A Novel Artificial Neural Network (ANN) Using The Mayfly Algorithm for Classification

Syed Kumayl Raza Moosavi
National University of Science and Technology
Islamabad, Pakistan
moosavi7@hotmail.com

Muhammad Hamza Zafar
Capital University of Science and Technology
Islamabad, Pakistan
mee193005@cust.pk

Malik Naveed Akhter
National University of Science and Technology
Islamabad, Pakistan
naveedakhter01@gmail.com

Shahzaib Farooq Hadi
National University of Science and Technology
Islamabad, Pakistan
sfh28450@gmail.com

Noman Mujeeb Khan
Capital University of Science and Technology
Islamabad, Pakistan
mee193004@cust.pk

Filippo Sanfilippo Senior Member, IEEE
Dept. of Engineering Sciences
University of Agder (UiA)
Grimstad, Norway
filippo.sanfilippo@uia.no

Abstract--**Training of Artificial Neural Networks (ANNs) have been improved over the years using meta heuristic algorithms that introduce randomness into the training method but they might be prone to falling into a local minima in a high-dimensional space and have low convergence rate with the iterative process. To cater for the inefficiencies of training such an ANN, a novel neural network is presented in this paper using the bio-inspired algorithm of the movement and mating of the mayflies. The proposed Mayfly algorithm is explored as a means to update weights and biases of the neural network. As compared to previous meta heuristic algorithms, the proposed approach finds the global minima cost at far less number of iterations and with higher accuracy. The network proposed, which is named Mayfly Algorithm based Neural Network (MFANN) consists of an input layer, a single hidden layer of 10 neurons and an output layer. Two University of California Irvine (UCI) database sample datasets have been used as benchmark for this study, namely the Banknote Authentication (BA) and the Cryotherapy, for which the training accuracy achieved is 99.2350% and 96.6102%, whereas the Testing accuracy is 99.1247% and 90% respectively. Comparative study with grey wolf optimization neural network (GWONN) and particle swarm optimization neural network (PSONN) reveal that the proposed MFANN achieves 1-2% better accuracy with training dataset and 2% better accuracy with testing dataset.**

Keywords— Bio-inspired Neural Network, Intelligent Control System, Mayfly Algorithm, Training algorithms, Classification

I. Introduction

Since the advent of artificial intelligence (AI) in the 1950s researchers have proposed the concept of intelligent learning systems that can simulate human capabilities and adaptability. In the late 1990s, intelligent control systems (ICS) overtook the capability of human minds by being able to beat grandmasters at chess [1]. Since then, these systems have been able to accomplish feats beyond the reach of human intelligence. Typical intelligent systems include Machine learning [2], Reinforcement learning [3], Fuzzy control [4], Expert Systems [5], Genetic Control [6], Artificial Neural networks (ANNs) [7].

In many different fields of science and engineering ANNs have become the main research focus because of their inherent nature of knowledge storage and processing of complex data to solve real world practical problems. Application of ANNs appear in the fields of signal processing [8], pattern recognition [9] and many more. Just as how neurons in the human brain react and adapt to the external stimuli provided to adjust the classification, so too do the weights and biases of the ANN adjust their values according to dataset entries and cost minimization algorithms. In other words ANNs are an emulation of the biological neural network of the human brain. These ANNs are basically a set of algorithms purpose of which is clustering/labelling of data. There is a relevant number of training algorithms for ANNs available, each consisting of varying architectures, primarily being used for projection and building models of real world conditions. ANNs are non-linear modeling techniques. They can adapt and adjust with the given dataset without any specificity of the functionality of the data. Traditionally ANNs are based on the error back propagation (BP) algorithm [10]. They consist of an input layer single/multiple hidden layer(s) and an output layer. A single perceptron network consists of one neuron at the hidden layer and one input and output layer. However a single perceptron was incapable of solving non-linear data set patterns therefore multiple neurons and multi-layered hidden networks were proposed by researchers. A set of input features given to such neural networks at the input layer pass through the connection of hidden layers, activation function, weights and biases and are converted into output features. The difference function of the dataset in this predicted output and actual output is called the cost function which is used to adjust the weights and biases through the gradient descent technique [11]. These adjusted weights and biases are sent back into the neural network via the BP algorithm. However such traditional techniques have inherent weaknesses such as a high computational time, slow convergence and finding the local minimum optimum solution instead of the global minima [12].

