



Improved Genetic Optimized Feature Selection for Online Sequential Extreme Learning Machine

Archana P. Kale^{1*}, Sumedh Sonawane², Revati M. Wahul¹, Manisha A. Dudhedia³

¹ Department of Computer Engineering, M. E. S. College of Engineering, S.P. Pune University, Pune 411001, India.

² Department of Computer Engineering, G. S. M. College of Engineering, S.P. Pune University, Pune 411045, India

³ Department of Electronics and Telecommunication, Marathwada Mitra Mandal's College of Engineering, S. P. Pune University, Pune 411052, India

Corresponding Author Email: archana.kale@mescoepune.org

<https://doi.org/10.18280/isi.270519>

ABSTRACT

Received: 8 August 2021

Accepted: 20 June 2022

Keywords:

genetic algorithm, online sequential extreme learning machine, pattern classification problem, feature selection problem, random search strategy

Extreme learning machine (ELM) is a rapid classifier, evolved for batch learning mode which is not suitable for sequential input. As retrieving of data from new inventory which is leads to time extended process. Therefore, online sequential ELM (OSELM) algorithm is progressed to handle the sequential input in which data is read 1 by 1 or chunk by chunk mode. The overall system generalization performance may devalue because of the amalgamation of random initialization of OS-ELM and the presence of redundant and irrelevant features. To resolve the said problem, this paper proposes a correspondence improved genetic optimized feature selection paradigm for sequential input (IG-OSELM) for radial basis or function by using clinical datasets. For performance comparison, the proposed paradigm experimented and evaluated for ELM, improved genetic optimized for ELM classifier (IG-ELM), OS-ELM, IG-OSELM. Experimental results are calculated and analyzed accordingly. The comparative results analysis illustrates that IG-ELM provides 10.94% improved accuracy with 43.25% features as compared to ELM.

1. INTRODUCTION

Now a days artificial intelligence is the growing and critical area [1]. Feature subset selection (FSS) is an intricate procedure in the fields of artificial intelligence and machine learning. The prime objective of feature selection is to adopt the optimal features for further evaluation. The features which are relevant non-redundant are called as optimal features. It is entangled to decide the significance of features [2]. To enhance the system generalization performance, it is necessary to search and finalize only the optimal features.

Various FSS algorithms like Half selection, Neural Network for threshold, Mean Selection etc. are used for FSS. Random search based genetic algorithm is the random searching optimization technique which uses to select an optimal feature subset [3]. The big search space is handled by GA very effectively [4] and has a maximum chance of a global optimal solution.

Machine learning is one of the important sub-areas of Artificial intelligence [1]. Feature subset selection (FSS) is an intricate process in the fields of machine learning and data mining. The key objective of FSS is to provide the same or improved classification accuracy with a minimum number of relevant and non-redundant features instead of using all features. It is very intricate to decide the importance of and hence requirement of features without any prior information [2]. Hence, a large number of features are usually included in the input dataset, which include all types of features like relevant, irrelevant, bad and redundant etc. Perhaps, only relevant and non-redundant features are required for classification and also for improvement of the system

generalization performance. However, in many real time applications, it may be possible that the redundant or irrelevant features may become relevant while functioning jointly with other features, which makes it one of the most critical tasks to appropriately discriminate these features.

Extreme Learning Machine is rapid classifiers which have various advantages like good generalization performance, high speed, require less training time, etc. ELM is primarily designed for batch mode in which all data is available before training. However, it is not suitable for sequential input [5]. Therefore, OS-ELM is designed by Liang et al. for sequential input. Zhu et al. [6] developed Evolutionary ELM and Han et al. [7] developed the particle swarm optimization based Evolutionary ELM. In many papers, for ELM, sig activation function is used [8, 9]. Huang et al. [10] designed Incremental – ELM. ELM is also used to solve real time applications like medical data classification [11-13], universal approximation [14], and big data [15].

The original Extreme learning machine (ELM) is primarily designed for batch mode. Liang et al. [5] emerge online sequential – ELM (OS-ELM) for linear, incremental or sequential input. As ELM and OS-ELM calculates the input to hidden layer neurons by randomly assigning the specified input weights and biases, and the target output is calculated [11, 12] by analytically evaluation the weights in between the hidden layer and the resulted layer as shown in Figure 1. Thus, the generalization performance of the system may deteriorate due to the random initialization. One of the most significant steps is required, i.e., optimal feature subset selection.

