.0369	.0624	.7608	.6164	.0631	.7588	.6448	.0446	.7839	.6073	.0515			
100	.0250	.6199	.8085	.0266	.0624	.7849	.6068	.0300	.7553	.6063	.0273	.7889	.6094	.0461			
110	.0250	.6288	.8251	.0227	.0593	.7905	.6000	.0261	.7477	.6240	.0580	.8035	.6052	.0630			
120	.0312	.6075	.8307	.0243	.0640	.7950	.6130	.0514	.7678	.6302	.0494	.8045	.6010	.0703			
130	.0343	.5904	.8558	.0300	.0686	.8010	.6329	.0386	.7688	.5990	.0488	.7990	.6292	.0476			
140	.0359	.6158	.8367	.0171	.0608	.8221	.6178	.0379	.7794	.6365	.0359	.8171	.6458	.0374			
150	.0296	.6082	.8739	.0297	.0920	.8367	.6116	.0405	.8030	.6260	.0409	.8422	.6167	.0332			
160	.0328	.5897	.8725	.0170	.0858	.8437	.6274	.0330	.8035	.6365	.0492	.8347	.6552	.0490			
170	.0343	.5952	8739	.0135	.1030	.8427	.6119	.0361	.8146	.6021	.0542	.8658	.5802	.0547			
180	.0499	.5966		.8755	.0322	.0967	.8422	.6055	.0479	.8176	.6240	.0397	.8503	.6188	.0408		
190	.0406	.5692	.8759	.0179	.1030	.8663	.5959	.0420	.8357	.6198	.0672	.8749	.5771	.0414			
200	.0593	.5829	.8759	.0180	.1108	.8623	.6096	.0473	.8271	.6396	.0534	.8844	.6177	.0354			
210	.0608	.5849	.8800	.0184	.1186	.8678	.5890	.0491	.8397	.6094	.0323	.8824	.6000	.0469			
220	.0702	.5849	.8801	.0149	.1170	.8789	.6075	.0342	.8447	.6021	.0285	.8774	.6260	.0359			
230	.0718	.5692	.8809	.0163	.1404	.8472	.5945	.0535	.8357	.5885	.0330	.9060	.5979	.0717			
240	.0608	.5740	.8809	.0196	.1466	.8628	.5760	.0324	.8719	.5969	.0414	.9111	.5813	.0524			
250	.0624	.5425	.8809	.0222	.1685	.8874	.6041	.0438	.8628	.6031	.0331	.9025	.6302	.0432			
260	.0608	.5637	.8810	.0131	.1420	.9131	.5911	.0449	.8714	.5760	.0506	.9136	.6063	.0495			
270	.0764	.5616	.8815	.0159	.1669	.9060	.5774	.0453	.8709	.6177	.0529	.9171	.5865	.0562			
280	.0952	.5527	.8830	.0187	.1778	.9327	.5911	.0394	.8658	.5948	.0352	.9407	.5875	.0527			
290	.0811	.5500	.8835	.0114	.2012	.9246	.5404	.0348	.8834	.5927	.0693	.9266	.5760	.0447			
300	.0718	.5507	.8836	.0104	.1997	.9347	.5836	.0341	.8739	.5979	.0508	.9452	.5542	.0413			

Table 10

Sensitivity, norm, Gmean and F-score value (ELM, CSELM and ICSELM) (Bupa dataset).

L	ELM				CSELM				ICSELM			
	Sen.	Norm	Gmean	Fscore	Sen.	Norm	Gmean	Fscore	Sen.	Norm	Gmean	Fscore
10	0	6.9284e+10	0	0	.4359	2.7102e+010	.5866	.5617	.3947	2.4983e+09	.5656	.5309
20	.4032	1.2280e+11	.5411	.5185	.3888	4.2385e+010	.5749	.5336	.6750	4.4042e+08	.6587	.6585
30	.6727	5.9879e+10	.5159	.4982	.3077	7.3237e+11	.5037	.4482	.5854	6.5122e+08	.7222	.7065
40	.3548	6.8039e+10	.5477	.4998	.3864	1.2781e+11	.5519	.5186	.3571	4.2499e+09	.5270	.4895
50	.5152	1.0329e+11	.5501	.5490	.2340	2.8877e+10	.4584	.3713	.4865	1.5140e+010	.5814	.5723
60	.4127	3.1632e+10	.5370	.5189	.4615	1.5243e+011	.6036	.5825	.5758	2.8583e+010	.6413	.6376
70	.5345	1.3165e+11	.5832	.5810	.5385	7.3335e+010	.6804	.6622	.4878	2.8458e+10	.5729	.5655
80	.4912	1.4612e+11	.5850	.5762	.6207	1.9357e+11	.6668	.6651	.6053	2.3392e+010	.6541	.6521
90	.4615	1.6337e+11	.5120	.5092	.4773	2.6352e+11	.5748	.5650	.5263	3.1532e+010	.6390	.6272
100	.4921	1.4234e+011	.5658	.5603	.4878	8.2329e+010	.5881	.5780	.6190	2.7113e+010	.6151	.6151
110	.4030	1.6686e+11	.5624	.5325	.6358	9.6716e+010	.6458	.6318	.4634	1.5894e+010	.5732	.5605
120	.4133	1.9885e+11	.5555	.5321	.5581	7.9648e+010	.6071	.6050	.7000	5.3313e+010	.7159	.7157
130	.6491	5.1792e+10	.5978	.5958	.3333	1.4826e+011	.4783	.4487	.5714	6.8048e+010	.6347	.6312
140	.6071	1.8171e+11	.6362	.6355	.4545	2.7823e+011	.5610	.5488	.6486	6.7403e+010	.6291	.6288
150	.4237	2.2952e+011	.5222	.5110	.5000	1.5443e+011	.5798	.5735	.5476	6.1889e+010	.5872	.5858
160	.5303	1.6819e+011	.5522	.5517	.5455	1.1389e+011	.5972	.5948	.5263	9.6357e+010	.6319	.6215
170	.6207	1.2710e+011	.6056	.6054	.5750	2.8019e+011	.5642	.5641	.4565	2.7365e+010	.5812	.5647
180	.7333	1.2190e+011	.6530	.6486	.4694	2.5398e+011	.5912	.5758	.5128	5.6740e+010	.6073	.5988
190	.5741	1.2428e+011	.5473	.5467	.5294	8.8249e+010	.5917	.5880	.5714	7.2589e+010	.6197	.6177
200	.5179	1.3320e+011	.5417	.5412	.5952	2.2758e+011	.6472	.6449	.5952	6.8054e+010	.6211	.6206
210	.5185	1.2452e+11	.5414	.5409	.4894	2.3669e+011	.5564	.5519	.6316	7.2981e+010	.6261	.6261
220	.5294	1.2076e+11	.5826	.5799	.4681	2.4468e+011	.5442	.5381	.6053	1.0350e+010	.6044	.6044
230	.6087	1.0448e+11	.6095	.4762	.17402e+011	.5712	.5619	.4186	2.8426e+010	.5105	.5006	
240	.5938	1.0989e+11	.6136	.6133	.6970	1.3498e+011	.6042	.5981	.6889	6.2138e+010	.6259	.6230
250	.6508	1.1035e+11	.5940	.5915	.4762	1.7019e+011	.5789	.5680	.5854	1.8236e+010	.6525	.6486
260	.6393	6.1851e+10	.5687	.5648	.5366	1.4892e+011	.5926	.5897	.6154	3.8729e+010	.6405	.6400
270	.6129	1.3733e+11	.6040	.6039	.6579	1.1744e+011	.6565	.6565	.6383	4.4269e+010	.6355	.6355
280	.6866	1.1942e+11	.6184	.6150	.4872	2.2122e+011	.5391	.5363	.4667	3.1242e+010	.5897	.5739
290	.4464	6.6809e+10	.5127	.5079	.6383	9.3382e+010	.6456	.6456	.5500	2.8617e+010	.6189	.6146
300	.4921	1.0657e+11	.5279	.5266	.6000	2.0712e+011	.6628	.6595	.6047	4.8295e+010	.5752	.5745

Table 11

Specificity (Spe.), TP, FN, FP, TN of ELM, CSELM, ICSELM (Bupa dataset).

