

Training Recurrent Neural Networks by a Hybrid PSO-Cuckoo Search Algorithm for Problems Optimization

Ruba Talal Ibrahim
Computer Science Department,
Faculty of Computer
Science & Mathematics,
Mosul University, Iraq.

Zahraa Tariq Mohammed
Statistics and information Department,
Faculty of Computer
Science & Mathematics,
Mosul University, Iraq

ABSTRACT

Because of computational drawbacks of conventional numerical methods in solving complex optimization problems, researchers may have to rely on meta-heuristic algorithms. Particle swarm optimization (PSO) is one of the most widely used algorithms due to its simplicity of implementation and fast convergence speed. Also, the cuckoo search algorithm is a recently developed meta-heuristic optimization algorithm, which is suitable for solving optimization problems. Normally, the parameters of the cuckoo search are kept constant. This may make algorithm suffering from slow convergence rate. To overcome with this issue, a hybrid algorithm called (PSO-CS classifier) for adjusting the cuckoo search parameters is presented to improved cuckoo search algorithm by particle swarm optimization (PSO) for training recurrent neural network which its weights and bias trained using the (PSO-CS classifier) to deviate from being stuck in local minima for two benchmark classification problems. Moreover, to combine the ability of social communication in PSO with the local search capability of CS. Finally, the performance of the proposed algorithm is compared with that of the standard cuckoo search and PSO Algorithms. The simulation results show that the proposed (PSO-CS classifier) algorithm performs better than other algorithms in decrease number of training errors with a fast convergence rate and high accuracy.

General Terms

PSO, Cuckoo search algorithm, Classification.

Keywords

Particle swarm optimization (PSO); cuckoo search algorithm (CS); Recurrent Neural Networks (RNN); Classification..

1. INTRODUCTION

The artificial neural network (ANN), is a soft computing technique, a powerful prediction tool and information processing system. It has been successfully applied to many manufacturing, finance, military, telecommunication, engineering areas and classify nonlinear problems [1][2]. In fact, feed forward neural networks (FNNs) are the most popular neural networks in practical applications. The architecture of Artificial Neural Networks (ANNs) composed of neurons organized in input, output, and hidden layers in types of topology. The neurons are connected to each other by links which have an associated weights that determine by training, learning or algorithm. The training process is an important aspect of an ANN model when performance of ANNs is mostly dependent on the success of the training process. Each neuron applies an activation function (nonlinear) to its net input (sum of weighted input signals) to determine its output signal. The aim of the training phase is to

minimize a cost function defined as a mean squared error (MSE), or a sum of squared error (SSE), between its actual and target outputs by adjusting weights and biases[3]. An alternate neural network approach is to use recurrent neural networks (RNN) in various applications, such as forecasting of financial data [4], electric power demand[5], and data classification[6].

This neural network have inside feedback loops within network allowing them to store previous memory to train past history [7][8]. Previously, RNN trained by traditional methods like back propagation algorithms which uses the gradient-descent technique and self-scaling method of Oren and Spedicato [9], but several problems can occur in its use in practical applications, such as trapping in local minima, slow convergence rates and takes an enormous amount of time for training a network of even a reasonable size. To overcome these drawbacks many metaheuristic evolutionary computing technique have been used, including genetic algorithm (GA) [10], particle swarm optimization (PSO) [11], and ant colony optimization (ACO) [12]. A metaheuristic can be defined as an iterative generation process that guides a subordinate heuristic by combining intelligently different concepts to explore and exploit the search space.

In (2015)[13], Travis Desell and others used Ant colony optimization for training Recurrent Neural Networks. This strategy is easily parallelizable and can use any neural network training algorithm. Results show that ACO evolved neural networks provided a 63% improvement over airspeed, a 97% improvement over altitude and a 120% improvement over pitch, without requiring additional input neurons from previous time steps. but they did not Investigate different strategies for incrementing/decrementing pheromones. In (2015)[14], Garro and Vázquez presented automatic methodology that designs an ANN using particle swarm optimization algorithms. The aim of these algorithms is to evolve the three principal components of an ANN: the set of weights, the connections or architecture, and the transfer functions for each neuron. In (2015)[3], Jeng-Fung Chen, Quang Hung Do, and Ho-Nien Hsieh presented an efficient training algorithm for feed forward neural networks (ANN) by Particle swarm optimization (PSO) and Cuckoo Search (CS) algorithm to investigate fast convergence speed and have a good ability for finding the global optimum. The experimental results of Approximation Problem and Iris classification problem show that the proposed hybrid algorithm outperforms both PSO and CS in training FNNs. In (2013)[15], Researchers applied a cuckoo search Algorithm (CS) along with a Genetic Algorithm to solve traveling salesman problem (TSP), to overcome the local minima problem and to benefit from the advantages of both types of algorithms. The results show that the hybrid algorithm yields

