Combining Artificial Algae Algorithm to Artificial Neural Network for Optimization of Weights

Gülay TEZEL[1]*, Sait Ali UYMAZ[1], Esra YEL[2]
[1]*Konya Technical University, Faculty of Engineering and Natural Sciences, Computer Engineering Department, Konya, Turkey*
[2]*Konya Technical University, Faculty of Engineering and Natural Sciences, Environmental Engineering Department, Konya, Turkey*

***Abstract*—Artificial Neural Network (ANN) is one of the most important artificial intelligent algorithms used for classification problems. The structure of ANN depends on the learning algorithm used for adjusting the weights between neurons of the layers according to the calculated error between model value and the real value. Recently the weights between layers in ANN have been optimized by using metaheuristic optimization algorithms. One of the recent high performance nonlinear optimization algorithms is Artificial Algae Algorithm (AAA) which is a bioinspired, successful, competitive and robust optimization algorithm. In this study, AAA was used as a tool for optimization of the weights in ANN algorithm. ANN and AAA were combined such that the training steps of the ANN modelling to be performed by AAA. After training, ANN continues testing with the optimized weights. The established model combination (AAANN) was tested on three benchmarked datasets (Iris, Thyroid and Dermatology) of the UCI Machine Learning Repository to indicate the performance of this hybrid structure. The results were compared with MLP algorithm in terms of Mean Absolute Error (MAE). Accordingly, up to 96% reduction in mean MSE levels could be achieved by AAANN for all models.**

***Keywords*— Artificial Neural Networks, Artificial Algae Algorithm, backpropagation, classification algorithms, optimization**

I. INTRODUCTION

Artificial Neural Network (ANN) developed by inspiring from the human brain is one of the most popular Artificial Intelligent Algorithms. Human brain is highly complex, nonlinear, and parallel computer (information-processing system) and its information-processing unit is neuron. Similar to brain, the process unit of ANN is neuron. Moreover, neurons are connected to each other with weights called as the memory of ANN. During the training of ANN the weights are optimized to minimize error between desired output and model output by using defined training algorithm. Hence, the selection of training algorithm is crucial. Furthermore, the performance of ANN is interested in its architecture (the numbers of hidden layers and neurons) and weights [1-3].

Manuscript received November 28, 2018; accepted December 25, 2018.
*Corresponding author: gtezel@ktun.edu.tr

Multi-Layer Perceptron (MLP), the most preferred ANN algorithm has one input layer, one or more hidden layer(s) and one output layer. As the number of neurons in input and output layers is determined according to the input and output parameters of dataset, the number of hidden neurons is assigned experimentally. A neuron in any layer of the MLP network is connected with weights to all the neurons in the previous and next layers. The training strategy of MLP is in supervised manner and uses backpropagation algorithm. Back propagation algorithm has two computation passes: forward pass, and backward pass. In the forward pass, input signal is propagated layer by layer using neurons and weights from input layer to output layer. In the backward pass weights are updated to train in accordance with the delta rule. In this pass the error signals are propagated layer by layer starting from output layer through the network, and performed recursively computing the error (i.e., the local gradient) for each neuron [2].

A. Recent tendencies that use metaheuristic methods in weight optimization

In the last decade, to minimize error in training stage of ANN, metaheuristic optimization algorithms (Particle Swarm Optimization (PSO), Genetic Algorithm (GA), etc.) have been used by changing the values of weights for solving problems in different engineering sciences. GA and PSO were the most preferred metaheuristic optimization algorithms for this purpose as they are easily applicable and successful. Momeni et al. developed GA-based ANN model trained with GA algorithm instead of the common backpropagation algorithm to predict the bearing capacity of piles [4]. Montana and Davis indicated that when GA is used with a hybrid model in a sonar image classification problem to train feedforward networks (ANN), training performance increases [5]. Garro and Vazquez used basic and two modified PSO algorithms as both in parameter optimization and in training process of ANN for the classification problems [6]. Gudise and Venayagamoorthy compared PSO and Backpropagation as Training Algorithms for ANN using a non-linear quadratic equation and investigated that PSO has higher training performance [3]. This literature indicated that metaheuristic optimization algorithms can be preferred in weight optimization during ANN training.

B. *Artificial Algae Algorithm (AAA)*

AAA is a recently developed metaheuristic optimization algorithm with its proved high performance in nonlinear optimization [7]. It has been developed by Uymaz et al [7,8] by inspiring the living behaviours of microalgae such that:

- algal cells live in colonies and colony moves as a whole;
- algal colony towards the light with a helical swimming;
- algal cells reproduce themselves by mitosis;
- dominant species may change when adapted to the ambient conditions.