Improvements to accommodate these shortcomings have been suggested by researchers by modifying the network structure. Common modifications include changing the cost function (Mean Square Error, Rastrigin, etc), activation function such as Weight-and-Structure-Determination (WASD) [13]; a method where the activation function is a set of Chebyshev/Euler polynomials; and replacing the BP algorithm with meta-heuristic algorithms.

II. Meta-heuristic Algorithms

Meta-heuristic algorithms have strong global minima finding capabilities due to the randomness of the technique. Since a large variety of such algorithms are available, they can be surmised into two basic categories:

A. *Single Agent Based*

These algorithms are based on the operation of finding the minimum solution in a search space using a single agent (candidate). Nature inspired heuristics are applied to randomize the transfer function. This is the case of algorithms like Simulated Annealing (SA) [14], Greedy Randomized Adaptive Search (GRAP) [15], etc. Improvement is done on the single agent across the search space until an acceptable solution is obtained.

B. *Multi-agent based*

The inherent weakness of a single agent based solution finding techniques is that they have a low global search ability. A single agent can only search so many instances for optimum solution in a search space. To circumvent such drawback, multi-agents are employed. Each agent learn from each other's position and movement through a complex network of relations to find the global minimum solution in the search space. Examples of population based algorithms include: the Group Teaching Optimization Algorithm (GTOA) [16], swarm intelligence algorithms such as Particle Swarm Optimization (PSO) [17], the Grey Wolf Optimization (GWO) [18], the Grasshopper Algorithm (GA) [19], the Firefly Algorithm [20].

The use of these multi-agent based algorithms can also be seen in the field of deep learning. Updation of weights and biases of an ANN is accomplished through the use of such heuristic algorithms. Several algorithms have already been employed for the purpose of optimizing a neural network but since it can never be certified that a single algorithm is best for every problem, more and more algorithms are introduced every day. Although NNs trained through these algorithms compare better than the BP algorithm, they are still prone to deficiencies that can be improved. Such as in PSO algorithm, it has no evolution operators such as mutation and crossover making it less practical and less accurate or in the case of the GWO algorithm, the search strategy used is mainly based on random walks and thus it cannot always deal with the problem successfully [21]. In this perspective, a novel, multi-agent based, mayfly algorithm [22] has been chosen as the focus of this research for the purposes of training an ANN, namely the Mayfly Algorithm based Neural Network (MFANN).

III. Mayfly Algorithm based Neural Network (MFANN)

A. *Mayfly Algorithm*

The Mayflies are insects belonging to group of insects designated as Palaeoptera. After hatching from their eggs the mayflies grow as aquatic nymphs and once fully grown ascend to the surface and live for only a couple of days to eventually breed and die. In order to mate with a female mayfly an adult mayfly performs a nuptial dance movement around a water body, the female mayflies mate with the males in the air and eventually drop offsprings/eggs and the life cycle continues, as illustrated in Fig. 1. The algorithm is inspired from the movement, dance and mating ritual of the male and female mayflies.

1) Movement of Male Mayflies

Male mayflies gather in swarms around a water body. This means their position and movement velocity is adjusted according to the neighbouring mayflies in the swarm. Consider x_i^t to be the current position of a mayfly 'i' at time step 't'. The next position at time't+1', denoted by x_i^{t+1} can be formulated as (1).

$$x_i^{t+1} = x_i^t + v_i^{t+1} \qquad (1)$$

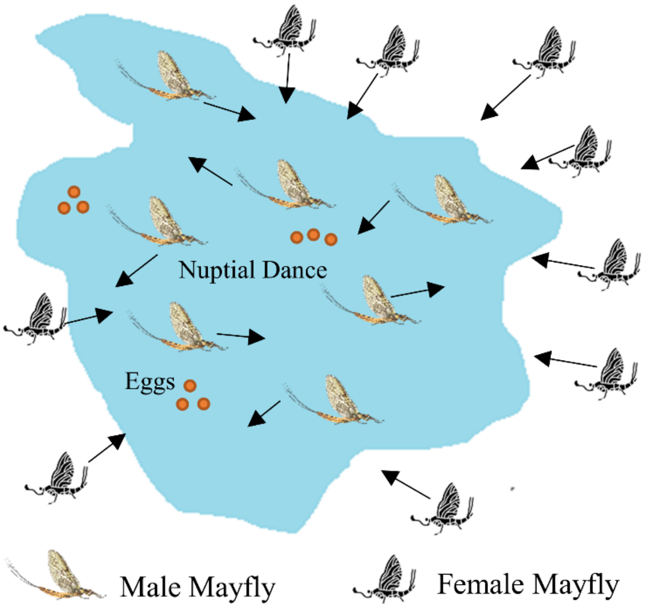

Fig. 1. Mayfly movement and mating ritual.