the best solutions as measured by tour length, quality, iterations, accuracy, and time required. In (2011)[16], Valian and others presented an improved cuckoo search algorithm (ICS) for training feed forward neural networks for two benchmark classification problems (Breast cancer dataset, Iris dataset). The proposed algorithm enhanced the accuracy and convergence rate of the standard cuckoo search algorithm by tuning the value of p_a and α , which are very important parameters in fine-tuning of solution vectors, and can be potentially used in adjusting convergence rate of algorithm.

The results demonstrate the effectiveness of the improved cuckoo search Algorithm. In (2011)[17], Nazri Mohd. Nawi, Abdullah, M. Z. Rehman proposed Cuckoo Search algorithm for training recurrent neural network on the 2-bit XOR, and 4-Bit OR benchmark problems. The results show that the CSRNN is simple and generic for optimization problems and has better convergence rate, SD, and accuracy than the Artificial Bee Colony Levenberg Marquardt (ABC-LM), Artificial Bee Colony Back-Propagation (ABC-BP) and Back-Propagation Neural Network (BPNN) algorithms. In (2007)[18], authors combined Particle Swarm Optimization (PSO) and Back-propagation Algorithms (BP) algorithm to train a recurrent neural network for the identification of impedances of power electronic systems. Simulation results demonstrated that the proposed training algorithm can statistically help avoid the training process being trapped in local minima, without being computationally demanding. But this paper could not determine the optimal division between PSO and BP to achieve a training process with minimal computational resources.

Conventional PSO suffers from the premature convergence problem, especially in complex problems. Cuckoo Search (CS) is a recent heuristic algorithm that has been developed by Yang and Deb [19][20]. This algorithm was inspired by the lifestyle of the cuckoo bird and has been used in solving optimization problems. But, CS may converge slightly slower, but it has better explorative ability. So that, in this paper, a hybrid algorithm of PSO and CS is proposed for training (weights and biases) for many benchmarks of classification problems.

This paper propose a new hybrid metaheuristic search algorithms based on PSO and CS to train recurrent network (RNN) and to make use of the advantages of both PSO and CS algorithms. The main goals of this study are to improve convergence to global minima, decrease the error, the communication ability for the cuckoo bird has been investigated, and accelerate the learning process using a hybridization method. The proposed algorithm called (PSO-CS classifier) that imitate animal behaviour and are valuable for global optimization. [20]. The performance of the proposed (PSO-CS classifier) is verified on selected benchmark classification problems such as IRIS, Glass and Diabetes Classification Problems and compared with standard particle swarm optimization algorithm, cuckoo search algorithm. The remaining paper is organized as follows: Section 2 describes Recurrent Neural Networks (RNN). Section 3 describes PSO algorithm. Section 4 describes Cuckoo Search Algorithm. Section 5 describes proposed algorithm. Section 6 describes Simulation Results. Finally, the paper is concluded in the Section 7.

2. RECURRENT NEURAL NETWORKS (RNN)

Recurrent neural networks (RNN) are a modification on neural networks which have inside feedback loops within

network allowing them to store previous memory to train past history [7], [18]. Unlike feed forward neural networks which only have forward links. Many data classified by recurrent neural networks (RNN) which trained by back propagation gradient methods. However, traditional back propagation algorithm suffers from two main drawbacks, that is, low convergence rate and instability. To overcome the weaknesses of this algorithm, recently many nature inspired metaheuristic algorithms provide derivative-free solution to optimize complex problems such as Bee colony, Ant colony, PSO algorithm, Cuckoo search algorithm and so on.

Basic characteristics of RNN[21]:

- All biological neural networks are recurrent.
- Mathematically, RNNs implement dynamical systems.
- Basic theoretical result: RNNs can approximate arbitrary (term needs some qualification) dynamical systems with arbitrary precision ("universal approximation property").
- Several types of training algorithms are known, no clear winner.
- Theoretical and practical difficulties by and large have prevented practical applications so far.