Artificial algae in the model have the same properties. Each individual is represented by an artificial algae colony in the algorithm and this colony is a candidate solution for an optimization problem. The global optimum searching process in the algorithm was composed of 3 main steps called 'Helical Movement', 'Reproduction' and 'Adaptation'. In each cycle colonies of the population were modified in helical movement phase. Colony swims by helical movement (in three dimension) to reach light. As it gets closer to the light it grows up, its energy increases and its movement slows down (due to increased friction force between the colony surface and the surrounding medium). The colonies that approached light search the space more with smaller steps and this increases the local search ability of the algorithm. Algal colonies that could not approach light cannot grow sufficiently, their energies reduce and movement speeds up. Consequently, colonies search with larger steps which increase their global search ability. After completion of helical movement phase, algal cell of the biggest algal colony is replicated by mitosis in evolutionary phase. Algal colonies that could not approach light adapt their dominant species to resemble themselves to the largest colony. These two phases increase the global optimum approximation rate of the algorithm. By all these properties AAA has a strong balance between exploration and exploitation. The performance of AAA in continuous, binary, multi objective function and real world problems have been proven in the literature [7-11].

C. *Purpose of the study*

The purpose of this study was to optimize weights between neurons by using AAA as training algorithm in classification modelling via ANN (MLP). The effectiveness of AAA as training algorithm was investigated for different neuron numbers in hidden layer of MLP and with datasets having different number of classes and attributes.

The broader objective was to achieve a new combination approach for a better training in MLP modelling. The performance of the established AAA–MLP combination (named as AAANN) was compared with single MLP based on mean square error (MSE).

II. MATERIALS AND METHOD

For a fair comparison of MLP and established AAANN, experimental studies were performed on the same platform. All experimental works were conducted by using Matlab (Release R2010a). The computer platform used to perform the experiments was an Intel(R) Core(TM) i7 3.80 GHz processor, 24 GB of RAM, and the Microsoft Windows 8 operating system.

In the study, general scheme of the study indicated in Fig. 1 was followed. Within this scope MLP was used as a reference model to compare the performance of established algorithm (AAANN).

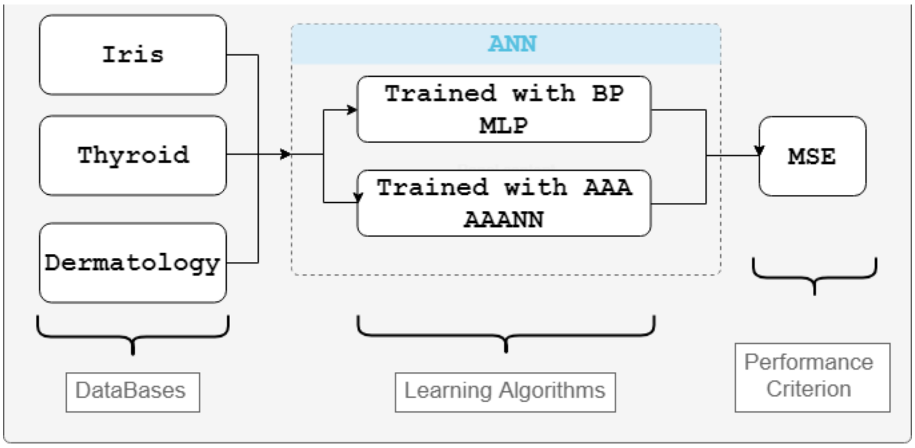


Fig. 1. General scheme of the study.

In both model structures (MLP and AAANN) the following parameters were the same: one number of hidden layer, logarithmic sigmoid activation function of neurons in hidden layer and linear activation function in the output layer. Moreover, in order to observe the effects of number of neurons in hidden layer, two different studies were performed as one with 4 neurons and the other with 10 neurons in both MLP and AAANN algorithms. Three datasets (Iris, Thyroid and Dermatology) were used to indicate the effects of AAA on weight optimization of ANN modelling (Fig. 1).

A. *Datasets*

The main features of three benchmark datasets (Iris, Thyroid and Dermatology) obtained from UCI KDD Machine Learning Repository were shown in Table I [12-

14] (https://archive.ics.uci.edu/ml/datasets.html). The datasets are described as follows:

Iris Dataset: Fisher's Iris dataset consists of three classes each has 50 instances. Each class refers different kind of Iris plant (iris setosa, iris virginica and iris versicolour). Each instance has one class label and four attributes (sepal length, sepal width, petal length and petal width) [12].

Thyroid Dataset: Thyroid gland dataset consists of 215 instances and each instance has five attributes. Dataset contains three different categories (euthyroidism (normal), hypothyroidism (hypo) and hyperthyroidism (hyper)) [13].