The key intent of this paper is the innovative use of a genetic algorithm with an improved optimization approach for OS-

ELM (IG-OSELM) for clinical datasets. In various papers, the authors evaluated OSELM only by changing hidden nodes, but an extensive literature will break down to recognize the changes in the inceptive training data (block) according to the quantity of hidden nodes.

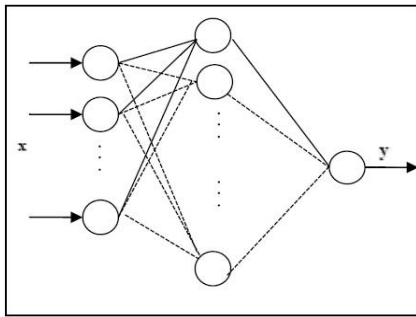


Figure 1. Basic ELM architecture

The structure of the paper is organized as follows: The detail structure of the proposed methodology of IG-OSELM with the aid of the paradigm is detailed in Section 2. Innovative results and comparison of results are mentioned in Section 3. Finally, the future works in combination with conclusions are described in Section 4.

2. PROPOSED IG-OSELM APPROACH

The paradigm of the proposed IG-OSELM approach is as shown in Figure 2, which is categorized into threefold subsystems – a. Pre-processing subsystem, b. FSS subsystem 3. Classification Subsystem.

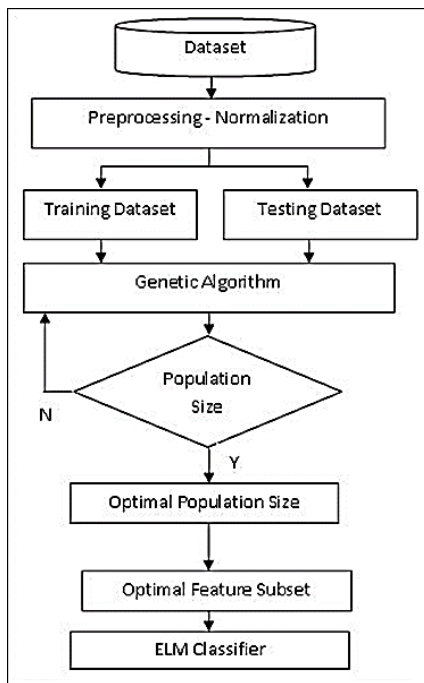


Figure 2. Proposed IG-OSELM paradigm

2.1 Datasets

The various datasets like Pima Indian Diabetes (PID), Statlog heart disease (SHD), Breast Cancer (BC), Australian (AS) [15, 16] are used. The dimensional scope of these

datasets is from 8 to 36. Most of the considered datasets are clinical datasets which are standard UCI repository datasets. PID, SHD, BC and AS datasets contain 8, 13, 10, 14 attributes and 768, 270, 699, 690 instances, respectively.

2.2 Preprocessing subsystem

Data Normalization is used in preprocessing subsystem. Data normalization is an intricate preprocessing method which is used in various artificial intelligence and machine learning algorithms [17]. The features present in the dataset are of different scales. Thus, it becomes very critical to handle such type of vector space as it contains the maximum range account of a vital task to convert all vector spaced features to unit space features which result lies between zero and one.

After normalization, all datasets are further divided into duplet parts, i.e., a training set in which 70% instances are considered and the remaining 30% instances are considered for the testing set. For example, the total number of instances present in PID dataset are 768, which are further divided into 538 instances that are used as training set and 230 instances are used for testing set. Same for SHD dataset, the total number of instances present in SHD dataset are 270, which are further divided into 189 instances are used as training set and 81 instances are used for testing set. For BC dataset, total number of instances present are 699, which are further divided into 490 instances that are used as training set and 209 instances are used for testing set. For AS dataset, the total number of instances present are 690, which are further divided into 483 instances that are used as training set and 207 instances are used for testing set.

2.3 Feature subset selection subsystem

Thousands of features are present in the high dimensional dataset. For classification, all features are not required as it may be the presence of non-optimal features, i.e., irrelevant and redundant features. These features act as noise which degrades the predictive accuracy.