A state layer (hidden layer) of RNN is updated not only with the external input of the network but also with activation from the previous forward propagation. The feedback is modified by a set of weights to enable regular adaptation through learning [17]. In this papers we used three layer network (see figure1) with one input layer, one hidden or 'state' layer, and one 'output' layer. Each layer will have its own index variable: k for output nodes, j and l for hidden, and i for input nodes. In a feed forward network, the input vector, x is propagated through a weight layer, [17]. Below the questions of RNN :

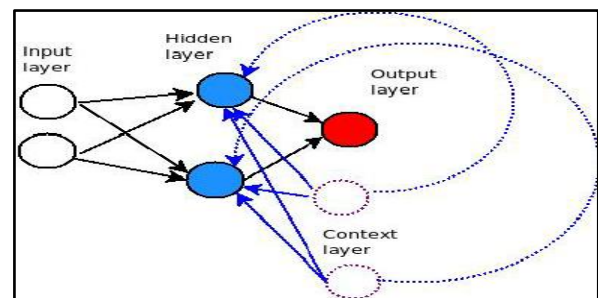


Fig1.show architecture of RNN

$$net_j(t) = \sum_i^n x_i(t)v_{ij} + b_j \quad (1)$$

Where, n the number of inputs is, b_j is a bias, and f is an output function. In a simple recurrent network, the input vector is similarly propagated through a weight layer, but also combined with the previous state activation through an additional recurrent weight layer, U .

$$y_j(t) = f(net_j(t)) \quad (2)$$

$$net_j(t-1) = \sum_i^n x_i(t)v_{ij} + \sum_l^m y_l(t-1)u_{jl} + b_j(t) \quad (3)$$

Where, m is the number of 'state' nodes. The output of the network is in both cases determined by the state and a set of output weights, W

$$y_k(t-1) = g(\text{net}_k(t-1)) \quad (4)$$

$$\text{net}_k(t) = \sum_j^m y_k(t-1)w_{kj} + b_k \quad (5)$$

Where, g is an output function and w_{kj} represents weights from hidden to output layer.

3. PARTICLE SWARM OPTIMIZATION (PSO)

In [22] PSO, intelligent swarms, taken from the behaviour of flocks in the search for food in open space [23]. Each bird begins to search for food in the random areas and then change his position by following the best value obtained in the each cycle search (local best) [24]. In the first cycle of research, the first position is best value that obtained But in the second cycle, PSO start updating the positions and compare the rest of the bird positions with each other and then choose the best position among them (global best) [25]. So the update divided into two category [24]:

1. Local Update.

2. Global Update.

Each bird is considered a potential solution to the problem. Birds' velocity depends on his experience and the experience of its neighbours. PSO finds a better solution in less evolution than the rest of the algorithms. Each bird in the swarm has the following characteristics:

- x_i : is the current position of the bird.
- v_i : the current velocity of the bird.
- i : best position taken by a bird.

Suppose that f represents the fitness function that measure how close the solution to the optimal solution and (t) is the current time, the best position taken by the bird is updated as follows[26]:

$$Pbest_i(t+1) = \begin{cases} Pbest_i(t) & \text{if } f(x_i(t+1)) \geq f(Pbest_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(Pbest_i(t)) \end{cases} \quad (6)$$

The best position in the swarm at time t is calculated as in the equation (7) [26]:

$$Gbest(t) \in \{Pbest_0(t), \dots, Pbest_{n_s}(t)\} | f(Gbest(t)) = \min \{f(Pbest_0(t), \dots, f(Pbest_{n_s}(t)))\}$$

$$Gbest(t) \in \{Pbest_0(t), \dots, Pbest_{n_s}(t)\} | f(Gbest(t)) = \min \{f(Pbest_0(t), \dots, f(Pbest_{n_s}(t)))\} \quad (7)$$

N_s : represents the number of elements in the swarm.

The velocity of the element is calculated as in the equation (8):

$$V_{ij}(t+1) = W V_{ij}(t) + c_1 r_1(t)[Pbest_{ij}(t) - X_{ij}(t) + c_2 r_2(t)[Gbest_j(t) - X_{ij}(t)] \quad (8)$$

$V_{ij}(t)$: represents the velocity of the element i in dimension j where $(j = 1 \dots n_x)$ at time t .

i : represents the position of bird $\{i = 1, 2, \dots, n\}$, where n represents the number of birds in the swarm.

j : $\{j = 1, 2, \dots, m\}$ where m represents the dimensions of the problem.

w : represents the weight of inertia value of $\{0,1\}$.

