International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Classification of Medical Data Using a Cuckoo Search-Based Hybrid Neural Network

[1]Abdul Razzaque, [2]Dr. K. P. Yadav, [3]Dr. Vishwaprakash Roy

Submitted: 17/05/2024 **Revised:** 30/06/2024 **Accepted:** 12/07/2024

Abstract: In this section, we classify four datasets: Dermatology, a small medical dataset, three benchmark datasets, and a training dataset. This research made use of a number of different types of hybrid neuro fuzzy networks, including a cuckoo search based functional link neural fuzzy network (CSFLNFN) and a cuckoo search based multilayered perceptron (CSMLP). Naive Bayes and K-Nearest Neighbor classifiers are used as benchmarks to evaluate these classifiers. We use principal component analysis (PCA) as a feature extraction method to reduce the dimensionality of these datasets, and we evaluate the differences between the two sets of results. In this research, we use three standard datasets and a small medical dataset called Dermatology.

Keywords: *Hybrid Neural Network, Medical Data, Classification, Development, Techniques.*

1. Introduction

The capabilities of both a neural network and a fuzzy set of rules can be leveraged in a hybrid neuro fuzzy network. As a result, it achieves better results than either a neural network or a fuzzy network would on their own. More accurate than standard artificial neural networks, the CSFLNFN and CSMLP models developed in this paper use functional link artificial neural networks, fuzzy logic, and cuckoo search to solve classification problems. Increasing prediction and classification accuracy is a common goal of many applications, and the recently developed meta-heuristic, evolutionary algorithm cuckoo search is often used to achieve this goal. The particle swarm optimization (PSO) method is superior to differential evolution (DE) and the artificial bee colony method. Civicioglu proves that the CSA provides a more robust and reliable answer (ABC). By combining various characteristics, feature extraction techniques can create novel characteristics. For better classification results and less strain on the system's resources, this is a great addition. Within the academic community, principal component analysis (PCA) is the most popular method for extracting features. Feature extraction from the datasets was accomplished in this research by means of principal component analysis. Each of the five classifiers is tested on both the original normalized datasets and the datasets with the extracted features (newly constructed datasets). Overall accuracy, sensitivity, specificity, confusion matrix, Gmean, F-score, and area under the receiver operating characteristic (ROC) curve are all used to assess the classifiers' efficiency.

- **NeuroFuzzy System with Functional Links Based on Cuckoo Search (CSFLNFN)**

Combining the Functional Link Neural Fuzzy Network model and the cuckoo search evolutionary learning approach, the proposed CSFLNFN model is presented. Cuckoo search learning is used to fine-tune the model parameters of a Functional Link Neural Fuzzy Network.

- **An MLP Architecture Based on Cuckoo Search (CSMLP)**

The MLP uses multiple layers of fully connected neurons that share information via weighted connections. Between the input and output layers in an MLP, there can be any number of hidden layers. However, this research revealed the existence of two hidden levels. Research into the efficacy of MLP for classifying medical datasets is currently underway. Each model's parameters are updated using a

1 Research Scholar, CSE Department, MSEIT, MATS University, Raipur (C.G) India. ab.razzaque@rediffmail.com.
2 VC and Professor, MATS School of Engineering and Information Technology, Raipur. vc@matsuniversity.ac.in
3 Professor, MATS School of Sciences, Raipur. vproy@matsuniversity.ac.in
Corresponding author: ab.razzaque@rediffmail.com

combination of back propagation, cuckoo search (CS), and improved cuckoo search (ICS) learning strategies. Before using MLP classification, the training and testing datasets must be prepared. The input layer receives the vector inputs x_1, x_2, ., x_n. The expected outcome is provided by the instructor. So, let's say we're using an MLP with two hidden layers, like the one in Fig. 1.1, the sum of all data received; x_j^{l+1}.