Genetic Algorithm (GA) is relevant technique for selecting ideal features. GA contains the population which is a collection of a set of possible solutions to solve the problem [18]. Three steps like selection, evaluation, and recombination are executed in every iterative step. Selection, crossover and mutation are the genetic operators mainly used in GA. The number of iterations is depended on the condition of termination. Based on the quality, the fitness function is evaluated. And based on the evaluated fitness value, the strings are selected for the new generations which have comparatively super power than other strings. From the population, the points are eliminated in which a moderate fitness value is present. For exploration, especially mutation and crossover are utilized to obtain new solutions [19]. Mutation is major contribute to change a part genetic randomly. Crossover is used to incorporate the fittest members of genetic material from populations [20].

GA provides a different results of feature subset as per the changes in population size. In the literature surveys, various authors use 50 and 70 as the population size. However, the vast literature survey has limitations as proper selection of population size is absent. Therefore, an improved genetic optimized feature selection paradigm is proposed for batch input as well as for sequential input in the context of this paper. Here, the feature subset is finalised by considering the various

population size from 10:10:90. Thus, the total number of feature subsets are evaluated are 9. One feature subset is selected for further experimentation as an ideal feature subset which provides maximum accuracy.

2.4 Classification subsystem

By using optimal features OS-ELM is evaluated. Initialization and sequential learning phase are the major two phases of OS-ELM. In the initialization phase, various parameters are executed like number of data required to fill up, the number of hidden nodes, defining of chunk size, etc. Target class is decided based on the initialized data and new arrived chunk data in the sequential learning phase [21, 22].

Extreme Learning Machine (ELM) is one of fastest and computationally efficient learning algorithm as no additional learning is required due to random forage over input to the hidden layer. ELM randomly assigns the weights within the range 0 to 1 between input layers to the hidden layer. And the weights between hidden layer to the output layer are calculated analytically as shown in Figure 2. ELM has various advantages over traditional neural network classifiers like back propagation algorithm (BP), radial basis function (RBF), support vector machine (SVM), fuzzy neural network (FNN) in terms of speed, reliability and generalization. To clarify, consider the input data set with N instances and the number of neurons present in the input layer, hidden layer and output layer is n, m, k respectively. The β_i is the bias parameter for i^{th} hidden layer neuron. An activation function $g(\Phi)$ is used to connect input and the output layers by using weight vectors $w_i = (w_{i,1}, w_{i,2}, \dots, w_{i,n})^Y$ and $\beta_i = (\beta_{i,1}, \beta_{i,2}, \dots, \beta_{i,n})^Y$. The output vector Y_j can be calculated as:

$$Y_j = \sum_{i=1}^m \beta_i G_i(x_j)$$

For $j = 1, 2, \dots, N$

$$Y_j = \sum_{i=1}^m \beta_i G(w_i \cdot x_j + b_j)$$

The same equations can be rewritten briefly as:

$$H\beta = Y$$

where,

$$H = \begin{bmatrix} G(w_1 x_1 + b_1) & \dots & G(w_1 x_1 + b_n) \\ \vdots & \dots & \vdots \\ G(w_1 x_N + b_1) & \dots & G(w_1 x_N + b_n) \end{bmatrix}_{N \times n}$$

$$\beta = \begin{bmatrix} \beta_1^Y \\ \vdots \\ \beta_N^Y \end{bmatrix}_{n \times k}$$

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times k}$$

The output matrix of the hidden layer is H with respect to inputs x_1, x_2, \dots, x_N and Each output weight β is represented as:

$$\beta = H^+ Y$$

$$H^+ Y = (H^Y H)^T H^Y Y$$

Although, in various existing applications, it is difficult to make available data previously specifically for the sequential input. The conventional batch ELM modifies to OS-ELM to resolve the problems.

3. RESULTS AND DISCUSSION

Related experimentation has been conducted in MATLAB© R2014a. The two activation functions, sigmoidal and radial basis activation function (rbf), are used for simulation. For evaluation, in the literature survey, various authors consider only the number of hidden nodes in OS-ELM [5, 11, 12], but the initial block size is not considered for evaluation. By virtue of experiment, it is observed that the initial training data is much important. Thus, every step of hidden layer, training data (n) is also changing with the number of hidden node (j) like j to n with incremental I value. The output results are calculated by using fixed chunk size (1 or 20) or randomly changing the chunk size between 10 to 30.