Dermatology Dataset: The dermatology database was generated for diagnosis of erythemato-squamous diseases. The diseases are psoriasis, seboreic dermatology, lichen planus, pityriasis rosea, cronic dermatology, and pityriasis rubra pilaris. This database contains 12 clinical features and 22 histopathological features (totally 34 attributes), 33 of which are linear valued and one of them is nominal. The age attribute was removed from the dataset since there are missing values [14].

TABLE I. SUMMARY OF DATASETS

Dataset	Attributes	Classes	Instances (Train/Test)
Iris Dataset	4	3	150 (90/60)
Thyroid Dataset	5	3	215 (129/86)
Dermatology Dataset	33	6	366 (220/146)

B. *AAA-ANN Combination (AAANN)*

The example architecture of ANN is indicated in Fig. 2. In this study, AAA was implemented into ANN to update weights of ANN as training algorithm instead of backpropagation.

Each colony (individual) in AAA represents a vector. This vector consists of both weights between neurons in different layers (w) and weights between bias and neurons (b) in ANN. Each weight in the colony was represented by an algal cell in AAA. At the same time, the number of algal cells in each colony in AAANN indicates the dimension of problem to be solved. Therefore, as the number of weights increases the dimension of the problem increases. Each colony of AAANN algorithm for Fig. 2 is shown as follows:

X_i = [w11,w12,w21,w22,w31,w41,b1,b2,b3]
i=1,2, …, Number of Colony

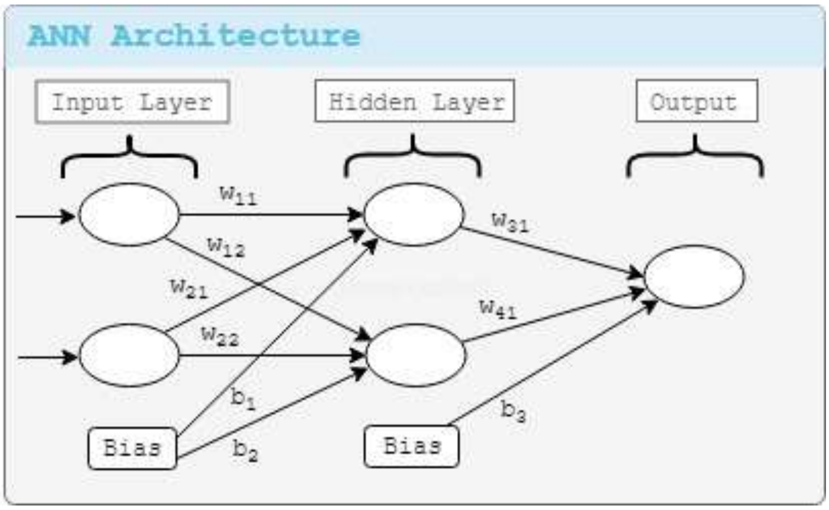


Fig. 2. An example ANN architecture

Training process with AAA was carried out as batch mode learning. In this process, each colony represents a candidate solution and the success of each candidate solution (fitness value) was specified with cumulative error (MSE) (Eq.1) which was computed after all training instances presented to the neural network.

$$MSE = \sum_{i=1}^{N} \frac{(ModelValue_i - ActualValue_i)^2}{2} \quad (1)$$

where N is number of instances in Dataset. At the beginning of the process, random values between [-1 1] were assigned to all colonies (individuals). The recommendations in Uymaz et al. [7] were followed by setting the number of individual in population as 40, the energy loss, e = 0.3, the shear force, Δ = 2 and the adaptation parameter, Ap = 0.5.

Network training was completed when the maximum number of iteration was achieved and the final weight and bias vector by which the MSE value gets its minimum was considered as optimized vector (i.e., the trained ANN model). The flowchart of AAANN is shown in Fig. 3.

In AAANN dimension of the problem (the number of algal cells in each colony) is directly related to the number of input parameters and number of hidden neurons for an output neuron. As the number of attributes of datasets and the number of hidden neurons increase dimension of the problem increases as well. The dimension is calculated by Eq. 2 (Table II).

$$Dimension=(IN+1)*HN+(HN+1)*ON \quad (2)$$

IN: The number of Attributes of Dataset (which is the number of input neurons) (Table I)

HN: Number of Hidden Neurons (either 4 or 10 in this study)

ON: Number of Output Neurons (which is 1 for this study)

TABLE II. THE TRAINING AND TEST PARAMETERS OF AAANN FOR EACH DATASET

Dataset	Number of Hidden Neurons	Number of Iteration	Dimension
Iris Dataset	4 10	250 1000	25 61
Thyroid Dataset	4 10	250 1000	29 71
Dermatology Dataset	4 10	250 1000	141 351

III. RESULTS

The fitness values of 30 runs of MLP and AAANN architectures described in Fig. 3 and Table II were compared for three datasets having different properties. Fig. 4, Fig. 5 and Fig. 6 shows the results of 30 runs of MLP and AAANN for Iris, Thyroid and Dermatology Datasets, respectively. The closer predictions to the real outputs were obtained with AAANN. Although MLP illustrate similar results in a few runs with AAANN it was seen that MLP was unsuccessful in the other runs.