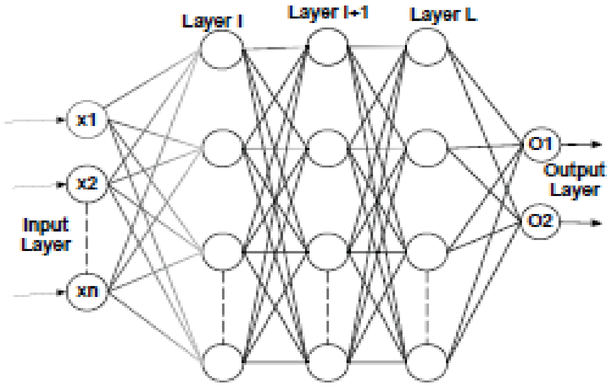


Figure 1.1: The Perceptron Architecture with Many Layers

According to neuron j in layer 1+-1 as,

$$x_j^{l+1} = \sum_i y_i^l w_{ji}^l$$

In this equation, w^l_{ji} is the strength of the connection between the ith neuron in layer | and the j^{th} neuron in layer 1+1, where yi^l is the ith neuron from the l^{th} layer above. It is shown that the neuron's output is a nonlinear sigmoid activation function of the neuron's total input, as

$$y_i^l = \frac{1}{1+e^{-x_j^l}}$$

Data from every input layer node,

$$y_j^0 = x_j^0$$

Where the j^{th} part of the input vector at the input layer is denoted by$x\ _j^0$. It's a complete set. All the hidden unit's internal weights will be calculated using a back propagation learning algorithm. For any pair of output vectors and any network weight vector w, the Least Mean Square (LMS) error is calculated as

$$E(w) = \frac{1}{2}\sum_{js}\left(y_{js}^L(w) - d_{jx}\right)^2$$

Where d_{js} is the desired output and y^L_{js},(w) is the actual output of node j in the L^{th} layer for the s^{th}input-output case. To achieve this goal, we employ the gradient-descent method and perform a number of weight updates according to the formula

$$\varDelta w_{ji}^l(t) = -\in + \alpha \varDelta w_{ji}^l(t-1)$$

Where ∊ represents the momentum damping coefficient, which ranges from $0 \leq a \leq 1$. Information about the MLP can be found in. This model's parameters, or weights, are, however, optimized using a back propagation learning algorithm. To solve the local minima problem of the backpropagation (BP) technique and boost classification accuracy, we combine MLP with the cuckoo search (CS) method and evaluate the results against the proposed model.

2. The Type Of Learning Algorithm That Was Implemented

This section focuses on the Back propagation algorithm and the Cuckoo search method, both of which are employed as learning algorithms to maximize the parameters of all of the models for all of the datasets.

- **A Learning Algorithm Based on Back Propagation**

A supervised learning method, the back propagation algorithm. Changing the link weights and settings to generate Gaussian membership functions is then utilized to reduce the objective function. Parameters are optimized by deriving the negative gradient of the cost function with respect to a specific weight parameter. The objective function is defined as follows:

$$E(t)=\frac{1}{2}\left[\hat{y}(t)-y(t)\right]^2$$

where y(t) is the desired result and y(t) is the calculated one. Given that the FLANN outputs are fed into the FLNFN model's consequent component, the algorithm's task is to maximize the weights utilized in both the antecedent and consequent components of the model. Earlier, we saw how the parameters (center-c and width-) are used to construct the FLNFN membership functions. It is necessary to utilize the gradient descent algorithm to optimize the parameters because they are initially chosen at random. You can modify the settings in the previous section using the following formula (c and).

$$c(t+1)=c(t)-s\eta\frac{\partial E}{\partial c}$$

$$\sigma(t+1)=\sigma(t)-s\eta\frac{\partial E}{\partial \sigma}$$

In which s is the membership function learning coefficient. For FLANN, there are actually two distinct weighting schemes (data and bias). We employ the gradient descent learning approach to maximize both the data weights and the bias. All parameter modifications in terms of weighting are as follows:

$$w(t+1)=w(t)+\Delta w(t)=w(t)+\left[-w\eta\frac{\partial E(t)}{\partial w(t)}\right]$$

Where $w\eta$ is the learning rate used in FLANN. $\frac{\partial E(t)}{\partial w(t)}$ depicts a weight's partial derivative in terms of the error it produces. If the rate of learning is set too low, the network will learn too slowly, making it one of the most important decisions to make while training a neural network. If it's too difficult, no progress may be made. Sometimes the datasets used have a role as well. The BP method is more likely to get trapped in the local solution and fail to achieve the global solution if the initial weights are on a local grade, as BP only converges locally. The model parameters were optimised using CSA to get around this restriction.