where v_i^{t+1} is the velocity of the mayfly 'i' at time step 't+1'. The position of the mayfly always remain inbound of the search space (x_{min} , x_{min}) and the velocity of the mayflies will be bounded by the following rule:

$$v_i^{t+1} = \left\{ \begin{array}{ll} V_{max}, & if\ v_i^{t+1} > V_{max} \\ -V_{max}, & if\ v_i^{t+1} < -V_{max} \end{array} \right\}. \qquad (2)$$

Moreover, the equation for the velocity of the mayfly in the swarm is given by the following equation:

$$\begin{aligned} v_i^{t+1} = v_i^t &+ a_1 e^{-\beta r_p^2}(pbest_i - x_i^t) \\ &+ a_2 e^{-\beta r_g^2}(gbest - x_i^t), \end{aligned} \qquad (3)$$

where, β is the visibility coefficient and a_1 and a_2 are positive constants. Furthermore, $pbest_i$ and gbest are the personal best position of mayfly 'i' and global best position of the swarm respectively. The value of $pbest_i$ is only updated if the objective function of x_i^{t+1} is lower than the objective function of $pbest_i$. This is illustrated using the following expression:

$$pbest_i = \left\{ \begin{array}{l} x_i^{t+1}, if\ f(x_i^{t+1}) < f(pbest_i) \\ is\ kept\ the\ same, otherwise \end{array} \right\}. \qquad (4)$$

gbest, on the other hand, is the best position of a mayfly compared to that of the whole swarm of N mayflies. r_p and r_g are defined as the Cartesian distance between x_i and $pbest_i$ and between x_i and gbest respectively. These distances can be calculated using the following equation:

$$r_p\ or\ r_g = \|x_i - X_i\| = \sqrt{\sum_{j=1}^{n} (x_{ij} - X_{ij})^2}, \qquad (5)$$

where x_{ij} is the jth element of the mayfly 'i' and X_{ij} corresponds to $pbest_i$ in case of r_p and gbest in case of r_g. As per the characteristic of the male mayfly, they perform a nuptial dance to attract the female of their species. This movement can be calculated for the best mayflies as it follows:

$$v_i^{t+1} = v_i^t + d * r, \quad (6)$$

where d is the nuptial dance coefficient and r is a random value in the range $[-1,1]$, which also introduces a heuristic element into the algorithm.

2) Movement of Female Mayflies

Let us denote the apparent position of the female mayfly as y_i^t. Females do not gather in swarms, rather they move towards the position of the male in order to breed. The change in this position can be surmised according to the following equation:

$$y_i^{t+1} = y_i^t + v_i^{t+1}, \quad (7)$$

where v_i^{t+1} is the velocity of the female mayfly added to its current position to calculate the female's position at time step t+1. According to a deterministic approach, the ranked female mayflies are attracted to the male mayflies of the same rank. Ranks are appropriated according to the fitness function. Hence their velocities are calculated as it follows:

$$v_i^{t+1} = \left\{\begin{matrix} v_i^t + a_2 e^{-\beta r_{mf}^2}(x_i^t - y_i^t), if\, f(y_i) > f(x_i) \\ v_i^t + fl * r, if\, f(y_i) \le f(x_i) \end{matrix}\right\}, \quad (8)$$

where a_2 and β is the same as mentioned in equation 2. r_{mf} is the distance between male mayfly i and the female mayfly i and fl is a random walk coefficient and r in the range $[-1,1]$.

3) Mating

The offspring are selected the same way as the female mayfly chooses its male mayfly to breed. The best male mayfly mates with the best female mayfly to create and offspring. Similar ranking is followed for all male and female mayflies. The equation of the crossover of the mayflies is determined as it follows:

$$\begin{matrix} offspring_1 = L * x_i^t + (1-L) * y_i^t \\ offspring_2 = L * y_i^t + (1-L) * x_i^t, \end{matrix} \quad (9)$$

where, x_i^t and y_i^t are the parent male and female mayfly respectively and L is a random variable within a specific range.

4) Mutation

Alteration of the offspring is done so as to avoid the algorithm search to get stuck on a local minima. This mutation is done on some of the offspring mayflies by introducing a random variable into the offsprings by using the following equation:

$$offspring'_n = offspring_n + \sigma N_n(0,1), \quad (10)$$

where σ and N_n is the standard deviation and standard normal distribution of mean 0 and variance of 1. Pseudo Code of the algorithm is shown in Fig. 2.