L	ELM					CSELM					ICSELM				
	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN
10	.9880	0	63	1	82	.7895	17	22	12	45	.8103	15	23	11	47
20	.7262	25	37	23	61	.8500	14	22	9	51	.6429	27	13	20	36
30	.3956	37	18	55	36	.8246	12	27	10	47	.8909	24	17	06	49
40	.8452	22	40	13	71	.7885	17	27	11	41	.7778	15	27	12	42
50	.5875	34	32	33	47	.8980	11	36	5	44	.6949	18	19	18	41
60	.6988	26	37	25	58	.7895	18	21	12	45	.7143	19	14	18	45
70	.6364	31	27	32	56	.8596	21	18	8	49	.6727	20	21	18	37
80	.6966	28	29	27	62	.7164	18	11	19	48	.7069	23	15	17	41
90	.5679	30	35	35	46	.6923	21	23	16	36	.7759	20	18	13	45
100	.6506	31	32	29	54	.7091	20	21	16	39	.6111	26	16	21	33
110	.7848	27	40	17	62	.7115	25	19	15	37	.7091	19	22	16	39
120	.7465	31	44	18	53	.6604	24	19	18	35	.7321	28	12	15	41
130	.5506	37	20	40	49	.6863	15	30	16	35	.7049	20	15	18	43
140	.6667	34	22	30	60	.6923	20	24	16	36	.6102	24	13	23	36
150	.6437	25	34	31	56	.6724	19	19	19	39	.6296	23	19	20	34
160	.5750	35	31	34	46	.6538	24	20	18	34	.7586	20	18	14	44
170	.5909	36	22	36	52	.5536	23	17	25	31	.7400	21	25	13	37
180	.5814	44	16	36	50	.7447	23	26	12	35	.7193	20	19	16	41
190	.5217	31	23	44	48	.6613	18	16	21	41	.6721	20	15	20	41
200	.5667	29	27	39	51	.7037	25	17	16	38	.6481	25	17	19	35
210	.5652	28	26	40	52	.6327	23	24	18	31	.6207	24	14	22	36
220	.6410	36	32	28	50	.6327	22	25	18	31	.6034	23	15	23	35
230	.6104	42	27	30	47	.6852	20	22	17	37	.6226	18	25	20	33
240	.6341	38	26	30	52	.5238	23	10	30	33	.5686	31	14	22	29
250	.5422	41	22	38	45	.7037	20	22	16	38	.7273	24	17	15	40
260	.5059	39	22	42	43	.6545	22	19	19	36	.6667	24	15	19	38
270	.5952	38	24	34	50	.6552	25	13	20	38	.6327	30	17	18	31
280	.5570	46	21	35	44	.5965	19	20	23	34	.7451	21	24	13	38
290	.5889	25	31	37	53	.6531	30	17	17	32	.6964	22	18	17	39
300	.5663	31	32	36	47	.7321	24	16	15	41	.5472	26	17	24	29

Table 12

Training time (Trg Time), training accuracy (Trg Acc), testing accuracy (Tst Acc) and standard deviation of testing accuracy (Tst Dev) for ELM, OSELM, CSELM and ICSELM (Hepatitis dataset).

L	ELM				OSELM				CSELM				ICSELM			
	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Time	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc	Tst Dev	Trg Acc	Tst Acc
10	.0000	.8267	.8160	.0847	.0047	.8320	.8027	.0401	.8463	.8680	.0655	.8513	.8612	.0611		
20	.0064	.8240	.8400	.0754	.0062	.8495	.8120	.0678	.8975	.8516	.0965	.9013	.9150	.0129		
30	.0062	.8213	.7520	.1029	.0054	.7980	.7667	.0654	.9375	.7920	.0559	.9388	.9388	.0111		
40	.0109	.8027	.7480	.0999	.0043	.8005	.7787	.0597	.9613	.8040	.0832	.9688	.9712	.1334		
50	.0094	.7613	.7600	.0864	.0031	.8004	.7440	.0618	.9850	.8080	.0920	.9763	.9875	.0144		
60	.0078	.7480	.7480	.0755	.0143	.8054	.7544	.0665	.9988	.7560	.0832	.9963	1	0		
70	.0234	.7027	.7200	.0864	.0234	.8034	.7532	.0623	1	.6920	.0962	1	1	0		
80	.0156	.5413	.6960	.0908	.0343	.8043	.7342	.0723	1	.7000	.0736	1	1	0		
90	.0125	.7200	.7240	.0788	.0332	.7998	.7331	.0763	1	.7200	.0777	1	1	0		
100	.0125	.7080	.6960	.0804	.0250	.7772	.7267	.0889	1	.7480	.1386	1	1	0		
110	.0125	.7453	.7720	.0681	.0443	.7789	.7234	.0756	1	.7480	.0732	1	1	0		
120	.0094	.7453	.7600	.0884	.0332	.7765	.7221	.0687	1	.7160	.1057	1	1	0		
130	.0156	.7067	.7480	.0732	.0454	.7734	.7199	.0274	1	.7600	.0822	1	1	0		
140	.0187	.7707	.7840	.0965	.0551	.7654	.7107	.0343	1	.6880	.0527	1	1	0		
150	.0172	.7667	.6680	.1051	.0499	.7427	.7027	.0189	1	.7120	.1080	1	1	0		
160	.0234	.7693	.7360	.1239	.0432	.7432	.7187	.0188	1	.7200	.0980	1	1	0		
170	.0125	.7653	.7280	.0920	.0545	.7544	.7234	.0186	1	.7000	.1120	1	1	0		
180	.0172	.7787	.7640	.0832	.0565	.7546	.7454	.0187	1	.7320	.1118	1	1	0		
190	.0234	.7747	.7560	.0717	.0544	.7564	.7443	.0213	1	.7880	.0823	1	1	0		
200	.0094	.7773	.7840	.0540	.0671	.7826	.7573	.0187	1	.6800	.0908	1	1	0		
210	.0608	.7800	.7720	.1100	.0656	.7843	.7595	.0256	1	.7560	.0741	1	1	0		
220	.0296	.7733	.7800	.0567	.0677	.7954	.7612	.0523	1	.7240	.0853	1	1	0		
230	.0250	.7893	.7640	.0875	.0687	.7934	.7668	.0453	1	.8000	.0777	1	1	0		
240	.0203	.7920	.7840	.0631	.0898	.8056	.7776	.0345	1	.7720	.0823	1	1	0		
250	.0187	.7787	.7680	.0620	.0936	.8081	.7773	.0350	1	.7440	.0711	1	1	0		
260	.0187	.7867	.8080	.0700	.0974	.8002	.7667	.0434	1	.7440	.0602	1	1	0		
270	.0187	.7840	.7880	.0501	.0985	.7989	.7664	.0342	1	.8160	.1070	1	1	0		
280	.0187	.8027	.7960	.0638	.1115	.7965	.7589	.0345	1	.7600	.1398	1	1	0		
290	.0187	.7773	.8000	.0873	.1123	.7998	.7511	.0334	1	.7480	.0681	1	1	0		
300	.0905	.7907	.7680	.0648	.1342	.7938	.7427	.0252	1	.8000	.0732	1	1	0		

Table 13

Sensitivity, norm, Gmean and F-score value (ELM, CSELM and ICSELM) (Hepatitis dataset).

