Robust Classifier Design with Ensemble Neural Network using Differential Evolution

Shobha T[1] and Dr. R J Anandhi[2]

[1]Research Scholar, Department of CSE, The Oxford College of Engineering, Bengaluru
Visvesvaraya Technological University, Belagavi, Karnataka, India

[2]Professor and Head, Department of ISE, New Horizon College of Engineering, Bengaluru
Visvesvaraya Technological University, Belagavi, Karnataka, India

Abstract — *An ensemble neural network using Differential Evolution (DEENN) for the classification has been designed and implemented in this work. The ensemble structure with two levels of classifier has been proposed and designed. Four multilayer perceptron classifiers of the first level have been trained with same sized data with overlapping and learnt through gradient descent algorithm. Second level single-layer neural network fuses the output of first level classifiers and the final output has been derived using differential evolution (DE). To demonstrate the performance of proposed work, the decision integrating methods, majority voting method and mean decision value method has also been implemented, tested and compared. Results show that the proposed method is highly efficient and resistant to trial variation.*

Keywords — *Classification, Differential Evolution, Ensemble Neural Network, Feed forward Architecture*

I. INTRODUCTION

Machine learning refers to a data analysis method that can automatically build an analytical model without explicit programming. Classification is one of the important mechanism available in machine learning. It is a supervised learning method that gains knowledge from the given data and uses this learning to predict the labels of new data. There are various classification algorithms in machine learning such as support vector machine, nearest neighbour, swarm intelligence, neural network, decision tree, evolutionary algorithm etc.

Artificial Neural Networks (ANN) are considered as one of the very powerful classifiers [6]. ANN is a biological neural networks basedcomputational model for information processing. The main drawback of the single classifier is not having generalized knowledge of classification. And also single classifiers fail to provide satisfactory results when dealing with the data with more category and noise [5]. To overcome the problems of single classifiers, one common solution is to use an ensemble of classifiers, which combines the decisions of multiple classifiers to obtain a better and efficient result. Ensemble neural network (ENN) originated from Hansen and Salamon’s work [3], showed that the generalization ability of an NN system can be significantly improved through ensembling several NNs. Hence, ENN has been extensively studied by many researchers [1-12], due to its remarkable improvement in the aspect of the generalization ability. It has already been applied to diversified areas such as protein modification detection[2], syntax analysis of natural language [4], metabolomics studies[1], time series prediction [7], human pose estimation[8], blur image identification [9], protein-protein interactions [10] etc.

Structure formation and identifying the importance of each classifier are the very important factors of ensemble network that effects its generalization capability. The generalization ability of ENN can be enhanced, by improving the generalization ability of individual NN as well as an increase in diversity between individual NN [4]. For aggregating the decisions of individual NN, general approaches used are majority voting [3,4,9], simple averaging[5,10] and weighted averaging[6]. There are many other approaches for aggregating the decisions such as entropy [11], Akaike information criterion [12], type 2 fuzzy system [7], genetic algorithm [13] etc. The proposed method in this paper uses differential evolution (DE) based aggregation of decisions, which is not been found in any other similar work.

In this proposed work, the two-level classifier has been designed using feed-forward NN (FF-NN) at the first level and using differential evolution, an ensemble of NN architecture has been developed. Gradient descent learning method is used to train each of the FF-NN. Some overlapping training data has been used to introduce diversity for each individual classifier. A single layer feed-forward architecture has been used in the second-level classifier, whose weights have been evolved using DE.

To understand the quality involved with the developed algorithm, the algorithm has to be evaluated over the complex and standard dataset. With this respect, the NN benchmark problem XOR is considered because of its nonlinear classification characteristics. To analyze the capability of the proposed work, the proposed differential evolution based learning method was evaluated on benchmark XOR classification problem. Further, to define classification over other data sets, ensemble network with DE has been developed.