For evaluation, the accuracy measure is a utilised as an evaluation measure. The performance metrics are computed by evaluating the values of false negatives and positives both as well as true negatives and positives also [20]. Eq. (1) shows the formula of the calculation of accuracy,

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

The experimental results of ELM and OS-ELM by using all features for binary classification problem are given in Table 1. The accuracy is estimated by batch as well as sequential mode (1-by-1, 20-by-20, 10-30) with various initial block sizes and hidden layer neuron size. The performance evaluation is compared in different ways as 1. IG-ELM and ELM 2. IG-OSELM and OS-ELM.

3.1 IG-ELM and ELM

To enhance the efficiency and effectiveness of IG-ELM paradigm, the clinical datasets are used for experimentation by using ELM classifier. Table 2 indicates GA results changing with the size of population. Total 9 feature subsets are evaluated by varying the population size. Out of all these subsets, one subset is finalized which has the maximum occurrence as an optimal subset as shown in the second last column in Table 2. For example, for PID dataset, the accuracy is calculated by differentiating the value of population. For each subset, the accuracy is shown. From all these subsets, one subset needs to finalize which produced the maximum accuracy. For PID dataset, the optimal feature subset is {2, 5, 6} with accuracy 77.82%. ELM classifier is the source of computation of the classification accuracy. Comparative result analysis of ELM and IG-ELM is as shown in Figure 3. With the result analysis, it is observed that IG-ELM has success to achieve 10.94% improved classification accuracy over 56.75% reduction in features as in ELM.

Table 1. Experimental results of ELM and OSELM on binary classification application

Dataset	Act. Fun.	Algorithm	Learning mode	Training accuracy	Testing accuracy	Node	Initial block size	
PID	Sig	ELM	Batch	87.91	81.30	50	-	
			1-by-1	88.29	82.60	115	465	
		OSELM	20-by-20	87.36	82.17	25	175	
				10,30	83.82	83.04	15	65
	Rbf	ELM	Batch	88.66	81.73	157	-	
			1-by-1	87.17	82.17	80	180	
OSELM		20-by-20	87.91	82.17	80	180		
			10,30	81.59	83.47	45	245	
SHD	Sig	ELM	Batch	99.47	88.88	17	-	
			1-by-1	99.47	88.88	25	125	
		OSELM	20-by-20	99.47	88.88	20	70	
				10,30	99.47	91.35	30	30
	Rbf	ELM	Batch	99.47	86.41	89	-	
			1-by-1	96.29	87.65	50	100	
OSELM		20-by-20	98.41	88.88	30	30		
			10,30	98.94	90.12	20	20	

Table 2. IG-ELM for binary classification problem for clinical dataset

Dataset	Population size											Optimal subset	ELM Acc	
	All	10	20	30	40	50	60	70	80	90				
PID	69.56	2,4,8	2,5,6	2,5,8	2,5,6	2,5,6	2,5,6	2,5,6	2,5,6	2,5,6	2,5,6	2,5,6	2,5,6	77.82
SHD	77.77	3,8,9, 10,13	2,3,10, 12,13	1,2,3, 7,12,13	1,2,3, 9,12	3, 8,9, 10,13	1,2,3, 9,12	1,2,3, 9,12	1,2,3, 9,12	1,2,3, 9,12	1,2,3, 9,12	1,2,3, 9,12	1,2,3, 9,12	83.95
BC	85.16	2,7,8,9	2,7,8,9	2,7,8,9	2,7,8,9	2,3,4, 5,6,7,10	2,3,4, 5,6,7,10	2,3,4, 5,6,7,10	2,3,4, 5,6,7,10	2,3,4, 5,6,7,10	2,3,4, 5,6,7,10	2,3,4, 5,6,7,10	2,3,4, 5,6,7,10	99.52
AS	73.91	3,4,8, 9, 11	3,5,8, 9	5,7,8, 10,11	3,5,8,9	3,5,8,9	3,5,8,9	3,5,8,9	3,5,8,9	3,5,8,9	3,5,8,9	3,5,8,9	3,5,8,9	88.88