L	ELM				CS ELM				ICS ELM			
	Sens.	Norm	Gmean	Fscore	Sens.	Norm	Gmean	Fscore	Sens.	Norm	Gmean	Fscore
10	.5294	6.6841	.7085	.6795	.2857	804.808	.5040	.4324	.5000	215.278	.6726	.6441
20	.4667	13.437	.6716	.6295	.7500	619.572	.8018	.8000	.5000	507.84	.7071	.6667
30	.7000	17.1635	.7555	.7533	.8333	1.58e+03	.8635	.8629	.3333	515.3952	.5222	.4737
40	.5294	33.3029	.6823	.6609	.3333	2.73e+03	.5130	.4688	.8333	2.62e+03	.8885	.8867
50	.3889	60.376	.5416	.5132	.5556	281.235	.5893	.5882	.5714	695.10	.6667	.6588
60	.3125	94.6315	.4427	.4171	.4000	3.42e+03	.5831	.5440	.6000	2.74e+03	.6928	.6875
70	.6000	150.550	.6325	.6316	.5714	1.54e+03	.5909	.5906	.6000	6.14e+03	.6708	.6667
80	.6000	1.31e+03	.5385	.5354	.7143	1.22e+03	.7715	.7692	.4000	1.74e+03	.5292	.5091
90	.4706	103.1974	.6241	.6000	.7500	2.11e+03	.8135	.8108	.5714	1.80e+03	.6172	.6154
100	.5294	92.380	.5889	.5856	.7500	180.580	.8402	.8348	.7500	638.926	.8018	.8000
110	.4375	64.138	.6269	.5884	.6000	417.339	.7141	.7034	.4286	381.786	.5976	.5660
120	.5853	56.0925	.6667	.6608	.5000	280.206	.6901	.6557	1	132.819	.9129	.9091
130	.2308	43.4463	.4227	.3556	.5000	143.775	.6283	.6122	.8571	227.772	.8729	.8727
140	.5000	43.4050	.6489	.6275	.4286	134.639	.6172	.5783	.6000	92.5041	.6708	.6667
150	.6923	43.865	.7397	.7381	.4000	91.6880	.5831	.5440	1	124.083	.9177	.9143
160	.3889	34.915	.5782	.5335	.7500	69.7249	.5976	.5825	.5000	70.0541	.5774	.5714
170	.4118	33.6604	.6017	.5609	.4000	47.7588	.5657	.5333	.5000	94.5578	.6901	.6557
180	.5385	33.8960	.6590	.6458	.6000	40.3268	.7348	.7200	.7368	48.2879	.7009	.7000
190	.4667	30.6813	.6360	.6067	.7500	54.6723	.8018	.8000	.8571	73.2227	.6547	.6316
200	.8000	28.0492	.8327	.8320	.2500	46.9155	.4756	.3918	.5000	36.5790	.6183	.6047
210	.5000	28.1452	.6667	.6400	.5000	27.0618	.6860	.6531	.8889	28.6032	.6172	.5783
220	.5833	25.5912	.7005	.6890	.6667	31.8385	.7493	.7442	.7500	37.7322	.7792	.7786
230	.6154	26.5485	.7781	.7572	.4286	33.9168	.6172	.5783	1	39.1419	.9487	.9474
240	.7143	24.8238	.8098	.8034	.6667	27.9864	.7493	.7442	.8500	23.5364	.5831	.5440
250	.6154	27.8362	.7321	.7212	.3750	17.4761	.5557	.5152	.5000	30.9204	.6427	.6230
260	.6667	23.5485	.7454	.7407	.5000	23.8084	.6183	.6047	.8182	28.3170	.5222	.4737
270	.6000	19.1912	.7160	.7050	.2000	14.1713	.4243	.3273	.8000	36.2411	.8718	.8686
280	.3158	22.2128	.5518	.4758	.3333	18.0515	.4924	.4571	.3750	17.7348	.5752	.5263
290	.5833	22.2191	.7071	.6942	.5000	22.8261	.6183	.6047	.6000	13.9849	.6928	.6857
300	.5789	21.8163	.6896	.6792	.5714	19.0177	.6667	.6588	.3333	21.2561	.5620	.4932

Table 14

Specificity (Spe.), TP, FN, FP, TN of ELM, CSELM, ICSELM (Hepatitis dataset).

L	ELM				CSELM				ICSELM						
	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN	Spe.	TP	FN	FP	TN
10	.9483	9	8	3	55	.8889	2	5	2	16	.9048	2	19	2	2
20	.9667	7	8	2	58	.8571	3	1	3	18	1	1	1	0	23
30	.8154	7	3	12	53	.8947	5	1	2	17	.8182	1	2	4	18
40	.8793	9	8	7	51	.7895	2	4	4	15	.9474	5	1	1	18
50	.7544	7	11	14	43	.6250	5	4	6	10	.7778	4	3	4	14
60	.6271	5	11	22	37	.8500	2	3	3	17	.8000	3	2	4	16
70	.6667	9	6	20	40	.6111	4	3	7	11	.7500	3	2	5	15
80	.4833	9	6	31	29	.8333	5	2	3	15	.7000	2	3	6	14
90	.8276	8	9	10	48	.8824	6	2	2	15	.6667	4	3	6	12
100	.6552	9	8	20	38	.9412	6	2	1	16	.8571	3	2	8	12
110	.8983	7	9	6	53	.8500	3	2	3	17	.8333	3	4	3	15
120	.7619	7	5	15	48	.9524	2	2	1	20	.8333	7	0	3	15
130	.7742	3	10	14	48	.7895	3	3	4	15	.8889	6	1	2	16
140	.8421	9	9	9	48	.8889	3	4	2	16	.7500	3	2	5	15
150	.7903	9	4	13	49	.8500	2	3	3	17	.8421	6	0	3	16
160	.8596	7	11	8	49	.4762	3	1	11	10	.6667	2	2	7	14
170	.8793	7	10	7	51	.8000	4	6	3	12	.9524	2	2	1	20
180	.8065	7	6	12	50	.9000	3	2	2	18	.6667	4	2	5	14
190	.8667	7	8	8	52	.8571	3	1	3	18	.5000	2	2	3	18
200	.8667	12	3	8	52	.9048	1	3	2	19	.7647	4	4	4	13
210	.8889	6	6	7	56	.9412	4	4	1	16	.4286	3	4	2	16
220	.8413	7	5	10	53	.8421	4	2	3	16	.8095	3	1	4	17
230	.9839	8	5	1	61	.8889	3	4	2	16	.9000	5	0	2	18
240	.9180	10	4	5	56	.8421	4	2	3	16	.4000	2	3	3	17
250	.8710	8	5	8	54	.8235	3	5	3	14	.8261	1	1	4	19
260	.8333	6	3	11	55	.7647	4	4	4	13	.3333	1	2	4	18
270	.8545	12	8	8	47	.9000	1	4	2	18	.9500	4	1	1	19
280	.9643	6	13	2	54	.7273	1	2	6	16	.8824	3	5	2	15
290	.8571	7	5	9	54	.7647	4	4	4	13	.8000	3	2	4	16
300	.8214	11	8	10	46	.7778	4	3	4	14	.9474	2	4	1	18

dataset are given in Table 1 and the overviews of datasets are presented in Sections 4.1–4.4

$$x_n = a + (b - a) * \frac{(x - X_{Min})}{X_{Max} - X_{Min}} \quad (20)$$

where x_n is the normalized form of original value x , a and b are -1 and 1 , X_{Min} and X_{Max} are the minimum and maximum of the dataset respectively.

4.1. Breast Cancer

The Breast Cancer dataset is obtained from University of Wisconsin hospital. It is a nine dimensional dataset with the following features: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitosis.