II. PROPOSED METHODOLOGY

It is well known that all-natural computing paradigms have some kind of variability in their outcomes with the repetition of the process. The minimum value of variability is always desired characteristics of any classifier. To minimize the variability, the ensemble of different classifiers (which may differ in their structure and/or in their knowledge) is one of the best possible approaches, which has been considered in this work. The issue that occurs in optimal ensemble architecture design is assigning the weightage to individual classifier outcome, which provides the challenge. In this work, the difference in the knowledge of the classifier has been considered by providing the different set of training data set to same sized architecture. To ensemble the classifier, rather than assigning linear weighted or efficiency-based weightage, a single layer feed-forward neural network architecture has been considered which decides the weightage of the individual classifier through the differential evolution based learning process. Such a process considers the individual outcomes of each classifier for final weightage value.

The detailed structure of the development of first stage classifiers is as shown in Figure 1. For each classifier, a partial overlapping training data have been given along with their corresponding targets. The learning of optimal weights to minimize the error has been given through the gradient descent based algorithms. Such a process provides some kind of similarity as well as diversity in their solution development knowledge.

In the second stage, a neural network ensemble has been provided to integrate the classifier as shown in Figure 2. Each classifier outcomes become the input to this neural network ensemble which try to deliver the corresponding target outcome by weight up-gradation through the differential evolution. The use of differential evolution provides the facility to explore the weight domain in an optimal manner. The weight exploration process through differential evolution has been shown in Figure 3, where the first dimension of the solution is decided through the available number of the classifier. Hence a parent solution in DE will have the same size length as the number of classifiers. It is necessary that there should be relative weightage of each classifier outcome, hence the weight of each classifier will have the value in the range of [0,1]. So, in each iteration weight values have been normalized using Equation (1).

$$x_n = \frac{x - x_{mn}}{x_{mx} - x_{mn}} \quad (1)$$

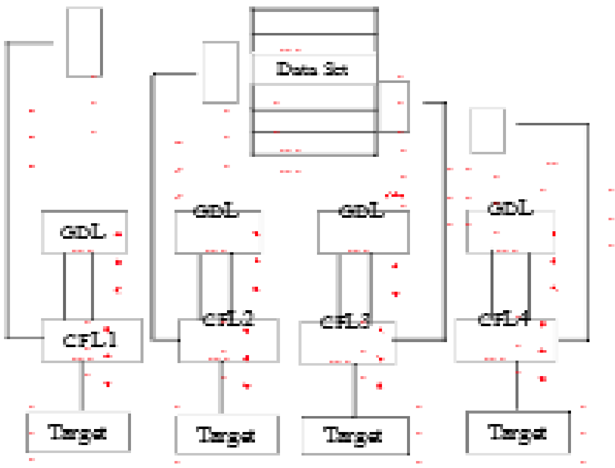

Figure 1: Architecture of training phase of first stage Neural Network

A. Differential Evolution

Differential evolution is a population-based meta-heuristic algorithm. It is a member of evolutionary computation with a high level of exploration that leads to delivering the global solution. As in the case of Genetic algorithm, there are three main operators which help to create the next generation members namely mutation, cross-over and selection. In the DE each parent creates an offspring through the differential change of other members in the population as given by Equation (2). First, a mutation vector is created using Equation (2) and then all-point crossover is applied with parent solution to form the offspring as given by Equation (3). The decision of survival of each offspring for the next generation is based on comparison with their corresponding parent, using Equation (4).

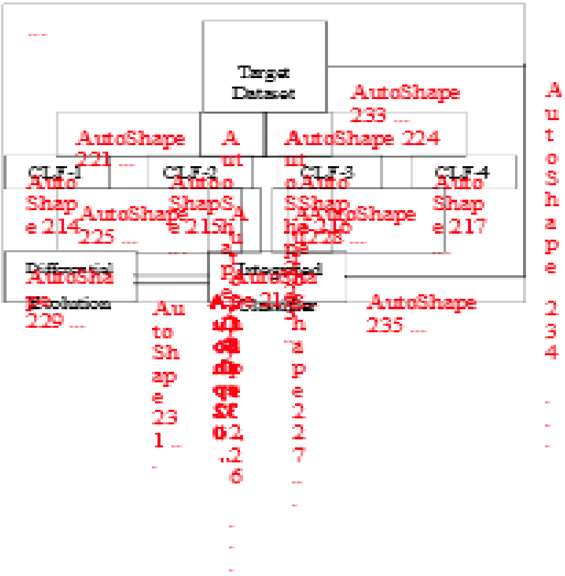

Figure 2. Functional module for Second stage Ensemble Neural Network

$$V_i^{(G)} = X_{r1}^{(G)} + F * \left(X_{r2}^{(G)} - X_{r3}^{(G)}\right) \quad (2)$$