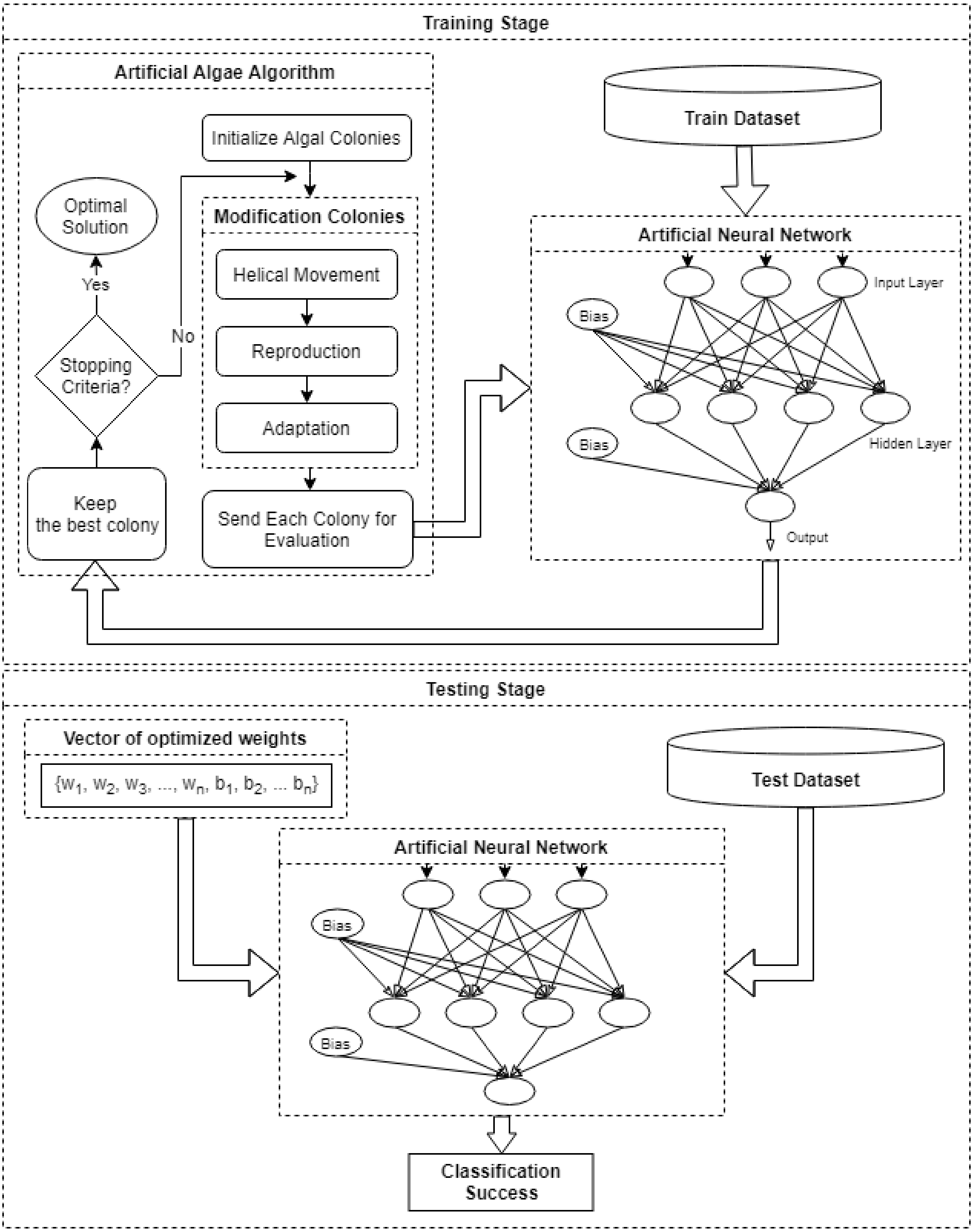


Fig. 3. AAANN flowchart

Iris was the dataset having the least number of attributes and classes, therefore, it is the simplest of three datasets. It is a commonly preferred benchmark problem. Among the three datasets, Iris resulted in the lowest MSE values for both architectures and algorithms and AAANN had better performance than MLP (Fig. 4). The results of 30 runs of AAANN were closer to each other than of MLP.