4.2. Diabetes

The dataset is referred as the diabetes in Pima Indian women. The data are collected by the US National Institute of Diabetes and Digestive and Kidney diseases.

The classes (1 for a healthy patient and 2 for a tested positive for a diabetes patient) have respectively 500 and 268 instances. Each class is having the following 8 attributes i.e. number of times pregnant, plasma glucose concentration a 2 h in an oral glucose tolerance test, diastolic blood pressure (mm Hg), triceps skin fold thickness (mm), 2-h serum insulin(μ U/ml), body mass index, diabetes pedigree function and age (in years).

Table 15

Max training accuracy, Max testing accuracy with respective hidden neurons for ELM, OSELM CSELM and ICSELM (Breast Cancer, Diabetes, Bupa and Hepatitis).

Dataset	ELM		OSELM		CSELM		ICSELM	
	Trg Acc	Tst Acc						
	Max	HN	Max	HN	Max	HN	Max	HN
Breast Cancer	.9675	1	230–300	.9979	.9649	20	1	.9707
Diabetes	.9265	.7721	20	.8649	.7479	80	.9245	.7838
Bupa	.6411	.8836	300	.9347	.6329	130	.8834	.7308
Hepatitis	.8267	.8400	20	.8495	.8120	20	1	.8680

Table 16

Testing accuracy for MLPBP, MLPCS MLPICS, RBFNNBP, RBFNNCS, RBFNNICS, ELM, OSELM CSELM, ICSELM (Breast Cancer, Diabetes, Bupa and Hepatitis).

Dataset	MLP BP	MLP CS	MLP ICS	RBFN NBP	RBFN NCS	RBFN NICS	ELM	OSELM	CSELM	ICSELM
Breast Cancer	.8600	.8914	.9003	.9423	.9491	.9571	1.00	.9649	.9707	.9777
Diabetes	.7278	.7600	.7637	.7492	.7653	.7645	.7721	.7479	.7838	.7850
Bupa	.6465	.6802	.6800	.7015	.7232	.7254	.8836	.6329	.7308	.7583
Hepatitis	.7400	.7600	.8125	.7953	.8349	.8476	.8400	.8120	.8680	1.00

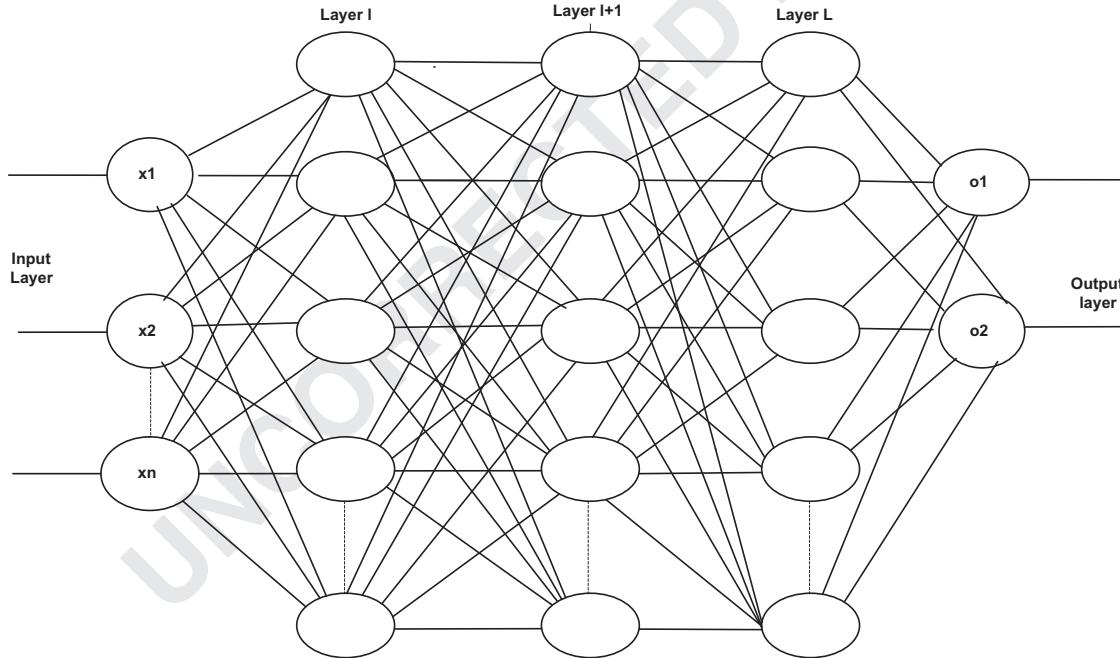


Fig. 65. Structure of Multilayer perceptron.

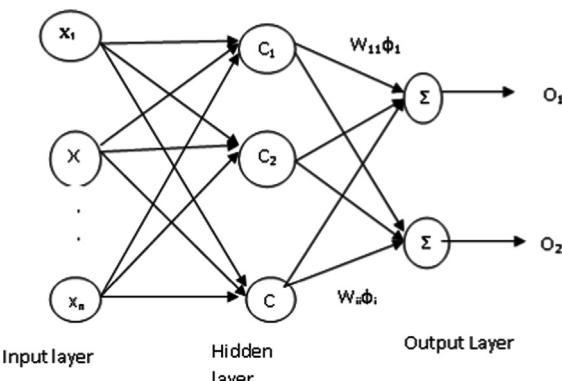


Fig. 66. Structure of radial basis function neural network.

4.3. Bupa

Bupa (liver disorder) is a binary dataset with 7 features. Feature or attributes are Mcv (mean corpuscular volume), Alkphos (alkaline phosphatase), Sgpt (alamine aminotransferase), Sgot (aspartate aminotransferase), Gammagt (gamma-glutamyl transpeptidase), Drinks (number of half-pint equivalents of alcoholic beverages drunk per day), Selector or class Id. The data are obtained from University of California at Irvine (UCI) Machine Learning Repository.

4.4. Hepatitis

This dataset identifies whether patients suffering from hepatitis are alive or not. It includes 155 samples and 19 features. Decision code is 1 for those alive and 0 for those dead.

1 5. Measures used to evaluate the performance of the models

2 Performance of all the classifiers discussed in this study is
 3 evaluated by different measures like training accuracy, testing
 4 accuracy, confusion matrix, Receiver Operating Characteristic
 5 curve (ROC), sensitivity, specificity, Gmean, and F-score.

6 5.1. Confusion matrix

7 Confusion matrix represents the values of TP, TN, FP, FN [75–77]
 8 considering actual positive-predicted positive, actual negative-
 9 predicted negative, actual negative-predicted positive and actual
 10 positive-predicted negative respectively. In other words they can
 11 be explained:

12 TP (True Positive) – correctly classified positive samples, TN
 13 (True Negative) – correctly classified negative samples, FP (False
 14 positive) – misclassified negative samples and FN (False negative)
 15 – misclassified positive samples.

16 5.2. Overall Accuracy

17 The Overall Accuracy [19,75] is a commonly used evaluation
 18 criterion for a standard classification system.

$$19 \text{Overall Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (21)$$

20 5.3. Sensitivity

21 Sensitivity [75] is also regarded as TPR (true positive rate) or
 22 recall. Sensitivity evaluates how well a model can recognize
 23 abnormal records. Higher the sensitivity is better the classifier
 24 becomes. It always identifies proportion of accurately classified
 25 samples to total samples. High value of TP means high value of
 26 sensitivity leading to high accuracy

$$27 \text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (22)$$

28 5.4. Specificity

29 Specificity is considered as the TNR (true negative rate).
 30 Specificity measures the ability of the proposed method to identify
 31 normal cases. It identifies accurately classified negative samples to
 32 total samples. If specificity becomes high like sensitivity and their
 33 difference is very less (< 1%), then the classifier is a better one

$$34 \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (23)$$

35 5.5. Gmean

36 Gmean [77] is used when the combined performance of the
 37 two classes need to be judged by a single metric. It measures both
 38 the sensitivity and specificity in a balanced manner. Gmean value
 39 cannot be high if the accuracy of one of the two classes is very low.