$$u_{ij}^{(G)} = \begin{cases} v_{ij}^{(G)} & if\ rand(0,1) \leq CR\ or\ j = j_{rand} \\ x_{ij}^{(G)} & otherwise \end{cases} \quad (3)$$

$$x_{ij}^{(G)} = \begin{cases} u_i^{(G)} & if\ f\left(u_i^{(G)}\right) \leq f\left(x_i^{(G)}\right) \\ x_i^{(G)} & otherwise \end{cases} \quad (4)$$

A. Gradient Descent Learning Algorithm

The weight up-gradation in the neural network for the 1st stage has been given by the gradient descent algorithm. The gradient over error directs the change in direction in the weight values. The change in the negative direction of the gradient ensures that there is a sharp decline in the error in the local space. The learning rate decides the pace of change over the landscape and it must be in the range of [0,1]. The momentum constant further helps to have convergence faster. The mathematical representations of weight up-gradation have been shown in Equation 5 to Equation 7.

Algorithm:

1. The weights of the network are initialized using the Gaussian distribution random number process.
2. Network response is derived from the set of training data.
3. The desired network responses is compared with actual output of the network for calculating local error, using Equation 5 and 6.

For the output layer:

$$\delta_i^s = \left(d_q - x_{out,i}^s\right) g\left(u_i^s\right) \quad (5)$$

$$\delta_i^s = \left(d_q - x_{out,i}^s\right) g\left(u_i^s\right) \quad (5)$$

For the hidden layer:

$$\delta_i^s = \sum_{h=1}^{n2} \delta_h^{s+1} w_{hi}^{s+1}\, g\left(u_i^s\right) \quad (6)$$

$$\delta_i^s = \sum_{h=1}^{n2} \delta_h^{s+1} w_{hi}^{s+1}\, g\left(u_i^s\right) \quad (6)$$

4. Equation 7 is used to update the network weights.

$$w_{ij}^s(t+1) = w_{ij}^s(t) + \mu \delta_i^s x_{out,j}^s\ \alpha\left[w_{ij}^s(t) - w_{ij}^s(t-1)\right] \quad (7)$$

5. Iteration is stopped if the network has converged.

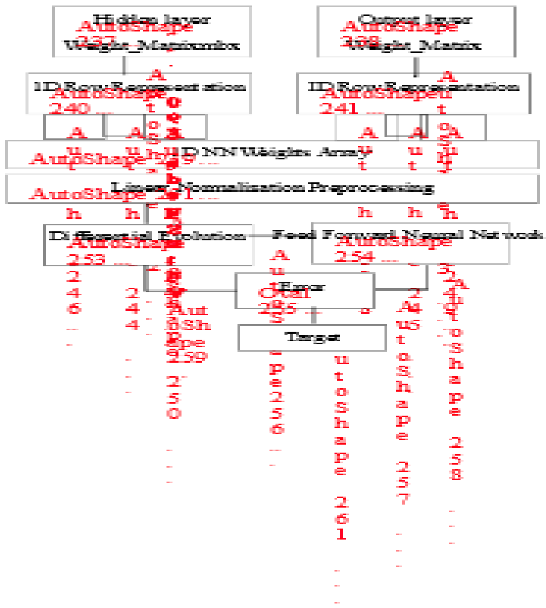

Figure 3. The architecture of evolving learning with Differential Evolution

III. DATASETS USED IN THE EXPERIMENT

A. Exclusive OR(XOR) dataset

A well-known classic problem XOR in Artificial Neural Network research has been considered for performance evaluation of the proposed ENN with Differential Evolution. The important feature of XOR data is that it is not linearly separable. The XOR dataset that is used for training is shown in Table 1. A feedforward NN architecture with size [2 3 1] has been considered.