B. Neural Network Classifier (NNC) Design

To construct an NNC, firstly a neural network structure needs to be achieved. The right number of hidden layers and the number of neurons in each hidden layer is paramount for a good classification accuracy. Too few hidden layer variables and the neural network may not be able to classify datasets with high accuracy. Consequently, too many hidden layer variables would cause the network to overfit and computationally would take more time than is necessary. Hence, obtaining the right set of variables for a neural network is essential for highest order of accuracy.

```
initialize male mayfly population $x_i$ (i=1, 2,…, n)
initialize female mayfly population $y_i$ (i=1, 2,…, n)
    evaluate the fitness
    find $g_{best}$ and $p_{best\_i}$
      while (iteration< max_iteration)
            update velocity of male mayfly using Eq. (3) and Eq. (6)
            update velocity of female mayfly using Eq. (8)
            rank mayflies and mate mayflies using Eq. (9)
            evaluate and mutate offspring's using Eq. (10)
            separate male and female
            replace the worst solution with new one
            update $g_{best}$ and $p_{best\_i}$
      end while
return $g_{best}$
```

Fig. 2. Pseudo Code.

Secondly, a suitable activation function needs to be chosen. The activation function determines the strength the neuron will produce and receive. In [23], a comparison was performed between the four activation function shown in the following:

$$\breve{a}_i = ax_i + b \qquad Linear\ Func. \quad (11)$$

$$\breve{a}_i = \frac{1}{1+e^{-x_i}} \qquad Sigmoid\ Func. \quad (12)$$

$$\breve{a}_i = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}}\ Hyberbolic\ Tan.Func. \quad (13)$$

$$\breve{a}_i = \exp\left[-\frac{1}{2\sigma^2}\, \| n_i - m_i \|^2\right] Gaussian\ Func. \quad (14)$$

It concluded that the sigmoid function substantially outperforms the other activation functions, therefore for the purposes of this study the sigmoid activation function is utilized.

Thirdly, the cost function, otherwise known as the fitness function, is the target minimization problem. The connecting weights and biases of a neural network need to be adjusted such that the cost function of the ANN as a whole is minimized. Popular types of cost functions include quadratic cost function (Normal Mean Square Error – NMSE) [24], Cross Entropy (CE), Kullback Leiber Divergence (KLD) and Rastrigin. For the purposes of this study NMSE equation is chosen as the cost function, (15):

$$C.F_i = \frac{1}{N}\sum_{i=1}^{N}(Output_{True} - Output_{Predicted})^2. \quad (15)$$

IV. Design of MFANN

The proposed 3 layered networked structure, as shown in Fig. 3, contains a single hidden layer with 10 neurons (chosen on a trial and error basis). Each neuron in the hidden layer is connected with all the features of the input layer. The weights and biases are updated using the Mayfly Algorithm.

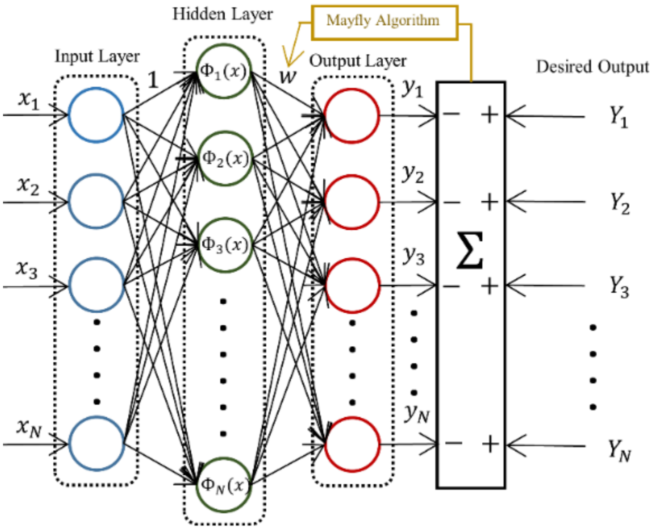

Fig. 3. MFANN Structure.

The input vector $x_i = [x_i^1, x_i^2, \dots x_i^N]$ contains an instance of N input features with the corresponding output value $y_i = [y_i]$. The MFANN algorithm is implemented in Matlab 2018, flowchart of which is illustrated in Fig. 4.