Table 3. Experimental results of IG-ELM and IG-OSELM

Dataset	Act. Fun.	Algorithm	Learning mode	Accuracy		Node	Initial block size	
				Training	Testing			
PID	Sig	IG -ELM	Batch	87.36	81.73	43	-	
			1-by-1	83.27	81.73	20	20	
		IG -OSELM	20-by-20	82.71	81.73	55	205	
				[10,30]	81.41	82.6	55	55
	Rbf	IG -ELM	Batch	82.89	81.73	12	-	
			1-by-1	83.64	81.3	40	290	
IG -OSELM		20-by-20	83.64	81.3	45	395		
			[10,30]	81.41	82.6	55	305	
SHD	Sig	IG -ELM	Batch	99.47	83.95	91	-	
			1-by-1	87.83	83.95	10	10	
		IG -OSELM	20-by-20	89.41	83.95	10	40	
				[10,30]	89.41	86.41	10	14
	Rbf	IG -ELM	Batch	99.47	85.18	36	-	
			1-by-1	87.3	85.18	20	30	
IG -OSELM		20-by-20	87.3	85.18	15	33		
			[10,30]	88.88	86.41	15	29	

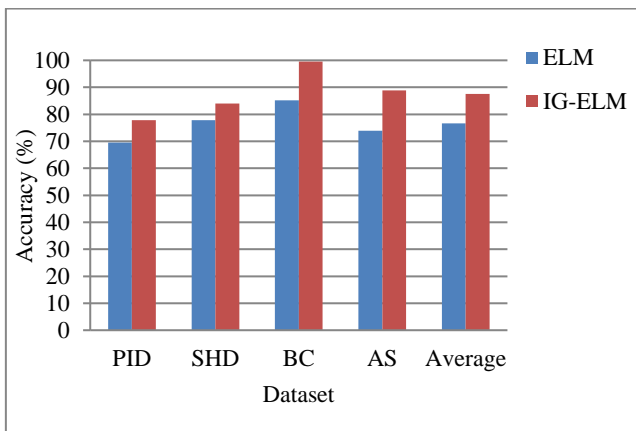


Figure 3. Comparative analysis of ELM and IG-ELM

3.2 IG-OSELM and OSELM

Table 3 presents the experimental results obtained by using the optimal features for ELM and OS-ELM. The comparative performance between OS-ELM and IG-OSELM is as shown in Table 3, obtained by estimating the average of sequential modes (1-by-1, 20-by-20 and (10,30)). It indicates the detailed comparative analysis of IG-ELM and IG-OSELM by using both activity functions like sig and rbf.

4. CONCLUSION

Genetic algorithm is a top priority-based optimization algorithm which to classify the best of optimal feature subset.

However, GA varies its results as per the changes in the population. To solve this problem, in this paper, an improved genetic optimized feature selection paradigm is proposed for sequential input (IG-OSELM) by using clinical datasets. The proposed paradigm is accomplished to handle the dimensionality reduction and optimization problems for sequential input. OS-ELM algorithm is used for sequential input and it trains only new arrival data instead of the whole training datasets, which saves the computational cost. To prove the importance and strength of the IG-OSELM, a comparative study of results for ELM, OS-ELM, IG-ELM are carried out. Here, the IG-OSELM paradigm is evaluated for the binary classification problem. The work can be extended by using the archetype for multiclass classification problem and an improved shuffled frog leaping algorithm [23], which may support the inclusive of clear insights and directions regarding future improvements.