TABLE 1: XOR DATASET

Input 1	Input 2	Target
0	0	0
0	1	1
1	0	1
1	1	0

B. Heart Disease dataset

We have also used a publicly available benchmark data, "Heart Disease" dataset from the UCI repository [14]. This database contains 303 datasets with 76 attributes, but the only subset of 13 of them from processed Cleveland database has been considered. Dataset has two classes, that refers to whether the heart disease is present or not in the patient. A feedforward NN architecture with size [13 6 1] has been considered.

IV. RESULTS AND DISCUSSION

A. Experimental Results with XOR dataset

The learning of neural weights using DE has been confirmed with the benchmark XOR classification problem. The success over this problem provides the guarantee of nonlinear classification capability of the classifier. The XOR problem has been considered as a benchmark because of its high non-linear separable characteristics. In this work, a feed-forward neural network architecture having a size of [2 3 1] has been considered where each active nodes has the nonlinear activation function as a sigmoid function. The learning iterations has been terminated based on self-terminating criteria when the mean square error becomes less than 0.007. The population size of 100 has been considered for the DE. This is basically a 9-dimensional problem where each weight represents a component. The explored weight for the hidden layer and output layer by the DE has been shown in Table 2 and Table 3. Figure 4 shows the convergence characteristics for the best member in each generation and the population mean. A very close progress in the convergence path can be observed which indicates that complete population converge to the global solution. The obtained final classified outcome for the different classes has been shown in Table 4. From Table 4, it can be observed that the difference between the target and delivered output is very small. Such kind of performance of learning ensures that DE can be very effective optimizer for the neural weights.

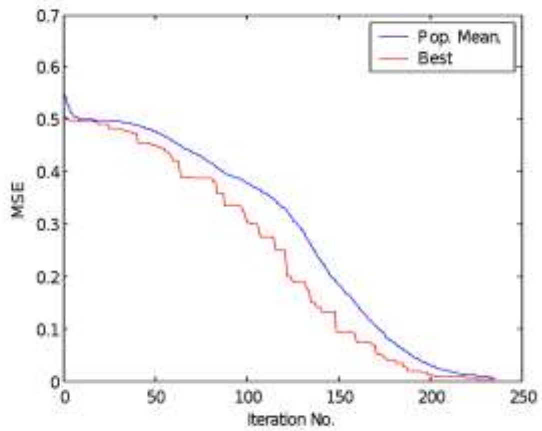

Figure 4: Learning error convergence characteristics for Ensemble Neural Network with Differential Evolution

TABLE 2: HIDDEN LAYER WEIGHTS OBTAINED AFTER COMPLETION OF LEARNING USING DEENN

Input node	Hidden node1	Hidden node2	Hidden node3
1	-4.3461	6.1494	12.0642
2	15.4913	9.5075	-5.7447

TABLE 3: OUTPUT LAYER WEIGHTS OBTAINED AFTER COMPLETION OF LEARNING USING DEENN

Hidden node	Output node
1	-17.5399
2	22.6341
3	-16.2271

TABLE 4: FINAL OBTAINED OUTPUT BY DEENN

Input Data		Target	DEENN Output
0	0	0	0.0038
0	1	1	0.9936
1	0	1	0.9978
1	1	0	0.0000

A. Experimental Results with Heart Disease dataset

To check the practical applicability of the proposed method, it has been applied over the heart disease data available in the UCI repository. Dataset has 13 attributes and 303 data samples. Here we have considered the four different classifiers, each classifier has training data of 100 data samples while remaining data samples have been used for test cases. For the 1st classifier, data samples from 1-100 have been used for training data, while for 2nd classifier it is 51-150, for 3rd classifier, it is 101-200 and for 4th classifier has training data from 151-250. Such distribution ensures the similarity because of overlapping while diversity because of the inclusion of other datasets. The parameter of gradient learning, learning rate and momentum constant have been considered as 0.2 and 0.1. A low value of learning rate ensures the exploration of local space much better. The number of iterations for each classifier was 2000. In Figure 5, the convergence characteristics for all the four classifiers have been shown. The classifier performances over training and test data along with final mean square error have been shown in Table 5. The outcome efficiency for each classifier over the training data is appreciating and is around [98.6%, 98.4%, 81.2% and 79%]. While performances were disappointed over test data and are [77%, 76.6%, 81.2% and 79%] for the individual classifier in sequence from 1st to 4th. Such performances are the cause of worry particularly for critical applications like health care.