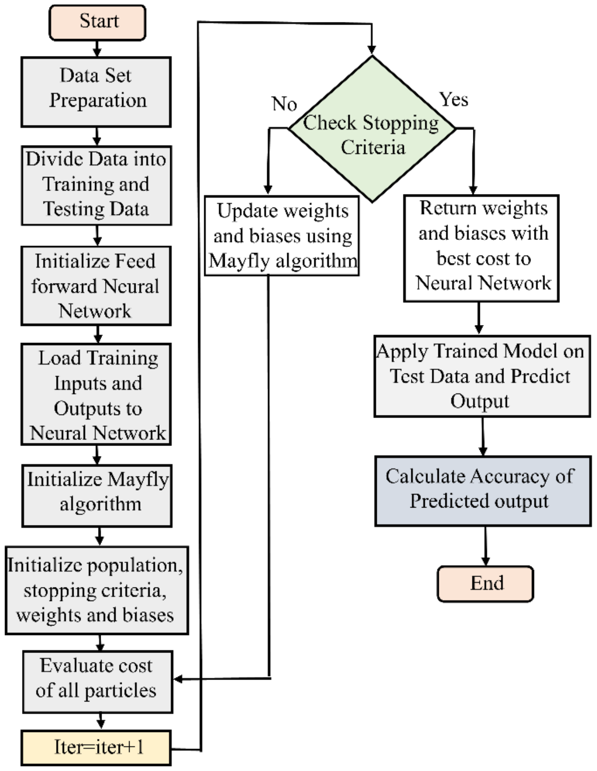
Start
Data Set Preparation
Divide Data into Training and Testing Data
Initialize Feed forward Neural Network
Load Training Inputs and Outputs to Neural Network
Initialize Mayfly algorithm
Initialize population, stopping criteria, weights and biases
Evaluate cost of all particles
Iter=iter+1
Check Stopping Criteria
No
Yes
Update weights and biases using Mayfly algorithm
Return weights and biases with best cost to Neural Network
Apply Trained Model on Test Data and Predict Output
Calculate Accuracy of Predicted output
End

Fig. 4. MFANN Flowchart.

V. Numerical Application and Comparison

Two University of California Irvine (UCI) database sample datasets are selected for the proposed MFANN classification experiments, TABLE I. Banknote Authentication (BA) was extracted from images that were taken from genuine and forged banknote-like 1372 specimens and Cryotherapy dataset contains information about wart treatment results of 90 patients using cryotherapy.

TABLE I. Details of Datasets

Dataset	Number of attributes	Number of classes	Number of instances
Banknote authentication (BA)	4	2	1372
Cryotherapy	6	2	90

In addition, to demonstrate the superiority of the MFA algorithm, a comparison is implemented with other bio-inspired neural network classifiers. The traditional particle swarm optimization neural network (PSONN) [25] and the recently introduced grey wolf optimization neural network (GWONN) will be used as a benchmark for the MFANN. The error cost stipulated for the training of each Neural Network can be seen in Fig. 5 and Fig. 6 which show that in both dataset cases, MFANN has a lower cost. 50 iterations stopping criteria is chosen for these classifiers and the search space is bounded in the range[-5,5]. The dataset is divided into two parts. One is the training set, which makes up two-thirds of the complete dataset, and the other one-third is the testing set. Results of the accuracy's for each algorithm are shown in 0 and TABLE IV. for both datasets. As it can be seen in TABLE II. , MFANN has a very high accuracy primarily because Mayfly Algorithm has the ability to enhance exploration in the search space by introducing the nuptial dance and random walk criterion into the equation. Secondly, by using two different sets of equations (for male and female mayfly) the exploration is further improved.

TABLE II. Classification Results of MFANN

Dataset	Training Accuracy (%)	Testing Accuracy (%)
Banknote authentication (BA)	99.2350	99.1247
Cryotherapy	96.6102	90

Using two input attributes from each dataset as the x and y axis for the scatter plots, the two classes are illustrated in this paper from Fig. 7 to Fig. 12. These figures show the position of true output class against the predicted output class on that same instance of both input attributes. The more the predicted and true outputs overlap on the plot, the higher the accuracy of the model.

This further clarify the superiority of the MFANN over PSONN and GWONN. The two benchmark algorithms used fail to compare better against MFANN because of inherent limitations in the algorithm. For example in the case of PSO, it cannot converge due to the randomness for high-dimensional state space; whereas for GWO, the parameter 'a' decreases with the number of iterations therefore it gets stuck on a local minima.