- **A Cuckoo Search Algorithm**

In recent years, metaheuristic algorithms have proven to be more effective than traditional approaches. generations of experience solving difficult engineering optimization challenges. Nature serves as an inspiration for most of them, who employ a strategy based on rules and chance. Many other optimization strategies, such as the ant colony optimization algorithm (ACO), particle swarm optimization (PSO), and tabu search, have found widespread use in recent years. In fact, they are still being employed regularly in cutting-edge research. The best results achieved by PSO, GA, DE, and other algorithms have been demonstrated to be significantly inferior to those achieved by the cuckoo search algorithm (CS), which was proposed by Yang and Deb in 2010. Yang and Deb developed the concept of Lévy flying to enhance the CS algorithm. Foraging behaviour in animals and Lévy flight both have their origins in the concept of a "random walk." The outcomes of earlier actions and the probabilities of proceeding to subsequent phases are constantly taken into account while planning the next step.

As a rule, the CS's settings remain unchanged. The CS algorithm was used to develop the CSELM classification algorithm. After analyzing the parameters of Yang and Deb's original CS algorithm, the paper introduces the revised ICSELM classifier method. Inspiring the CS algorithm was the behaviour of parasitic cuckoo species, which select a nest in which the host bird has just laid its own eggs. In most cases, cuckoo eggs will hatch before their host eggs. The cuckoo chick's first act after hatching is to push the host eggs out of the nest in a blind attempt to enhance its share of the food provided by the host bird. The cuckoo chick will mimic the host chick's call-in order to drive the host bird's eggs from the nest and gain access to more food. The three main strategies upon which Yang and Deb base the CS are as follows:

The best nests with the best eggs (solutions) are passed down through the generations like cuckoos laying one egg at a time in a nest picked at random.

An individual host has a probability of pa [0,1] of discovering an alien egg, where n is the number of probable nests. When this happens, the host bird has two options: either abandon the nest and the egg, or take the egg and move on to a new spot. Finally, a fraction pa (with new random solutions at new places) approximates the method of replacing the n nests with new nests. A Lévy flight is carried out to generate the new solutions in the new locations) for, let's say, cuckoo $x_i^{(t+1)}$.

$$x_i^{(t+1)} = x_i^t + \alpha \oplus le'vy(\lambda)$$

Where the step size a > o is connected to the relevant scales of the problem. Entry-wise multiplication is denoted by the symbol. As part of this research, we take into account a Lévy flight in which the step lengths are distributed according to a formula whose probability distribution has an infinite variance.

$$le'vyL = t^{-\lambda}, 1 < \lambda \le 3$$

Algorithm: A Cuckoo Search Algorithm

Begin

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

Generate an Initial Population of 'n' number of hosts nests or different solutions x_i*(I = 1, 2,….n)*

While *(t<Max_Iteration)*

do

Get a Cuckoo (say i) randomly by Levy flights

Evaluate it's fitness F_i

Choose a next 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. Integrating PCA-Based Feature Extraction

Karl Pearson came up with principal component analysis in 1901. In statistics, principal component analysis (PCA) is used to reduce the number of dimensions in a feature space. Particulars that are worth noting. To do this, PCA tightly rotates the axes of the p-dimensional space to new positions (principal axes), with the largest variance on principal axis 1, the next highest variance on principal axis 2, and so on. Axes are uncorrelated if and only if the covariance between any two sets of them equals 0. PCA has been incorporated for feature extraction.

$$G \leftarrow [a_1 a_2, \ldots\ldots\ldots\ldots a_d] \text{ where } d \ll M$$

if x is a test point

$$x \in R^M \rightarrow xG\,\varepsilon\, R^d \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

Algorithm: PCA Algorithm

Begin

Input Feature Matrix X

Normalize the matrix 'X' to ensure zero mean of each feature value

Let training set = [x^1, x^2, ……x^m]

for *(j=1 to m)*

Evaluate $\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_i^j$

__for__ (i – 1 to n)

Replace x^j with $(x^j - \mu_j)$

end for

end for

compute the covariance matrix of the normalized matrix

compute the eigen vectors of matrix

choose the first 'k' number of principal components as follows:

for** (every eigen vector I = 1 to m) **do

Evaluate cumvar = $\frac{\sum_{i=1}^{k} \lambda_{ii}}{\sum_{i=1}^{m} \lambda_{ii}}$ *{cumvar represents cumulative variance}*

and λ represents eigen values sorted in descending order}

if (cumvar$\geq$0.99) or (1-cumvar $\leq$ 0.01) then

return k

end if

end for

$eign_{reduced}$ = eign (: 1: k)

Evaluate Z = X$eign_{reduced}$*

end

4. Methods And Information Sources For Assessing Performance

Following is a brief description of the four reference datasets used for classification, all of which were obtained from the UCI repository. There are four types of data to choose from: Dermatology, Wine, Glass, and Iris.