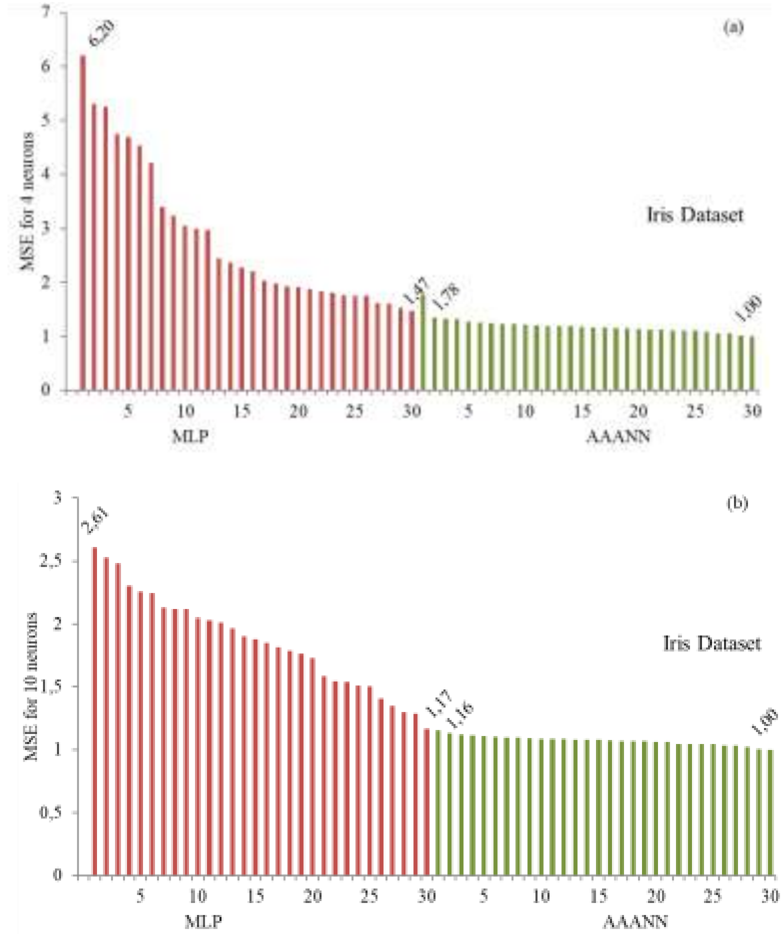
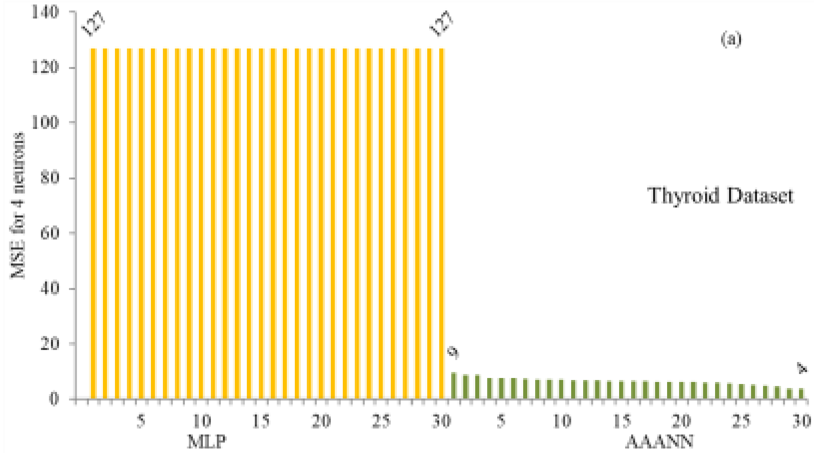


Fig. 4. MSE values of (a) 4 neurons (b) 10 neurons for MLP and AAANN runs with Iris Dataset.

Thyroid dataset is the one that indicated the most prominent success of AAANN (Fig. 5). MLP got stuck into local optima at around 127 MSE value for both 4 and 10 neurons. However, fitness values of 30 runs of AAANN were between 9 and 4 for 4 neurons and between 7 and 3 for 10 neurons (Fig. 5 (a) and (b)). Thyroid dataset has the same number of classes with Iris but more attributes, hence, a little more complex.