TABLE III. COMPARISON OF CLASSFICATION FOR TRAINING DATA

Dataset	Training accuracy (%)		
	PSONN	GWONN	MFANN
Banknote authentication (BA)	98.7978	98.3607	99.2350
Cryotherapy	94.9153	93.2203	96.6102
Average Rank	96.85655	95.7905	97.9226

TABLE IV. COMPARISON OF CLASSFICATION FOR TESTING DATA

Dataset	Testing accuracy (%)		
	PSONN	GWONN	MFANN
Banknote authentication (BA)	98.9059	98.2495	99.1247
Cryotherapy	86.6667	86.6667	90
Average Rank	92.7863	92.4581	94.56235

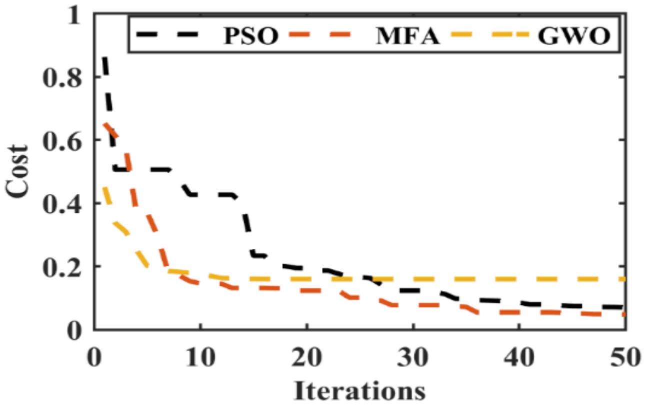

Fig. 5. Banknote Authentication Cost V Iterations.

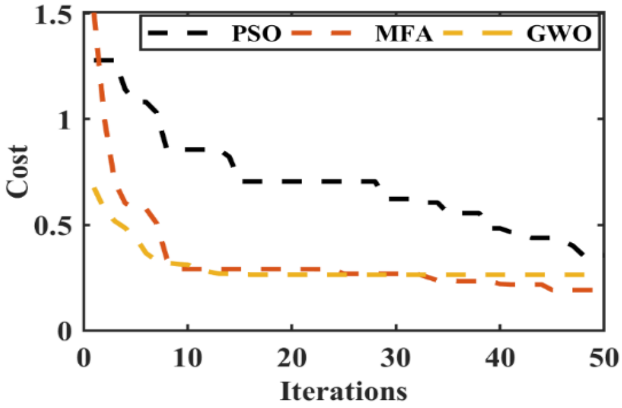

Fig. 6. Cryotherapy Cost V Iterations.

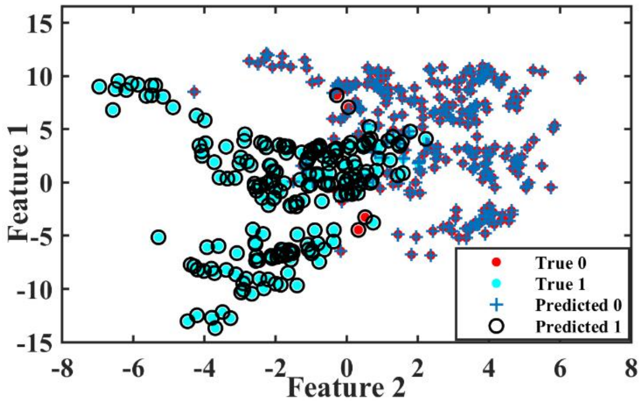

Fig. 7. BA GWO scatter plot of Predicted V Actual Data.

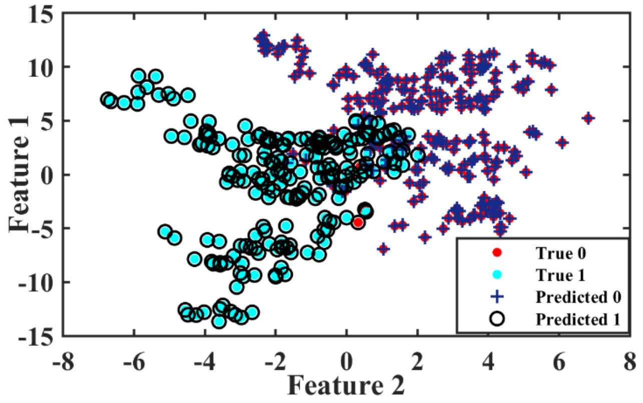

Fig. 8. BA MFA scatter plot of Predicted V Actual Data.