- **Dermatology:** The kind of Eryhemato-Squamous Disease is determined with the help of this data collection. There are 366 unique instances, 34 characteristics, and 6 distinct classes.
- **Wine:**An Italian-wide three-class dataset. Many pattern recognition projects rely on the wine dataset. Alcohol, malic acid, ash, ash alkalinity, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, colour intensity, hue, diluted wines' OD280/OD315, and proline are some of the properties.
- **Glass:** When it comes to solving crimes, B. This information was compiled by the German, Central Research Establishment, Home Office Forensic Science Service, Reading, Aldermaston, Berkshire RG7 4PN. Identity, colour, and RI are the characteristics. The letters Na, Mg, Al, Si, Ca, Ba, and Fe stand for sodium, magnesium, aluminium, silicon, calcium, barium, and iron, respectively.
- **Iris:** This is the most widely-used dataset for pattern recognition studies. The characteristics include the size of the sepals and petals, the length of the sepals and petals, and so on. For each dataset, the table provides information on the total number of training samples, testing samples, features, and classes.

Table 4.1: The Accuracy of Each Model Put to the BP and CS Tests

Models	Dermatology	Wine	Glass	Iris
BPFLNFN	.682	.696	.632	.943
CSFLNFN	.923	.935	.828	.978
BPFLANN	.637	.628	.587	.908
CSFLANN	.885	.908	.817	.964
BPMLP	.642	.654	.619	.887
CSMLP	.849	.881	.768	.981

5. Theory And Experimental Findings

In this research, we utilize five different models (CSFLNFN, MLP, FLANN, Naive Bayesian, and k-nearest neighbour) on four multi-class datasets (dermis, wine, glass, and iris). One can evaluate the efficacy of the models using a number of different metrics, such as the confusion matrix, accuracy, sensitivity, specificity, F-score, g-mean, and area under the receiver operating characteristic (ROC) curve. PCA was used to extract characteristics from the data sets. Each of the five classifiers is given access to the original normalized datasets and features extraction procedures before their performance is compared.

- Outcomes Before Attempting to Extract Features
- The Outcome of Feature Extraction

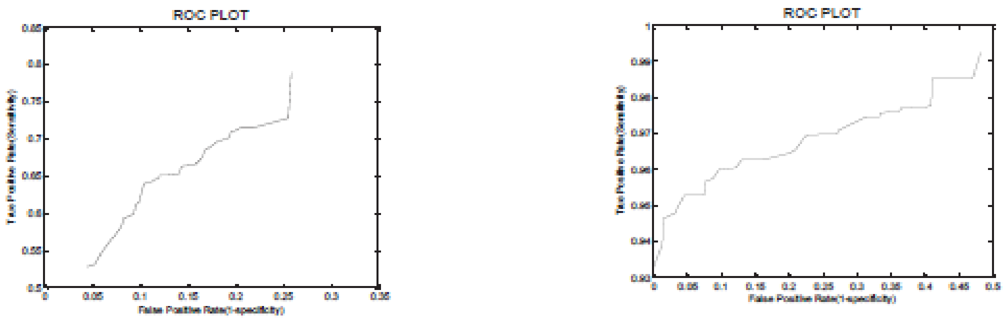
Table 5.1: Test. Overall Model Accuracy, Sensitivity, Specificity, Mean, and F-Score (Before Feature Extraction)

Dataset	Model	Test. Accu.	Sen.	Spe.	Gmean	F-Score
Dermatology	CSFLNFN	.923	.928	.505	.684	.654
	CSFLANN	.885	.875	.666	.763	.756
	CSMLP	.849	.833	.623	.720	.712
	Naïve Bayesian	.826	.846	.598	.711	.700
	KNN	.892	.891	.548	.698	.678
Wine	CSFLNFN	.935	.810	.578	.684	.674
	CSFLANN	.908	.798	.777	.787	.787
	CSMLP	.881	.876	.821	.847	.847
	Naïve Bayesian	.842	.918	.632	.746	.746
	KNN	.898	.898	.748	.816	.816
Glass	CSFLNFN	.828	.789	.500	.628	.612
	CSFLANN	.817	.888	.550	.699	.679
	CSMLP	.768	.727	.733	.730	.730
	Naïve Bayesian	.769	.772	.538	.645	.634
	KNN	.788	.750	.700	.7246	.724
Iris	CSFLNFN	.978	.976	.875	.933	.931
	CSFLANN	.964	.973	.855	.912	.910
	CSMLP	.981	.918	.822	.868	.867
	Naïve Bayesian	.900	.989	.935	.957	.957
	KNN	.967	.991	.982	.986	.986