40 If all the samples are classified into majority class, Gmean value
 41 becomes 0

$$42 \text{Gmean} = \sqrt{\text{Sensitivity} * \text{Specificity}} \quad (24)$$

5.6. F-score	67
Like Gmean, F-score [76,77] is also an important metric to check the combined performance of the two classes	68
<i>F</i>	69
– score = $(2 * \text{Sensitivity} * \text{Specificity}) / (\text{Sensitivity} + \text{Specificity})$	70
	71
	72
	73
	74
5.7. ROC	75
Receiver operating characteristics (ROC) graph [74] is a useful tool for visualizing the classifiers and examining their quality in terms of performance. ROC curve is a graphical representation of sensitivity against (1-specificity).	76
5.8. Norm value	77
In order to check the stability of a system, the norm and condition value of the weight matrix is to be determined. Condition number is defined from the analysis of error bounds for a linear system $\text{AX}=\text{B}$, it can be represented as the product of two matrix norms	78
$\kappa(A) = A \cdot A^{-1} $	79
$\kappa(A) = A \cdot A^{-1} \geq AA^{-1} = I = 1$	80
$\kappa(A) = \infty$ for a singular matrix. $\kappa(A)$ value is large means the system is ill-conditioned and $\kappa(A)$ value is small means the system is well conditioned [55].	81
6. Experimental results	82
All the classifiers taken in this study are implemented in the following environment, Operating System: Windows XP professional, CPU: Intel Core i3-370M (2.4 GHz), and Memory: 2 GB RAM.	83
In ELM, Sigmoid, Sine, Hard limit, Triangular and Radial basis function are used as the activation functions. However, in this study, sigmoidal function $g(x) = 1/(1 + \exp(-x))$ is used as the activation function for all the models. The range of the number of hidden nodes in ELM is varying from 10 to 300 with an increment of 10 neurons each time (Table 2).	84
6.1. Experiment 1: Breast Cancer dataset	85
In experiment 1, Breast Cancer dataset is taken into consideration. Figs. 2–5 show training and testing accuracy of ELM, OSELM, CSELM and ICSELM. Fig. 6 presents training and testing time required for training of ELM. Sensitivity of ELM, CSELM and ICSELM is shown in Figs. 7–9. Figs. 10–12 present sensitivity vs specificity vs Gmean for ELM, CSELM and ICSELM and Fig. 13 presents sensitivity vs specificity vs F-score of (ICSELM). Figs. 14–16 give ROC for ELM, CSELM and ICSELM. Figs. 17–19 give box plot for ELM, CSELM and ICSELM. Figs. 20–22 show number of hidden neurons vs testing accuracy vs norm for ELM, CSELM and ICSELM. Fig. 23 shows the comparison of norm value for ELM, CSELM, ICSELM. The time required for training (Trg Time), training accuracy (Trg Acc), testing accuracy (Tst Acc) and standard deviation of testing accuracy (Tst Dev) for ELM, OSELM, CSELM and ICSELM appear in Table 3. In all the tables, L stands for hidden neurons. The sensitivity, norm, Gmean and F-score values are demonstrated in Table 4 for ELM, CSELM and ICSELM. Table 5 provides specificity, TP, FN, FP and TN for ELM, CSELM and ICSELM.	86

1 6.2. Experiment 2: Diabetes dataset

2 Diabetes data is classified using all the models in experiment 2.

3 Figs. 24–27 show training and testing accuracy of ELM, OSELM,

4 CSELM and ICSELM. Fig. 28 presents training and testing time

5 required for training of ELM. Sensitivity of ELM, CSELM and

6 ICSELM is shown in Figs. 29–31. Figs. 32–34 present sensitivity

7 vs specificity vs Gmean for ELM, CSELM and ICSELM. Figs. 35–37

8 show ROC and Figs. 38–40 depict box plot for ELM, CSELM and

9 ICSELM. Figs. 41–43 show number of hidden neurons vs testing

10 accuracy vs norm. The comparison of norm value for ELM, CSELM,

11 ICSELM is shown in Fig. 44. The time required for training (Trg

12 Time), training accuracy (Trg Acc), testing accuracy (Tst Acc) and

13 standard deviation of testing accuracy (Tst Dev) for ELM, OSELM,

14 CSELM and ICSELM are given in Table 6. The sensitivity, norm,

15 Gmean and F-score value appear in Table 7 for ELM, CSELM,

16 ICSELM. Table 8 presents specificity, TP, FN, FP and TN for ELM,

17 CSELM and ICSELM.

20 6.3. Experiment 3: Bupa dataset

21 Bupa dataset is classified using all the models as classifiers in

22 experiment 3. Figs. 45–48 show training and testing accuracy of

23 ELM, OSELM, CSELM and ICSELM whereas Fig. 49 presents training

24 and testing time required for training of ELM. Figs. 50–52 present

25 sensitivity of ELM, CSELM and ICSELM. Figs. 53–55 present

26 sensitivity vs specificity vs Gmean for ELM, CSELM and ICSELM.

27 Figs. 56 and 57 depict ROC for ELM and ICSELM and Figs. 58–60

28 give box plot for ELM, CSELM and ICSELM. Figs. 61–63 show no. of

29 hidden neurons vs testing accuracy vs norm for ELM, CSELM and

30 ICSELM. The comparison of norm value for ELM, CSELM, ICSELM is

31 shown Fig. 64. The time required for training (Trg Time), training

32 accuracy (Trg Acc), testing accuracy (Tst Acc) and standard deviation

33 of testing accuracy (Tst Dev) for ELM, OSELM, CSELM and

34 ICSELM appear in Table 9. The sensitivity, norm, Gmean and F-

35 score value are demonstrated in Table 10 for all the models.

36 Specificity, TP, FN, FP and TN for ELM, OSELM, CSELM and ICSELM

37 are shown in Table 11.

39 6.4. Experiment 4: Hepatitis dataset

40 Similar with the other three experiments, Hepatitis dataset is

41 also classified using all the models. However, due to space

42 constraint the graphs are not given. The time required for training

43 (Trg Time), training accuracy (Trg Acc), testing accuracy (Tst Acc)

44 and standard deviation of testing accuracy (Tst Dev) for ELM,

45 OSELM, CSELM and ICSELM appear in Table 12. The sensitivity,

46 Norm, Gmean and F-score value are demonstrated in Table 13 for

47 all the models. Specificity, TP, FN, FP and TN for ELM, OSELM,

48 CSELM and ICSELM are shown in Table 14.

53 7. Result discussions

54 In this paper, a robust meta-heuristic evolutionary algorithm

55 based ICSELM model is used to classify four benchmark medical

56 datasets: Breast Cancer, Diabetes, Bupa, Hepatitis. The obtained

57 results are compared with nine other models: ELM, OSELM,

58 CSELM, MLPBP, MLPCS, MLPICS, RBFNNBP, RBFNNCS and

59 RBFNNICS. A set of performance evaluation measures like, training

60 accuracy, testing accuracy, confusion matrix, receiver operating

61 characteristic curve (ROC), sensitivity, specificity, Gmean, and

62 F-score have been taken to compare and test the accuracy, stability

63 and robustness of the proposed model. Out of them training and

64 testing accuracy is the most important one. Summary analysis of

65 maximum training accuracy and maximum testing accuracy with

66 respective hidden neurons of testing accuracy for all four ELM

67 based models is presented in Table 15.