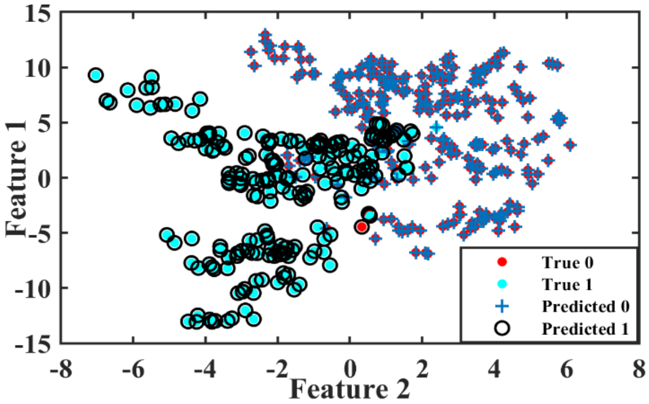

Fig. 9. BA PSO scatter plot of Predicted V Actual Data.

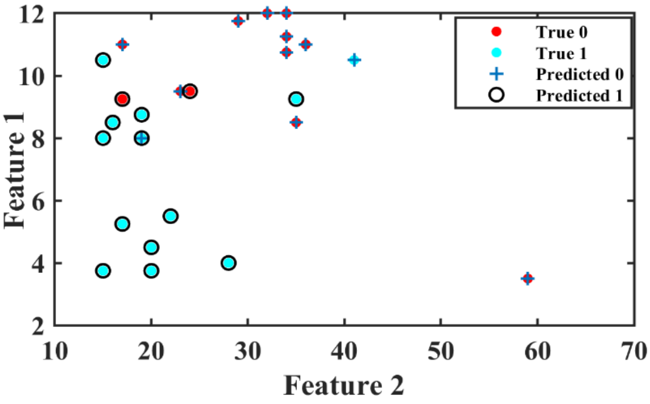

Fig. 10. Cryotherapy GWO scatter plot of Predicted V Actual Data.

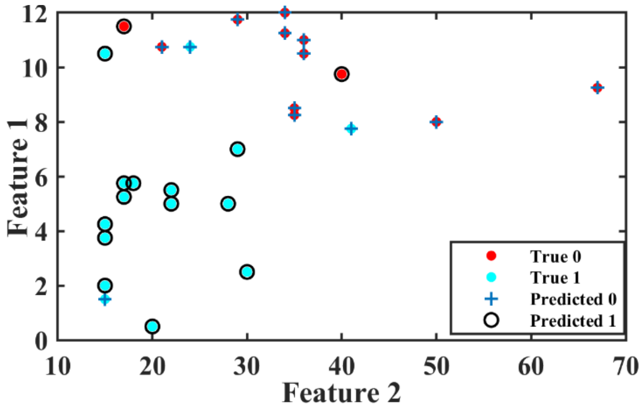

Fig. 11. Cryotherapy MFA scatter plot of Predicted V Actual Data.

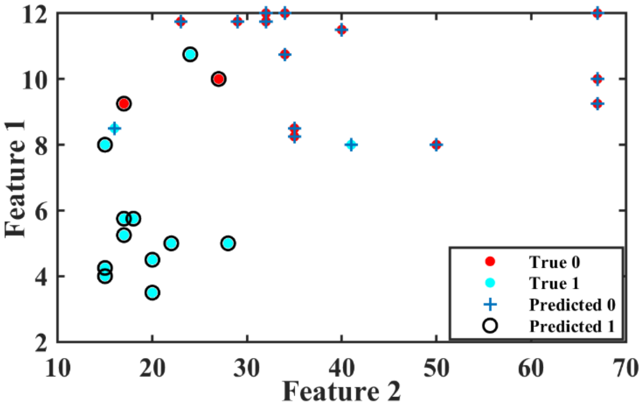

Fig. 12. Cryotherapy PSO scatter plot of Predicted V Actual Data.

VI. Conclusion and Future Work

A novel multi-agent meta-heuristic algorithm for the optimization of ANN namely; Mayfly Algorithm based Neural Network (MFANN) has been proposed in this paper. A three layered network has been chosen for the purposes of this study with 10 neurons at the hidden layer, a sigmoid activation function and a Quadratic Cost Function. To verify the potential of this classifier we have trained and tested this network with two varying datasets namely Banknote Authentication (BA) and Cryotherapy, training accuracy of which is 99.2350% and 96.6102% and the Testing accuracy is 99.1247% and 90% respectively. Comparison with existing multi-agent based neural networks has also been illustrated in this paper to present the effectiveness of this algorithm namely PSONN and GWONN. The results have shown that under the same number of iterations and search space for the multi-agents, classification of the dataset demonstrate 1-2% better accuracy for the training datasets and 2% better accuracy for testing datasets. It is possible to increase the complexity of the neural network by increasing the number of hidden layers and/or neurons and widening the search space to examine the classification of multifaceted datasets.