Table 5.2: Test. Model Comparisons on Accuracy, Sensitivity, and Specificity (Gmean, Fscore) (After Feature Extraction)

Data-set	Model	Test. Accu.	Sen.	Spe.	Gmean	FScore
Dermatology	CSFLNFN	.959	.939	.529	.704	.677
	CSFLANN	.903	.952	.589	.748	.728
	CSMLP	.852	.946	.685	.805	.795
	Naïve Bayesian	.834	.976	.600	.765	.743

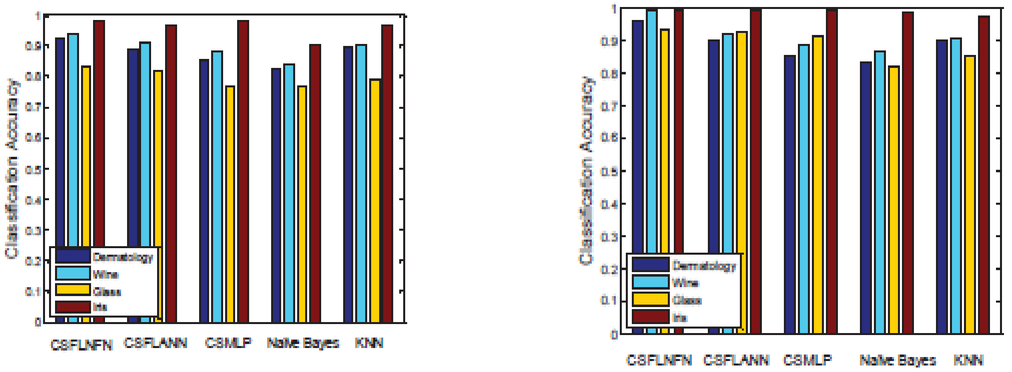
	KNN	.898	.965	.517	.707	.674
Wine	CSFLNFN	.995	.826	.652	.734	.728
	CSFLANN	.917	.904	.851	.877	.877
	CSMLP	.885	.928	.884	.906	.906
	Naïve Bayesian	.866	.965	.687	.814	.803
	KNN	.906	.933	.750	.836	.831
Glass	CSFLNFN	.9307	.833	.666	.745	.740
	CSFLANN	.9281	.800	.590	.687	.679
	CSMLP	.9116	.588	.600	.591	.591
	Naïve Bayesian	.8181	.875	.764	.823	.634
	KNN	.8484	.769	.900	.800	.8319
Iris	CSFLNFN	.995	1	1	1	1
	CSFLANN	.991	.954	1	.977	.976
	CSMLP	.991	1	1	1	1
	Naïve Bayesian	.985	1	1	1	1
	KNN	.972	1	1	1	1



(a) ROC For Dermatology Using CSFLNFN Model Before Feature Extraction

b) ROC For Dermatology Using CSFLNFN Model After Feature Extraction

Figure 5.1: ROC for Dermatology



(a) Comparison Of Classification Accuracies (Testing) Before Feature Extraction

(b) Comparison Of Classification Accuracies (Testing) After Feature Extraction

Figure 5.2: Accuracy Assessments of Different Classification Methods

One medical dataset (dermatology) and three additional frequently used datasets are classified using the suggested CSFLNFN model (wine, glass, and iris). The cuckoo search learning technique was used to optimize the parameters of the CSFLNFN, FLANN, and MLP models.

6. Conclusion

This paper proposes a taxonomy for four different multi-class datasets, one of which is a medical dataset called dermatology, and the other three are wine, glass, and iris, all of which are often utilized. Naive Bayes, k-nearest neighbour, CSFLANN, and CSMLP are the other four models covered here. To find the best settings for the CSFLNFN, CSMLP, and CSFLANN models, an evolutionary learning strategy called the cuckoo search was employed. Extraction of the attributes of the datasets is also performed using PCA to enhance precision. All five classifiers are tested by being fed both whole datasets and the extracted features (which are generated anew). The models in this analysis were compared using a number of different performance metrics, such as the confusion matrix, accuracy, sensitivity, specificity, F-score, gmean, and area under the receiver operating characteristic (ROC) curve. Results from simulations indicate that feature extraction using principal component analysis (PCA) increases classification accuracy regardless of the models or datasets employed. As a conclusion, it is clear that the suggested CSFLNFN model beats the other four models (CSFLANN, CSMLP, Naive Bayes, and KNN) independent of the dataset type or feature extraction approach used.