68 It is clearly seen from the above table that CSELM and ICSELM

69 require less number of hidden neurons than that of ELM and

70 OSELM to yield either same or similar testing accuracy. It suggests

71 the superiority of CSELM and ICSELM. Further, when CSELM and

72 the proposed ICSELM are compared, ICSELM gives better results in

73 terms of testing accuracy with respect to number of hidden

74 neurons. It is also compared with other artificial neural network

75 (ANN) based models and the result appearing in Table 16 proves

76 its superiority. Similarly, the better performance of the ICSELM can

77 easily be seen from the three dimensional plots of sensitivity vs

78 specificity vs Gmean of all the models with three datasets

79 (Figs. 10–12, 32–34, and 53–55). In case of ICSELM, the sensitivity

80 is more than that of other ELM based models in respect of three

81 experiments – Breast Cancer, Bupa and Hepatitis – out of four

82 experiments.

83 In respect of ROC, the ICSELM model performs better than that

84 of other models as can be seen in graphs (Figs. 14–16, 35–37, and

85 56–57).

86 In case of ICSELM, the box plots (Figs. 17–19, 38–40, and 58–60)

87 show that TP, TN exceed FP and FN in respect of all datasets. But it

88 is not seen always in respect of other models discussed in

89 this study.

90 Graphs (Figs. 23, 44 and 64) and Tables 4, 7, and 10 show that

91 ICSELM produces less norm of output weights as compared to

92 other models.

93 From the above discussion it is clearly demonstrated that ELM

94 based models such as ELM, OSELM, CSELM and ICSELM are better

95 classifiers as compared to ANN based models like MLPBP, MLPCS,

96 MLPICS, RBFNNBP, RBFNNCS and RBFNNICS. Further, when com-

97 parison is being made between the ELM based models, ICSELM

98 proves to be the best one.

8. Conclusion

99 In this paper, ICSELM is proposed for classification of four bench-

100 mark medical datasets. Other three ELM based models such as

101 simple ELM, OSELM, CSELM and six ANN based models – MLPBP,

102 MLPCS, MLPICS, RBFNNBP, RBFNNCS and RBFNNICS – are also

103 discussed and compared. A series of empirical studies has been

104 conducted to evaluate the efficacy of all the models. Different

105 performance measures have been adopted in order to provide a fair

106 comparison among all the models discussed in this paper. From the

107 results it can be concluded that compared to ANN models all the ELM

108 based classifiers show better performance as they map the features

109 to a higher dimensional space. However, in order to get more stable

110 and robust classifier, ELM is combined with CS and ICS. The findings

111 reveal that ICSELM restrict very efficiently the ill-condition problem

112 and that leads to better performance in comparison with simple ELM,

113 OSELM and CSELM. This work indicates that the proposed ICSELM

114 model can effectively be used for medical data classification tasks.

119 Appendix A

120 The Multilayer perceptron (MLP) [15,78] consists of multiple

121 layers of fully connected neurons that interact among themselves

122 using weighted connections. Generally in MLP, any number of

123 hidden layers may appear in between input and output layer. But

124 in this study, two hidden layers MLP is considered for classification

125 of medical datasets. For classification using MLP, initially the

126 training and testing datasets are prepared. The input vector, $\{x_1,$

127 $x_2, \dots, x_n\}$ is supplied to the input layer. The desired output is

128 supplied by the teacher [78]. Considering a two hidden layer

129

1 MLP as shown in Fig. 65, the total input, x_j^{l+1} received by neuron j
 2 in layer $l+1$ is defined as
 3

$$x_j^{l+1} = \sum_i y_i^l w_{ji}^l \quad (\text{A.1})$$

6 where y_i^l represents the i th neuron in the preceding l th layer, w_{ji}^l is
 7 the weight of the connection from the i th neuron in layer l to the
 8 j th neuron in layer $l+1$. The output of a neuron is represented as a
 9 nonlinear sigmoid activation function of its total input and is given
 10 as
 11

$$y_j^l = \frac{1}{1 + e^{-x_j^l}} \quad (\text{A.2})$$

15 The outputs for all nodes in the input layer
 16

$$y_j^0 = x_j^0 \quad (\text{A.3})$$

19 where x_j^0 is the j th component of the input vector at the input
 20 layer. The set of all internal weights of the hidden units is to be
 21 determined by the back propagation learning algorithm. The Least
 22 Mean Square (LMS) error corresponding to output vectors and
 23 network weight vector ' w ' is given as
 24

$$E(w) = \frac{1}{2} \sum_{j,s} (y_{j,s}^l(w) - d_{j,s})^2 \quad (\text{A.4})$$

27 where $y_{j,s}^l(w)$ is the output for node j in l th layer for s th input-
 28 output case and $d_{j,s}$ is the desired output. The gradient-descent
 29 method is applied to minimize $E(w)$ and a series of weight updates
 30 are done using the formula
 31

$$\Delta w_{ji}^l(t) = -\varepsilon + \alpha \Delta w_{ji}^l(t-1) \quad (\text{A.5})$$

34 where ε is a positive constant, $0 \leq \alpha \leq 1$ represents the damping
 35 coefficient of momentum. The details about MLP are discussed in
 36 paper [78]. Further, to overcome the local minima problem faced
 37 by backpropagation (BP) algorithm and to get more classification
 38 accuracy, MLP is combined with CS and ICS algorithm and the
 39 results are compared with the proposed model.
 40

42 Appendix B

44 Radial basis function neural network (RBFNN) [16,79] is a three
 45 layers architecture, input, hidden and output layer. The structure
 46 of RBFNN is given in Fig. 66. In this model, a set of n neurons is
 47 considered for the input layer, which accepts the elements of an n -
 48 dimension input vector, $X = (x_1, x_2, x_3, \dots, x_n)$. The neurons in input
 49 layer are fully connected to the hidden layer neurons i.e. j hidden
 50 neurons, which are known as radial basis function units. These
 51 units are connected directly to all elements in the output layer. In
 52 this paper two output neurons are there for two classes. In RBFNN
 53 model, each radial basis function unit in hidden layer contains a
 54 radial basis function unit i.e. a decision-making model.
 55

56 For an input vector $x(t)$, the output $o_j(t)$ of j th node produced by
 57 RBFNN is

$$o_j(t) = \sum_{i=1}^h w_{ij} \phi_i(t) = \sum_{i=1}^h w_{ij} e^{-\|x(t) - c_i(t)\|^2 / 2\sigma_i^2} \quad (\text{B.1})$$

61 where c_i is center of the i th hidden node, σ_i is the variance and h is
 62 the total number of hidden neurons. Using vector notation, let $u =$
 63 $(\phi_1(t), \phi_2(t), \dots, \phi_h(t))$ and $w_j = (w_{1j}, w_{2j}, \dots, w_{hj})$. The RBFNN output
 64 can be written as
 65

$$o_j = w_j^T(t) * u \quad (\text{B.2})$$

67 The cost function of the network for the j th neuron is then
 68 calculated as
 69

$$e_j(t)^2 = (d_j(t) - o_j(t))^2 \quad (\text{B.3})$$

71 where $d_j(t)$ is desired output of the j th neuron. All the parameters,
 72 c_i , σ_i and w_{ij} are updated using backpropagation, CS and ICS
 73 learning algorithms and the obtained results are compared with
 74 the proposed model. The details about this model can be seen in
 75 Ref. [79].
 76