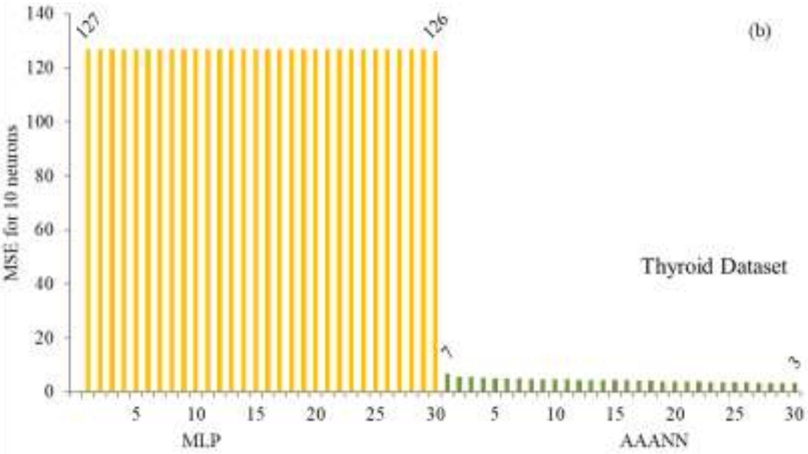


Fig. 5. MSE values of (a) 4 neurons (b) 10 neurons for MLP and AAANN runs with Thyroid Dataset.

Optimization of ANN model weights for Dermatology dataset is highly complex problem among the three datasets with its highest attribute and class numbers. The number of dimension that increase with increasing attributes makes the search space and problem more complex. This complexity reduced the approximation performance of both MLP and AAANN and highest MSE values of three datasets were observed with Dermatology (Fig. 6). The only exception of this result was MLP with 10 hidden neurons, in which MSE values of 30 runs (between 41 and 11) were lower than that of Thyroid dataset (127) which got stuck into local optima.