References

[1] A. Bender, H. Y. Mussa, R. C. Glen, and S. Reiling, “Molecular similarity searching using atom environments, information-based feature selection, and a naive bayesian classifier,” *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 170-178, 2004.

[2] B. Cai and X. Jiang, “A novel artificial neural network method for biomedical prediction based on matrix pseudo-inversion,” Journal of biomedical informatics, vol. 48, pp. 114-121, 2014.

[3] Chaudhuri and U. Bhattacharya, “Efficient training and improved performance of multilayer perceptron in pattern classification,” *Neurocomputing*, vol. 34, no. 1, pp. 11-27, 2000.

[4] C.-H. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R. M. Rifkin, M. An-gelo, M. Reich, E. Lander, J. Mesirov, and T. Golub, “Molecular classification of multiple tumor types,” Bioinformatics, vol. 17, no. suppl 1, pp. S316-S322, 2001.

[5] D.-C. Li, C.-W. Liu, and S. C. Hu, “A fuzzy-based data transformation for feature extraction to increase classification performance with small medical data sets,” *Artificial Intelligence in Medicine*, vol. 52, no. 1, pp. 45-52, 2011.

[6] E. Valian, S. Mohanna, and S. Tavakoli, “Improved cuckoo search algorithm for feedforward neural network training,” *International Journal of Artificial Intelligence & Applications*, vol. 2, no. 3, pp. 36-43, 2011.

[7] G.-D. Wu and Z.-W. Zhu, “An enhanced discriminability recurrent fuzzy neural network for temporal classification problems,” *Fuzzy Sets and Systems*, vol. 237, pp. 47-62, 2014.

[8] H. A. Abbass, “An evolutionary artificial neural networks approach for breast cancer diagnosis,” *Artificial Intelligence in Medicine*, vol. 25, no. 3, pp. 265-281, 2002.

[9] H. Chen and C. Yang, “The improvement of breast cancer prognosis accuracy from integrated gene expression and clinical data,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 4785-4795, 2012.

[10] J. Kuruvilla and K. Gunavathi, “Lung cancer classification using neural networks for ct images,” *Computer methods and programs in biomedicine*, vol. 113, no. 1, pp. 202-209, 2014.

[11] J. Leng, C. Valli, and L. Armstrong, “A wrapper-based feature selection for analysis of large data sets,” 2010.

[12] K. Crammer and Y. Singer, “On the algorithmic implementation of multiclass kernel-based vector machines,” *The Journal of Machine Learning Research*, vol. 2, pp. 265-292, 2002.

[13] Q. He, X. Jin, C. Du, F. Zhuang, and Z. Shi, “Clustering in extreme learning machine feature space,” *Neurocomputing*, vol. 128, pp. 88-95, 2014.

[14] S. K. Saha, S. P. Ghoshal, R. Kar, and D. Mandal, “Cat swarm optimization algorithm for optimal linear phase fir filter design,” *ISA transactions*, vol. 52, no. 6, pp. 781-794, 2013.

[15] Statnikov, I. Tsamardinos, Y. Dosbayev, and C. F. Aliferis, “Gems: a system for automated cancer diagnosis and biomarker discovery from microarray geneexpression data,” *International journal of medical informatics*, vol. 74, no. 7, pp.

491-503, 2005.

[16] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, “Medical data classification using interval type-2 fuzzy logic system and wavelets,” *Applied Soft Computing*, vol. 30, pp. 812-822, 2015.

[17] R. Diaz-Uriarte and S. A. De Andres, “Gene selection and classification of microarray data using random forest,” *BMC bioinformatics*, vol. 7, no. 1, p. 1, 2006.

[18] R. Haritha, Dr. D. Suresh Babu and Dr.P.Sammulal. A Hybrid Approach for Prediction of Type-1 and Type-2 Diabetes using Firefly and Cuckoo Search Algorithms. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 2 (2018) pp. 896-907.

[19] W.-Y. Deng, Q.-H. Zheng, L. Chen, and X.-B. Xu, “Research on extreme learning of neural networks,” *Chinese Journal of Computers*, vol. 33, no. 2, pp. 279-287, 2010.

[20] X.-S. Yang and S. Deb, “Multi objective cuckoo search for design optimization,” *Computers & Operations Research*, vol. 40, no. 6, pp. 1616-1624, 2013.