77 References

- [1] J.L. Blue, G.T. Candela, P. Grother, R. Chellappa, C.L. Wilson, Evaluation of pattern classifiers for fingerprint and OCR applications, Pattern Recognit. 27 (4) (1994) 485–501.
- [2] D. Merkl, Text classification with self-organizing maps: some lessons learned, Neurocomputing 21 (1) (1998) 61–77.
- [3] J. Blasco, N. Aleixos, E. Moltó, Machine vision system for automatic quality grading of fruit, Biosyst. Eng. 85 (4) (2003) 415–423.
- [4] E. Ngai, Y. Hu, Y. Wong, Y. Chen, X. Sun, The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature, Decis. Support Syst. 50 (3) (2011) 559–569.
- [5] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, Learning to construct knowledge bases from the world wide web, Artif. Intell. 118 (1) (2000) 69–113.
- [6] G.-M. Chen, B.-S. Lee, O. Rui, Foreign ownership restrictions and market segmentation in China's stock markets, J. Financ. Res. 24 (1) (2001) 133–155.
- [7] E.W. Ngai, L. Xiu, D.C. Chau, Application of data mining techniques in customer relationship management: a literature review and classification, Expert Syst. Appl. 36 (2) (2009) 25922602.
- [8] T. Helmy, Z. Rasheed, Multi-category bioinformatics dataset classification using extreme learning machine, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC'09, 2009, pp. 3234–3240.
- [9] D. Wang, G.-B. Huang, Protein sequence classification using extreme learning machine, in: Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN'05, Vol. 3, 2005, pp. 1406–1411.
- [10] A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recognit. 30 (7) (1997) 1145–1159.
- [11] C.-K. Siew, G.-B. Huang, Q.-Y. Zhu, Extreme learning machine: a new learning scheme of feed forward neural networks, in: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN2004), 2004.
- [12] S.L. Salzberg, On comparing classifiers: pitfalls to avoid and a recommended approach, Data Min. Knowl. Discov. 1 (3) (1997) 317–328.
- [13] P.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
- [14] D.A. Coast, R.M. Stern, G.G. Cano, S.A. Briller, An approach to cardiac arrhythmia analysis using hidden Markov models, IEEE Trans. Biomed. Eng. 37 (9) (1990) 826–836.
- [15] V. Ravi, M. Krishna, A new online data imputation method based on general regression auto associative neural network, Neurocomputing 138 (2014) 106–113.
- [16] F. Fernández-Navarro, C. Hervás-Martínez, R. Ruiz, J.C. Riquelme, Evolutionary generalized radial basis function neural networks for improving prediction accuracy in gene classification using feature selection, Appl. Soft Comput. 12 (6) (2012) 1787–1800.
- [17] E.K. Aydogan, I. Karaoglan, P.M. Pardalos, HGA: hybrid genetic algorithm in fuzzy rule-based classification systems for high-dimensional problems, Appl. Soft Comput. 12 (2) (2012) 800–806.
- [18] T. Wilk, M. Wozniak, Soft computing methods applied to combination of one-class classifiers, Neurocomputing 75 (1) (2012) 185–193.
- [19] V. Malathi, N. Marimuthu, S. Baskar, Intelligent approaches using support vector machine and extreme learning machine for transmission line protection, Neurocomputing 73 (10) (2010) 2160–2167.
- [20] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.
- [21] A. Bahrololoum, H. Nezamabadi-Pour, H. Bahrololoum, M. Saeed, A prototype classifier based on gravitational search algorithm, Appl. Soft Comput. 12 (2) (2012) 819–825.
- [22] J. Chen, H. Huang, S. Tian, Y. Qu, Feature selection for text classification with Naïve Bayes, Expert Syst. Appl. 36 (3) (2009) 5432–5435.
- [23] P.K. Srivastava, D. Han, M.A. Rico-Ramirez, M. Bray, T. Islam, Selection of classification techniques for land use/land cover change investigation, Adv. Space Res. 50 (9) (2012) 1250–1265.
- [24] Y. Shao, R.S. Lunetta, Comparison of support vector machine, neural network, and cart algorithms for the land-cover classification using limited training data points, ISPRS J. Photogramm. Remote Sens. 70 (2012) 78–87.
- [25] D.E. Brown, V. Corruble, C.L. Pittard, A comparison of decision tree classifiers with backpropagation neural networks for multimodal classification problems, Pattern Recognit. 26 (6) (1993) 953–961.
- [26] R. Khan, A. Hanbury, J. Stöttinger, A. Bais, Color based skin classification, Pattern Recognit. Lett. 33 (2) (2012) 157–163.