References

[1] Gibney, E., 2016. Google AI algorithm masters ancient game of Go. Nature News, 529(7587), p.445.

[2] Michie, D., "Memo" functions and machine learning. Nature, 1968. 218(5136): p. 19-22.

[3] Szepesvári, C., Algorithms for reinforcement learning. Synthesis lectures on artificial intelligence and machine learning, 2010. 4(1): p. 1-103.

[4] Nguyen, A.-T., et al., Fuzzy control systems: Past, present and future. IEEE Computational Intelligence Magazine, 2019. 14(1): p. 56-68.

[5] Dong Z, Zhao J, Duan J, Wang M, Wang H. Research on agricultural machinery fault diagnosis system based on expert system. In Proc. of the 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) 2018 May 25 (pp. 2057-2060), IEEE.

[6] Mester, G. and Pletl, S., Genetic algorithm based structure optimization of the fuzzy control system. In Proceedings of the 3rd ECPD International Conference on Advanced Robotics, Intrlligenl Automation and Active Systems (pp. 209-214).

[7] Mester, G. and Pletl, S., Genetic algorithm based structure optimization of the fuzzy control system. In Proceedings of the 3rd ECPD International Conference on Advanced Robotics, Intrlligenl Automation and Active Systems (pp. 209-214).

[8] Sainath, T.N., et al., Multichannel signal processing with deep neural networks for automatic speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017. 25(5): p. 965-979.

[9] Cherry, K.M. and L. Qian, Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature, 2018. 559(7714): p. 370-376.

[10] Cilimkovic, M., Neural networks and back propagation algorithm. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin, 2015. 15: p. 1-12.

[11] Bhama, S. and H. Singh, Single layer neural networks for linear system identification using gradient descent technique. IEEE Transactions on Neural Networks, 1993. 4(5): p. 884-888.

[12] Panchal, G., Ganatra, A., Shah, P. and Panchal, D., 2011. Determination of over-learning and over-fitting problem in back propagation neural network. International Journal on Soft Computing, 2(2), pp.40-51.

[13] Zhang, Y., et al., Cross-validation based weights and structure determination of Chebyshev-polynomial neural networks for pattern classification. Pattern recognition, 2014. 47(10): p. 3414-3428.

[14] Fathollahi-Fard, A.M., et al., A green home health care supply chain: New modified simulated annealing algorithms. Journal of Cleaner Production, 2019. 240: p. 118200.

[15] Exposito-Marquez, A., et al., Greedy randomized adaptive search procedure to design waste collection routes in La Palma. Computers & Industrial Engineering, 2019. 137: p. 106047.

[16] Zafar, M.H., et al., Group Teaching Optimization Algorithm Based MPPT Control of PV Systems under Partial Shading and Complex Partial Shading. Electronics, 2020. 9(11): p. 1962.

[17] Wang, D., D. Tan, and L. Liu, Particle swarm optimization algorithm: an overview. Soft Computing, 2018. 22(2): p. 387-408.

[18] Mirjalili, S., S.M. Mirjalili, and A. Lewis, Grey wolf optimizer. Advances in engineering software, 2014. 69: p. 46-61.

[19] Mirjalili, S.Z., et al., Grasshopper optimization algorithm for multi-objective optimization problems. Applied Intelligence, 2018. 48(4): p. 805-820.

[20] Gandomi, A.H., et al., Firefly algorithm with chaos. Communications in Nonlinear Science and Numerical Simulation, 2013. 18(1): p. 89-98.

[21] Kohli, M. and S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of computational design and engineering, 2018. 5(4): p. 458-472.

[22] Zervoudakis, K. and S. Tsafarakis, A mayfly optimization algorithm. Computers & Industrial Engineering, 2020. 145: p. 106559.

[23] Shenouda, E.A.A., 2006, May. A quantitative comparison of different MLP activation functions in classification. In International Symposium on Neural Networks (pp. 849-857). Springer, Berlin, Heidelberg.

[24] Lin, J.-W. and J.-S. Chiou, Active probability backpropagation neural network model for monthly prediction of probabilistic seismic hazard analysis in Taiwan. IEEE Access, 2019. 7: p. 108990-109014.

[25] Zhao, B., et al., Particle swarm optimized neural networks based local tracking control scheme of unknown nonlinear interconnected systems. Neural Networks, 2020. 134: p. 54-63.