REFERENCES

- [1] Cambria, E., Huang, G.B., Kasun, L.L.C., Zhou, H., Vong, C.M., Lin, J., Yin, J., Cai, Z., Liu, Q., Li, K. (2013). Extreme learning machines [trends & controversies]. *IEEE Intelligent Systems*, 28(6): 30-59. <https://doi.org/10.1109/MIS.2013.140>
- [2] Yu, L. Liu, H., Guyon, I., (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5: 1205-1224.
- [3] Pahuja, G., Nagabhushan, T. (2016). A novel GA-ELM approach for Parkinson's disease detection using brain structural T1-weighted MRI data. 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), 2016, pp. 1-6. <https://doi.org/10.1109/CCIP.2016.7802848>
- [4] Mao, L., Zhang, L., Liu, X., Yang, H. (2014). Improved extreme learning machine and its application in image quality assessment. *Mathematical Problems in Engineering*, Article ID 426152. <https://doi.org/10.1155/2014/426152>
- [5] Liang, N.Y., Huang, G., Saratchandran, B.P., Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*, 17(6): 1411-1423. <https://doi.org/10.1109/TNN.2006.880583>
- [6] Zhu, Y., Qin, A.K., Suganthan, P.N., Huang, G.B. (2005). Evolutionary extreme learning machine. *Pattern Recognition*, 38(10): 1759-1763. <https://doi.org/10.1016/j.patcog.2005.03.028>
- [7] Han, F., Yao, H.F., Ling, Q.H. (2013). An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing*, 116: 87-93. <https://doi.org/10.1016/j.neucom.2011.12.062>
- [8] Chen, Z.X., Zhu, H.Y., Wang, Y.G. (2013). A modified extreme learning machine with sigmoidal activation functions. *Neural Computing and Applications*, 22(3-4): 541-550. <https://doi.org/10.1007/s00521-012-0860-2>
- [9] Nguyen, D., Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. 1990 IJCNN International Joint Conference on Neural Networks, pp. 21-26. <https://doi.org/10.1109/IJCNN.1990.137819>
- [10] Huang, G.B., Chen, L. (2008). Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16): 3460-3468. <https://doi.org/10.1016/j.neucom.2007.10.008>
- [11] Wong, S.Y., Yap, K.S., Yap, H.J., Tan, S.C. (2015). A truly online learning algorithm using hybrid fuzzy artmap and online extreme learning machine for pattern classification. *Neural Processing Letters*, 42(3): 585-602. <https://doi.org/10.1007/s11063-014-9374-5>
- [12] Lan, Y., Soh, Y.C., Huang, G.B. (2009). Ensemble of online sequential extreme learning machine. *Neurocomputing*, 72(13): 3391-3395. <https://doi.org/10.1016/j.neucom.2009.02.013>
- [13] Seera, M., Lim, C.P. (2014). A hybrid intelligent system for medical data classification extreme learning machine: theory and applications. *Expert System Application*, 41(05): 2239-2249. <https://doi.org/10.1016/j.eswa.2013.09.022>
- [14] Zhang, R., Lan, Y., Huang, G.B., Xu, Z.B. (2012). Universal approximation of extreme learning machine with adaptive growth of hidden nodes. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2): 365-371. <https://doi.org/10.1109/TNNLS.2011.2178124>
- [15] Lichman, M. (2013). UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>, accessed on 11 March 2022.
- [16] Li, J. (2004). Bio-medical data set repository. School of Computer Engineering, Nanyang Technological University, Singapore. <http://datam.i2r.a{star.edu.sg/datasets/krbd/index.html>, accessed on 11 March 2022.
- [17] Akusok, A., Björk, K.M., Miche, Y., Lendasse, A. (2015). High-performance extreme learning machines: A complete toolbox for big data applications. *IEEE Access*, 3: 1011-1025. <https://doi.org/10.1109/ACCESS.2015.2450498>
- [18] Zheng, J., Li, X., Wang, F., Ai, B., Qian, J. (2008). A genetic algorithm-based wrapper feature selection method for classification of hyperspectral images using support vector machine. *Proc. SPIE 7147, Geoinformatics 2008 and Joint Conference on GIS and Built Environment: Classification of Remote Sensing Images*, 71471J (7 November 2008). <https://doi.org/10.1117/12.813256>
- [19] Fei, Y., Min, H. (2016). Simultaneous feature with support vector selection and parameters optimization using ga-based SVM solve the binary classification. *IEEE International Conference on Computer Communication and the Internet*, pp. 426-433. <https://doi.org/10.1109/CCI.2016.7778958>
- [20] Nahato, K.B. Nehemiah, K.H. Kannan, A. (2016). Hybrid approach using fuzzy sets and extreme learning machine for classifying clinical datasets. *Informatics in Medicine Unlocked*, 2: 1-11. <https://doi.org/10.1016/j.imu.2016.01.001>
- [21] Huang, G.B., Sundararajan, N., Sundararajan, N. (2004). An efficient sequential learning algorithm for growing and pruning rbf (gap-rbf) networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(6): 2284-2292. <https://doi.org/10.1109/TSMCB.2004.834428>
- [22] Huang, G.B., Sundararajan, N., Sundararajan, N. (2005). A generalized growing and pruning rbf (GGAP-RBF) neural network for function approximation. *IEEE Transactions on Neural Networks*, 16(1): 57-67. <https://doi.org/10.1109/TNN.2004.836241>

[23] Hu, B., Dai, Y., Su, Y., Moore, P., Zhang, X., Mao, C., Chen, J., Xu, L. (2016). Feature selection for optimized high-dimensional biomedical data using the improved shuffled frog leaping algorithm. IEEE/ACM

Transactions on Computational Biology and Bioinformatics, 15(6): 1765-1773.
<https://doi.org/10.1109/TCBB.2016.2602263>