The possibilities of ensembles have been defined previously through the voting system or average tendency. In voting system, decision is counted and the majority decision is the final outcome. Such a process has the limitation of complete ignorance of classifier decision which was not part of majority decision. In the averaging decision process, the outcome value of individual classifier has been considered and the final outcome value is estimated by averaging and later a threshold value is applied to decide the final decision. This process has the advantage of using each classifier outcomes and it is also observed that the averaging based performances are superior in comparison to the voting-based decision.

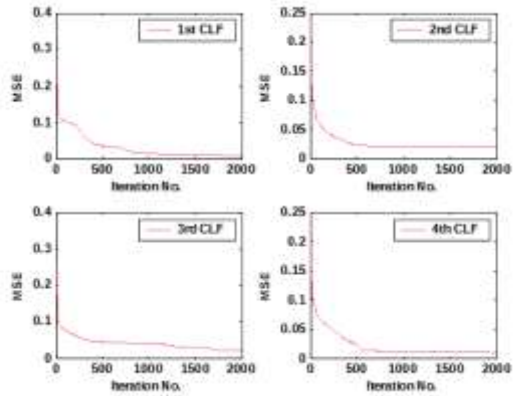

Figure 5. Error Convergence of Individual classifier

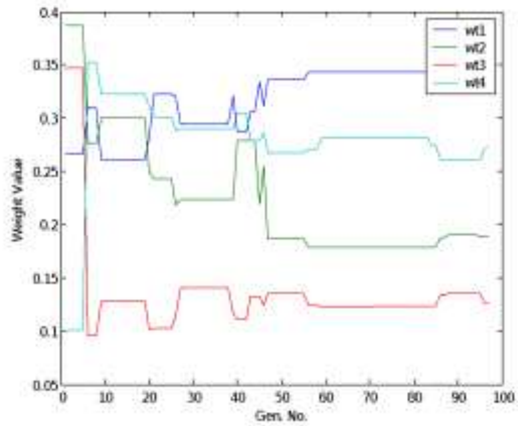

Figure 6. Weight evolution for ensemble by Differential Evolution

But the problem of averaging approach is that each classifier decision is given equal importance, which suppresses the cause of nullification i.e., means a better decision value is suppressed by inferior decision. The proposed DE based ensemble approach has been applied over 5 independent trials and their performance over training and test data along with obtained weightage for individual classifier has been shown in Table 6. The path of weights convergence for individual classifiers have shown in Figure 6 while Figure 7 represents the learning convergence of the proposed ensemble approach.

TABLE 5: EACH CLASSIFIER MEAN SQUARE ERROR / TRAINING/ TESTDATA PERFORMANCE

Trail No.	**CLF-1**			**CLF-2**			**CLF-3**			**CLF-4**		
	MSE	**Tr.**	**Test**	**MSE**	**Tr.**	**Test**	**MSE**	**Tr.**	**Test**	**MSE**	**Tr.**	**Test**
1	0.107	100	80	0.198	99	77	0.325	98	82	0.0106	98	82
2	0.0108	99	79	0.0105	98	77	0.0157	97	77	0.0105	98	80
3	0.0122	99	78	0.0205	99	78	0.0312	98	85	0.0016	99	79
4	0.0306	98	74	0.0204	99	78	0.0315	98	83	0.0103	100	80
5	0.0117	100	77	0.0105	98	78	0.0313	98	84	0.0106	100	80
Mean (Tr. / Test)	**99.2000**		**77.6000**	**98.6000**	**77.6000**		**97.8000**	**82.2000**		**99.000**		**80.2000**