- 1 [27] D.W. Aha, Tolerating noisy, irrelevant and novel attributes in instance-based
2 learning algorithms, *Int. J. Man Mach. Stud.* 36 (2) (1992) 267–287.
- 3 [28] Wouter F., Schmidt, Martin A. Kraaijveld, and Robert P.W. Duin, Feedforward
4 neural networks with random weights, in: Proceedings of the IEEE 11th IAPR
5 International Conference on Pattern Recognition, Vol. II. Conference B: Pattern
6 Recognition Methodology and Systems, 1992.
- 7 Q5 [29] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive
8 networks, *Complex Syst.* (1988).
- 9 Q6 [30] C.P. Chen, John Z. Wan, A rapid learning and dynamic stepwise updating
10 algorithm for flat neural networks and the application to time-series prediction,
11 *IEEE Trans. Syst. Man Cybern. Part B: Cybernet.* 29 (1) (1999) 62–72.
- 12 [31] Boris Igelnik, Y.-H. Pao, Stochastic choice of basis functions in adaptive
13 function approximation and the functional-link net, *IEEE Trans. Neural Netw.* 6 (6) (1995)
14 1320–1329.
- 15 [32] Yoh-Han Pao, Gwang-Hoon Park, Dejan J. Sobajic, Learning and generalization
16 characteristics of the random vector functional-link net, *Neurocomputing* 6 (2) (1994)
17 163–180.
- 18 [33] CL Philip. Chen, A rapid supervised learning neural network for function
19 interpolation and approximation, *IEEE Trans. Neural Netw.* 7 (5) (1996)
20 1220–1230.
- 21 [34] L.P. Wang, C.R. Wan, Comments on ‘the extreme learning machine’, *IEEE Trans.*
22 *Neural Netw.* 19 (8) (2008) 1494–1495.
- 23 [35] Zhihui Wang, et al., A high accuracy pedestrian detection system combining a
24 cascade AdaBoost detector and random vector functional-link net, *Sci. World*
25 *J.* 2014 (2014).
- 26 [36] Guang-Bin Huang, What are extreme learning machines? Filling the gap
27 between Frank Rosenblatt’s dream and John von Neumann’s puzzle, *Cogn.*
28 *Comput.* 7 (3) (2015) 263–278.
- 29 [37] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for
30 regression and multiclass classification, *IEEE Trans. Syst. Man Cybern. Part B:*
31 *Cybern.* 42 (2) (2012) 513–529.
- 32 [38] G.-B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, *Int. J.*
33 *Mach. Learn. Cybern.* 2 (2) (2011) 107–122.
- 34 Q7 [39] D. Wang, M. Alhamdoosh, Evolutionary extreme learning machine ensembles
35 with size control, *Neurocomputing* 102 (2013) 98–110.
- 36 [40] G.-B. Huang, D. Wang, Advances in extreme learning machines (ELM 2010),
37 *Neurocomputing* 74 (16) (2011) 2411–2412.
- 38 [41] G. Feng, G.-B. Huang, Q. Lin, R. Gay, Error minimized extreme learning
39 machine with growth of hidden nodes and incremental learning, *IEEE Trans.*
40 *Neural Netw.* 20 (8) (2009) 1352–1357.
- 41 [42] W.-Y. Deng, Q.-H. Zheng, L. Chen, X.-B. Xu, Research on extreme learning of
42 neural networks, *Chin. J. Comput.* 33 (2) (2010) 279–287.
- 43 [43] S. Ding, X. Xu, R. Nie, Extreme learning machine and its applications, *Neural*
44 *Comput. Appl.* (2013) 1–8.
- 45 [44] L. Wang, Y. Huang, X. Luo, Z. Wang, S. Luo, Image deblurring with filters
46 learned by extreme learning machine, *Neurocomputing* 74 (16) (2011)
47 2464–2474.
- 48 [45] Z.-L. Sun, T.-M. Choi, K.-F. Au, Y. Yu, Sales forecasting using extreme learning
49 machine with applications in fashion retailing, *Decis. Support Syst.* 46 (1)
50 (2008) 411–419.
- 51 [46] A. Grigorievskiy, Y. Miche, A.-M. Ventelä, E. Séverin, A. Lendasse, Long-term
52 time series prediction using OP-ELM, *Neural Netw.* 51 (2014) 50–56.
- 53 [47] H. Yang, J. Yi, J. Zhao, Z. Dong, Extreme learning machine based genetic
54 algorithm and its application in power system economic dispatch, *Neuro-*
55 *computing* 102 (2013) 154–162.
- 56 [48] R. Zhang, Z.Y. Dong, Y. Xu, K. Meng, K.P. Wong, Short-term load forecasting of
57 Australian national electricity market by an ensemble model of extreme
58 learning machine, *IET Gener. Transm. Distrib.* 7 (4) (2013) 391–397.
- 59 [49] K. Siah Yap, H. Jen Yap, Daily maximum load forecasting of consecutive
60 national holidays using OSELM based multi-agents system with weighted
61 average strategy, *Neurocomputing* 81 (2012) 108–112.
- 62 [50] Q. He, X. Jin, C. Du, F. Zhuang, Z. Shi, Clustering in extreme learning machine
63 feature space, *Neurocomputing* 128 (2014) 88–95.
- 64 [51] A. Bueno-Crespo, P.J. García-Laencina, J.-L. Sancho-Gómez, Neural architecture
65 design based on extreme learning machine, *Neural Netw.* 48 (2013) 19–24.
- 66 [52] J. Kim, H.S. Shin, K. Shin, M. Lee, Robust algorithm for arrhythmia classification
67 in ECG using extreme learning machine, *Biomed. Eng. Online* 8 (2009) 31.
- 68 [53] R. Zhang, G.-B. Huang, N. Sundararajan, P. Saratchandran, Multicategory
69 classification using an extreme learning machine for microarray gene
70 expression cancer diagnosis, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 4 (3)
71 (2007) 485–495.
- 72 [54] O. Fink, E. Zio, U. Weidmann, et al., Extreme learning machines for predicting
73 operation disruption events in railway systems, in: Proceedings of the
74 European Safety and Reliability Conference, 2013, pp. 1–8.
- 75 [55] G. Zhao, Z. Shen, C. Miao, Z. Man, On improving the conditioning of extreme
76 learning machine: a linear case, in: Proceedings of the IEEE 7th International
77 Conference on Information, Communications and Signal Processing (ICICS),
78 2009, pp. 1–5.
- 79 [56] G. Zhao, Z. Shen, Z. Man, Robust input weight selection for well-conditioned
80 extreme learning machine, *Int. J. Inf. Technol.* 17 (1) (2011) 1–13.
- 81 [57] L.D. Pacifico, T.B. Ludermir, Improved evolutionary extreme learning machines
82 based on particle swarm optimization and clustering approaches, *Int. J. Natl.*
83 *Comput. Res.* 3 (3) (2012) 1–20.
- 84 [58] F. Han, H.-F. Yao, Q.-H. Ling, An improved evolutionary extreme learning
85 machine based on particle swarm optimization, *Neurocomputing* 116 (2013)
86 87–93.
- 87 [59] G.-B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning
88 machine for classification, *Neurocomputing* 74 (1) (2010) 155–163.
- 89 [60] F. Han, H.-F. Yao, Q.-H. Ling, An improved extreme learning machine based on
90 particle swarm optimization, in: *Bio-Inspired Computing and Applications*,
91 Springer, 2012, pp. 699–704.
- 92 [61] Q.-Y. Zhu, A.K. Qin, P.N. Suganthan, G.-B. Huang, Evolutionary extreme learning
93 machine, *Pattern Recognit.* 38 (10) (2005) 1759–1763.
- 94 [62] X.-S. Yang, S. Deb, Multiobjective Cuckoo Search for Design Optimization, Vol.
95 40, Elsevier (2013) 1616–1624.
- 96 [63] I. Fister Jr., X.-S. Yang, D. Fister, I. Fister, Cuckoo Search: A Brief Literature
97 Review, *Cuckoo Search and Firefly Algorithm*, Springer (2014) 49–62.
- 98 [64] X.-S. Yang, *Nature-Inspired Meta-Heuristic Algorithms*, Luniver Press, 2010.
- 99 [65] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of the IEEE
100 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009,
101 pp. 210–214.
- 102 [66] X.-S. Yang, S. Deb, Engineering optimisation by cuckoo search, *Int. J. Math.*
103 *Model. Numer. Optim.* 1 (4) (2010) 330–343.
- 104 [67] R. Rajabioun, Cuckoo optimization algorithm, *Appl. Soft Comput.* 11 (8) (2011)
105 5508–5518.
- 106 [68] P. Civicioglu, E. Besdok, A conceptual comparison of the cuckoo-search,
107 particle swarm optimization, differential evolution and artificial bee colony
108 algorithms, *Artif. Intell. Rev.* 39 (4) (2013) 315–346.
- 109 [69] M. Marichelvam, An improved hybrid cuckoo search (IHCS) meta-heuristics
110 algorithm for permutation flow shop scheduling problems, *Int. J. Bio-Inspired*
111 *Comput.* 4 (4) (2012) 200–205.
- 112 [70] E. Valian, S. Mohanna, S. Tavakoli, Improved cuckoo search algorithm for feed
113 forward neural network training, *Int. J. Artif. Intell. Appl.* 2 (3) (2011) 36–43.
- 114 [71] Y. Ye, S. Squartini, F. Piazza, Online sequential extreme learning machine in
115 non-stationary environments, *Neurocomputing* 116 (2013) 94–101.
- 116 [72] J.-S. Lim, S. Lee, H.-S. Pang, Low complexity adaptive forgetting factor for
117 online sequential extreme learning machine (OSELM) for application to non-
118 stationary system estimations, *Neural Comput. Appl.* 22 (3–4) (2013) 569–576.
- 119 [73] Y. Lan, Y.C. Soh, G.-B. Huang, Ensemble of online sequential extreme learning
120 machine, *Neurocomputing* 72 (13) (2009) 3391–3395.
- 121 [74] A. Slaby, ROC analysis with MATLAB, in: Proceedings of the IEEE 29th Inter-
122 national Conference on Information Technology Interfaces (ITI), 2007, pp. 191–
123 196.
- 124 [75] Abdulhamit Subasi, Automatic recognition of alertness level from EEG by
125 using neural network and wavelet coefficients, *Expert Syst. Appl.* (2004) 1–11.
- 126 [76] César Ferri, José Hernández-Orallo, R. Modroiu, An experimental comparison of
127 performance measures for classification, *Pattern Recognit. Lett.* 30 (1) (2009)
128 27–38.
- 129 [77] D. Bowes, T. Hall, & D. Gray, Comparing the performance of fault prediction
130 models which report multiple performance measures: recomputing the
131 confusion matrix, in: Proceedings of the 8th International Conference on
132 Predictive Models in Software Engineering, ACM, 2012, pp. 109–118.
- 133 [78] S.K. Pal, S. Mitra, Multilayer perceptron, fuzzy sets, and classification, *IEEE*
134 *Trans. Neural Netw.* 3 (5) (1992) 683–697.
- 135 [79] S. Ding, G. Ma, Z. Shi, A novel self-adaptive extreme learning machine based
136 on affinity propagation for radial basis function neural network, *Neural*
137 *Comput. Appl.* 24 (7–8) (2014) 1487–1495.