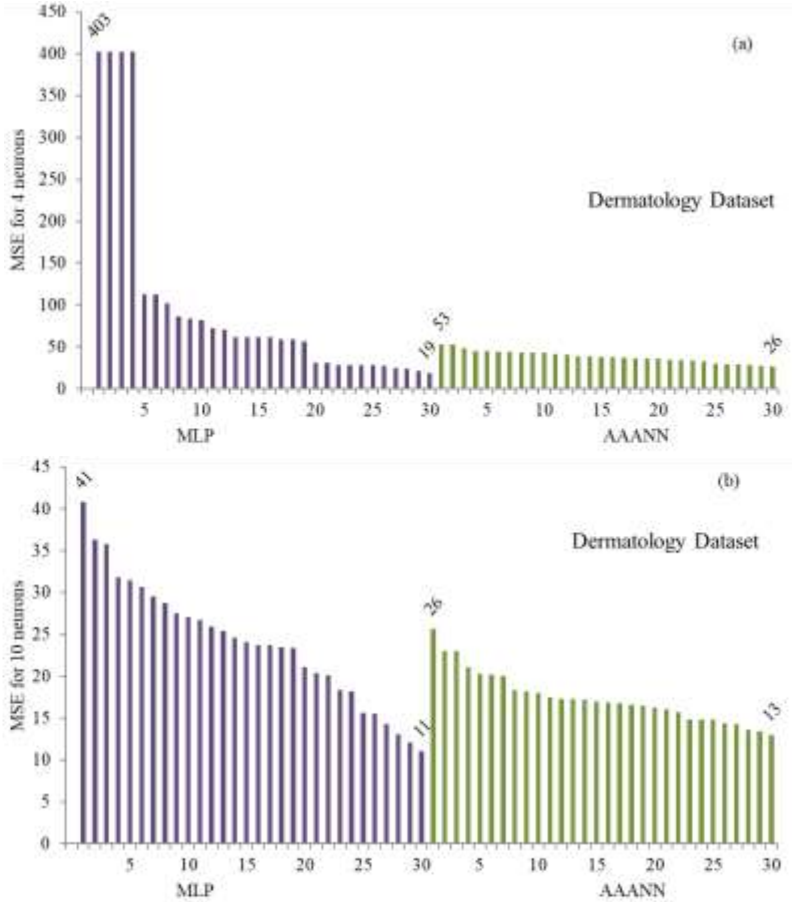


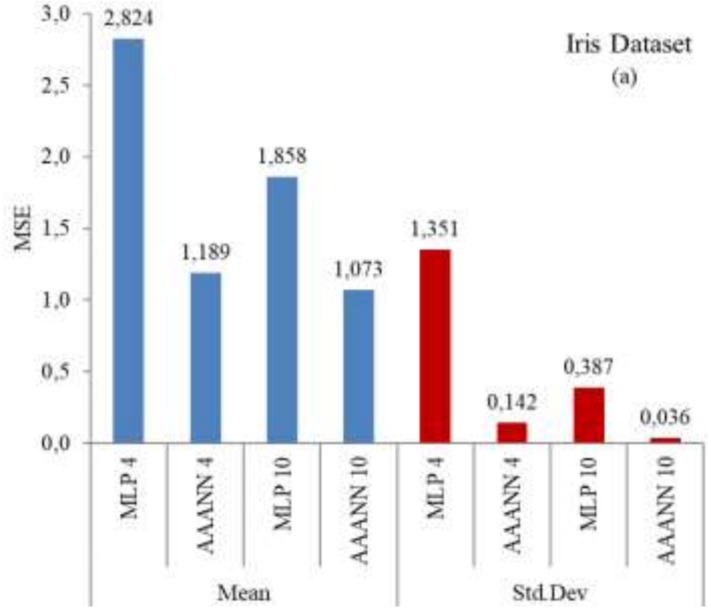
Fig. 6. MSE values of (a) 4 neurons (b) 10 neurons for MLP and AAANN runs with Dermatology Dataset.

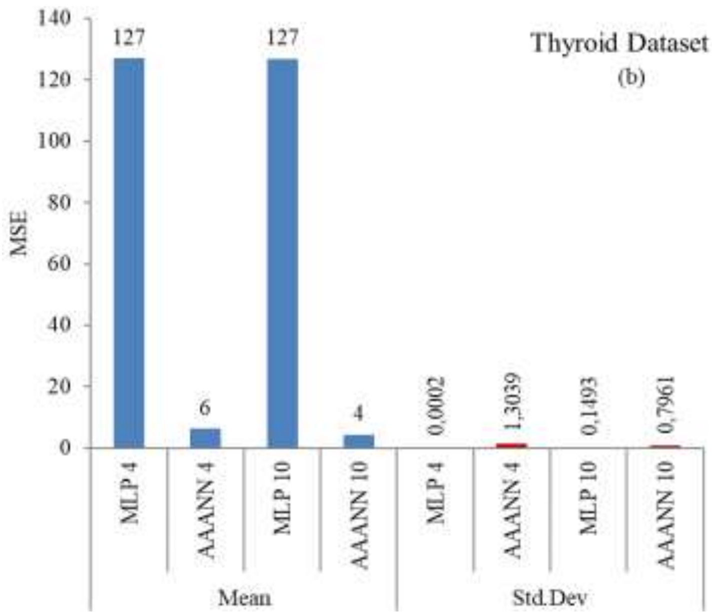
With both MLP and AAANN algorithms, slightly different optimums found in each run as reported in Fig. 4, Fig.5 and Fig. 6. Therefore, in order to compare the ultimate performances of models, the mean of results of 30 runs and their standard deviations (Std.Dev) for each dataset were calculated and compared in Fig. 7. Low mean and low Std.Dev together indicate that the algorithm does not get stuck local solution, it is consistent and it has high accuracy. On the other hand, the low Std.Dev with high mean fitness value indicates that the algorithm gets stuck into local solution, and it is not accurate as in the case of MLP with Thyroid dataset in this study. Mean MSE values of all AAANN models were lower than corresponding

MLP. Moreover, 10 hidden neurons of AAANN models resulted in higher performance. Similarly, Std.Dev values of AAANN runs were lower than MLP models except Thyroid dataset whose Std.Dev is too small because of getting stuck in local optima. Although the high number of parameters increases the dimension in Dermatology database, and makes the problem complex, the proposed AAANN algorithm resulted in high performance with low mean MSE and Std.Dev (Fig. 7(c)).

Moreover, the performance improvements of AAANN were investigated for each dataset in terms of mean MSE and the performance increments of all models were computed in percentages (Table III). Table III shows that the classification success of AAANN was higher than MLP for all datasets at both 4 and 10 hidden neurons. Therefore, the highest performance increase was achieved for thyroid dataset at both 4 and 10 hidden neurons.

As a result, backpropagation is a method for calculating the gradient of the error with respect to the weights for a given input by propagating error backwards through the network in MLP [5]. Although backpropagation works well on simple training problems, in case of increased dimensionality and/or greater complexity of the data, its performance decreases [5]. To overcome this problem, AAA was combined to ANN in this study for weight optimization instead of Backpropagation.