TABLE 6: PERFORMANCE OF DIFFERENT TYPES OF ENSEMBLE NEURAL NETWORK FOR HEART DISEASE DATASET

Trial No.	**MJVT Efficiency%**	**MNDS Efficiency%**	**DEENN**				
			CLF-1 WT	**CLF-2 WT**	**CLF-3 WT**	**CLF-4 WT**	**Efficiency %**
1	88.5183	88.8890	0.3535	0.1892	0.1267	0.2745	92.01
2	87.4075	90.1	0.2524	0.3338	0.1977	0.3379	92.02
3	90.1	90.7409	0.3948	0.1858	0.2796	0.1001	93.01
4	88.8890	89.6298	0.1407	0.4415	0.2580	0.2869	92.02
5	88.6847	89.7255	0.2439	0.3350	0.1730	0.3449	92.03

And also efficiency obtained for ensemble neural network with majority voting (MJVT) and mean value integration method (MNDS), over 5 independent trials have been shown in Table 6.

The proposed method (DEENN), has achieved high efficiency of 92%, along with consistent performance over 5 trials.

In the heart disease dataset, 13 attributes define a very complex co-relationship in the decision landscape, which makes the classification with high efficiency, very difficult. But with the proposed system, it can be observed that the outcome even over

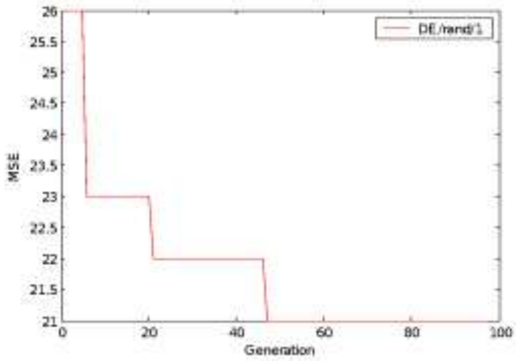

Figure 7.Learning error minimization by Differential Evolution for ensemble

such a complex landscape is very appreciable and also practically acceptable.

Table 7 illustrates the comparison of performances of all the three types of ensemble neural network classifiers (DEENN, MJVT, MNDS) and individual classifiers (CLF1-CLF4). Figure 8, shows the graph of performance comparison with respect to classification efficiency. It can also be observed that ensemble structures are more beneficial when compared to any individual classifier.

Each of the individual classifier performances was also evaluated in terms of sensitivity and specificity to know their predictable capabilities. Sensitivity and specificity performances of different classifier structures have been shown in Table 8. It can be observed that all the individual neural network classifier (CLF 1-CLF 4) have relatively the same performance.

From Table 8, it is clear that ensemble neural network has shown better specificity and sensitivity on the given dataset in comparison with individual NN classifiers (CLF 1-CLF 4), MJVT and MNDS.

TABLE 7: PERFORMANCE COMPARISON OF ALL THE CLASSIFIERS

CLF -1	**CLF -2**	**CLF -3**	**CLF -4**	**MJVT**	**MNDS**	**DEENN**
77.0000	76.6000	81.2000	79.0000	88.6999	89.7969	92.2000

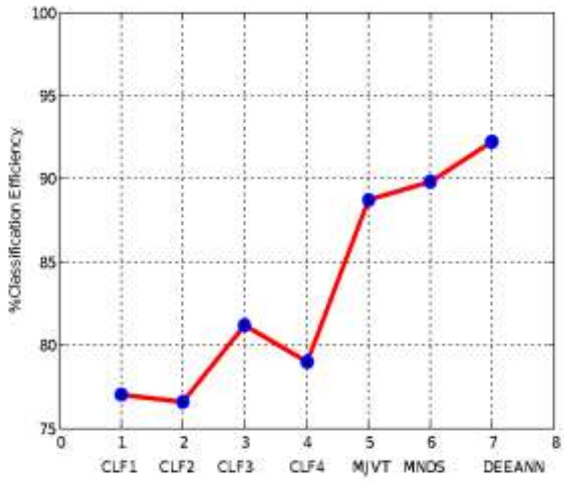

Figure 8. **P**erformance comparison of different classifiers

TABLE 8: DIFFERENT CLASSIFIER STRUCTURES SENSITIVITY AND SPECIFICITY PERFORMANCES

Classifier	**Sensitivity(%)**	**Specificity(%)**
CLF-1	85.84	83.35
CLF-2	84.18	86.01
CLF-3	82.51	92.68
CLF-4	82.51	90.01
MJVT	78.35	97.35
MNDS	84.19	94.02
DEENN	90.02	93.35