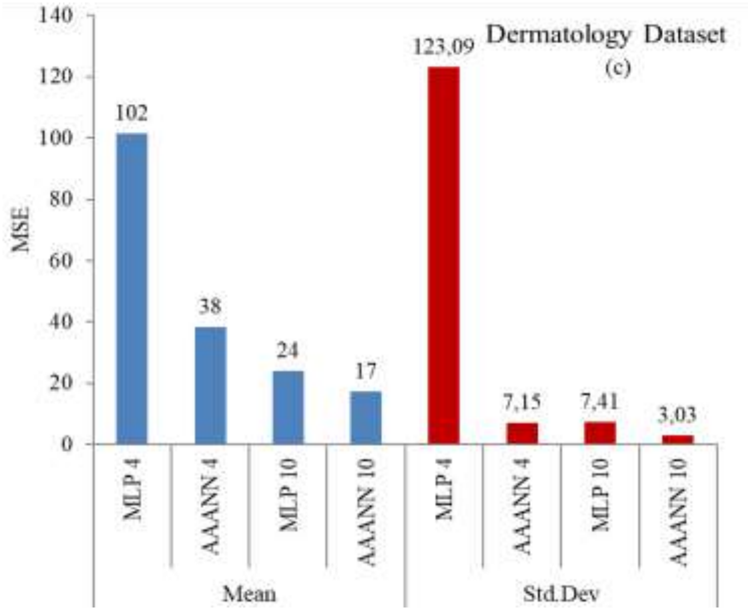


Fig. 7. Comparison of MLP and AAANN runs' mean and standard deviations of MSE values

TABLE III. THE PERFORMANCE INCREASE OF AAANN AS COMPARED TO MLP ALONE

Number of Hidden Neurons	Reduction in MSE		
	Iris	Thyroid	Dermatology
4	58%	95%	62%
10	42%	96%	28%

IV. CONCLUSION

This study proposed a new algorithm AAANN by combining ANN with AAA in order to use the metaheuristic algorithm, AAA, for optimization of ANN weights. The disadvantages of backpropagation, such as getting stuck in local optima or loosing performance with increased complexity, were removed by using AAA in training of ANN.

Three common datasets were utilized to indicate the performance of proposed algorithm. AAANN had better performance over MLP for all datasets and as the number of hidden neurons increased the performance increased as well. AAANN resulted in 58%, 95% and 62% mean MSE reductions for 4 hidden neurons for Iris, Thyroid and Dermatology datasets, respectively. Also, these values were computed as 42%, 96% and 28% for 10 hidden neurons.

Number of attributes, classes, hidden neurons and iterations were the variables tested with one hidden layer ANN in this study. Further studies on the effects of training of ANN with two or more hidden layers with AAA and studies on the performance of AAANN in more complex real world problems are recommended.

REFERENCES

[1] L. Fausett, Fundamentals of neural Networks Architectures Algoritms and Applications, Prentice Hall, Upper Saddle River, New Jersey, 1994.

[2] S. Haykin, Neural Networks: A Comprehensive Foundation, New York: Macmillan, 1994.

[3] V. G. Gudise and G. K. Venayagamoorthy, Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks, Proc. of the 2003 IEEE Swarm Intelligence Symposium, 2003.

[4] E. Momeni, R. Nazir, D. J. Armaghani and H. Maizir, Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN Measurement, 2014, vol. 57, pp. 22–131.

[5] D. J. Montana and L. Davis, Training Feedforward Neural Networks Using Genetic Algorithms, IEEE Int. Conf. on Systems, Man and Cybernetics, 2000.

[6] B. A. Garro and R. A. Vázquez, Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms, Computational Intelligence and Neuroscience, 2015.

[7] S. A. Uymaz, G. Tezel and E. Yel, Artificial algae algorithm (AAA) for nonlinear global optimization, Applied Soft Computing, vol. 31, pp. 153-171, 2015.

[8] S. A. Uymaz, G. Tezel and E. Yel, Artificial algae algorithm with multi-light source for numerical optimization and applications, Biosystems, 2015, vol. 138, pp. 25-38.

[9] X. Zhang, C. Wu, J. Li, X. Wang, Z. Yang, J. Lee and K. Jung Binary artificial algae algorithm for multidimensional knapsack problems, Applied Soft Computing, 2016, vol. 43, pp. 583-595.

[10] S. Korkmaz, A Babalik and M. S. Kiran, *An artificial algae algorithm for solving binary optimization problems*, Int. J. of Machine Learning and Cybernetics 2018, vol. 9(7), pp. 1233–1247.

[11] M. Beskirli, I. Koc and H. Kodaz, *Individual Parameter Selection Strategy for Artificial Algae Algorithm (AAA)*, Journal of Computers, 2018, vol. 13(4), pp. 450-460.

[12] R. A. Fisher, *The use of multiple measurements in taxonomic problems*, Annual Eugenics, 1936, vol. 7(II), pp. 179-188.

[13] J. R. Quinlan, P. J. Compton, K. A. Horn and L. Lazarus, *Inductive knowledge acquisition: a case study*, Proc. of the second Australian Conf. on the App. of Expert Systems, 1986, vol. 1, pp. 183-204.

[14] H. A. Güvenir, G. Demiröz and N. Ilter, *Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals*, Artificial Intelligence in Medicine, 1998, vol. 13, pp. 147.