V. CONCLUSION

This proposed work shows the efficiency of ensemble neural network in the classification area. Experimental results prove that ensemble classifier is always better than any single classifier. Evolutionary process-based ensemble structure has achieved outstanding efficiency and high-level consistency. When compared to the conventional form of ensemble methods, the proposed method is computationally efficient and also has achieved improvement in quality. Evaluation of proposed work has been done only on single UCI dataset. Further, the algorithm can be enhanced and tested on more application-specific datasets.

ACKNOWLEDGEMENT

We gratefully thank the UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] for providing datasets. We also thank the Visvesvaraya Technological University, Jnana Sangama, Belagavi for giving an opportunity and support for this research work.

REFERENCES

[1] (Asakura, T., Date, Y., & Kikuchi, J. (2018). Application of ensemble deep neural network to metabolomics studies. *Analytica Chimica Acta,1037*, 230-236. doi:10.1016/j.aca.2018.02.045

[2] Bao, W., Yang, B., Li, D., Li, Z., Zhou, Y., & Bao, R. (2019). CMSENN:Computational Modification Sites with Ensemble Neural Network. *Chemometrics and Intelligent Laboratory Systems, 185,* 65-72. doi:10.1016/j.chemolab.2018.12.009

[3] Hansen, L.K., & Salamon, P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(10)*, 993-1001. doi:10.1109/34.58871

[4] Kim, K., Jin, Y., Na, S.-H., & Kim, Y.-K. (2017). Center-shared sliding ensemble of neural networks for syntax analysis of natural language. *Expert Systems with Applications, 83(C),* 215-225. doi:10.1016/j.eswa.2017.04.048

[5] Li, H., Wang, X., & Ding, S. (2016). Research of multi-sided multi-granular neural network ensemble optimization method. *Neurocomputing, 197(C),* 78–85. doi:10.1016/j.neucom.2016.02.013

[6] Manjón, J. V., Coupé, P., Raniga, P., Xia, Y., Desmond, P., Fripp, J., & Salvado, O. (2018). MRI white matter lesion segmentation using an ensemble of neural networks and overcomplete patch-based voting. *Computerized Medical Imaging and Graphics, 69*, 43-51.doi:10.1016/j.compmedimag.2018.05.001

[7] Pulido, M., Melin, P., & Castillo, O. (2014). Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. *Information Sciences, 280,* 188–204. doi:10.1016/j.ins.2014.05.006

[8] Ukita, N., Kawana, Y., Huang, J.-B., & Yang, M.-H. (2018). Ensemble convolutional neural networks for pose estimation. *Computer Vision and Image Understanding, 169,* 62–74. doi:10.1016/j.cviu.2017.12.005

[9] Wang, R., Li, W., & Zhang, L. (2018). Blur image identification with ensemble convolution neural networks. *Signal Processing,155*, 73-82. doi:10.1016/j.sigpro.2018.09.027

[10] Zhang, L., Yu, G., Xia, D., & Wang, J. (2018). Protein-protein interactions prediction based on ensemble deep neural networks. *Neurocomputing. 324*, 10-19. doi:10.1016/j.neucom.2018.02.097

[11] Zhao, Z., & Zhang, Y. (2011). Design of ensemble neural network using entropy theory. *Advances in Engineering Software, 42(10),* 838–845. doi:10.1016/j.advengsoft.2011.05.027

[12] Zhao, Z., Zhang, Y., & Liao, H. (2008). Design of ensemble neural network using the Akaike information criterion. *Engineering Applications of Artificial Intelligence, 21(8),* 1182–1188. doi:10.1016/j.engappai.2008.02.007

[13] Zhou, Z.-H., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence, 137(1-2),* 239-263. doi:10.1016/s0004-3702(02)00190-x

[14] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science..