$x_{ij}(t)$: represents the position of the item i in dimension j at time t .

c_1, c_2 : represent constants.

r_{1j}, r_{2j} : represent random values in the range $(0,1)$ are taken from the uniform distribution, and these values add random to the algorithm.

$Pbest_{ij}(t)$: Represent the best local position visited element since i first time.

Finally the position of bird(i) is updated (x_i) by the following equation:

$$X(t+1) = X_i(t) + V_i(t+1) \quad (9)$$

The previous steps are repeated until the maximum of the specific iterations to be realized within the algorithm or conditions stop-and-so is getting the best solution.

4. CUCKOO SEARCH ALGORITHM (CS)

The Cuckoo Search (CS) developed in 2009 by Yang and Deb [19][20], is a stochastic global search based on population that mimicking animal behaviour, which has been used in a wide range of optimization problems, including NP-hard problems, engineering design, and neural network training. This algorithm is built on the obligate brood parasitic behaviour of certain cuckoo species in conjunction with the L'evy flight behaviour of some birds and fruit flies [19].

Cuckoos have aggressive and peculiar brooding strategy where the females lay their fertilized eggs in the nests of other birds. If the host birds discovered that the egg does not belong to her, it will either throw away or abandoned the nest and build a new one.

Generally, the cuckoo eggs is hatched before the host eggs. Once the first cuckoo chick is hatched, the first instinct action it will take is to remove the host eggs by blindly propelling the eggs out of the nest, which increases the cuckoo chick's share of food provided by its host bird[19].

4.1 L'evy Flights

CS algorithm was further enhanced by the so-called L'evy flights, rather than by simple isotropic random walk methods. Yang and Deep illustrated that CS algorithm based on three assumptions [19]:

1. Each cuckoo sets one egg at a time and dump it in a randomly chosen nest.
2. The best nests with the highest quality of eggs (solutions) carry over to the next generations.
3. The number of available host nests is fixed, and a host has a probability $p_a \in (0,1)$ of discovering an alien egg. In this case, the host bird either throws out the egg or abandons the nest to build a new one in a different location.

The third assumption can be approximated as a fraction: p_a of the n nests replaced with new nests (with new random solutions at different locations). In this Algorithm the fitness of a solution can be the value of the objective function or can be defined in a similar way to the fitness function in genetic algorithms. Each egg in a nest represents a solution, and a cuckoo egg represent a new solution, the aim is to use the new

better solutions (cuckoos) to replace a worse solution in the nests.

Based on the above-mentioned rules, the basic steps of the CS can be summarized as the pseudo(Figure 2):

Cuckoo Search via Lévy Flights

```

Begin
Step1: Objective function f(x), x = (x1,..., xd)T
Step2: Generate initial population of n host nests xi (i = 1,2., n)
While (t < MaxGeneration) or (stop criterion)
Step3: Get a cuckoo randomly by Lévy flights
Step4: Evaluate its quality/fitness Fi
Step5: Choose a nest among n (say, j) randomly
If (Fi > Fj),
Step6: Replace j by the new solution;
End
Step7: A fraction (pa) of worse nests are abandoned and new ones are built;
Step8: Keep the best solutions (or nests with quality solutions);
Step9: Rank the solutions and find the current best
End while
Step10: Post process results and visualization
End
    
```

Fig2: Pseudo code of the Cuckoo Search (CS).

The equation(10) can describe Lévy flight behaviour when generating new solutions x_i(t+1) for the ith cuckoo [27]:

$$x_i(t+1) = x_i(t) + \alpha \oplus \text{Lévy}(\lambda) \quad (10)$$

where $\alpha > 0$ is the step size which should be related to the scales of the problem of interests. The product \oplus means entry wise multiplications which is more efficient in exploring the search space as its step length is much longer in the long run.

The equation (11) describes the Lévy flight behaviour in which the step lengths fit a probability distribution is:

$$\text{Lévy} \oplus u = t^{-\lambda}, \quad (1 < \lambda \leq 3) \quad (11)$$

The Lévy step size is a probability distribution with an infinite variance and the flight steps form a random walk process which obeys a power-law step-length distribution with a heavy tail.

Finally, the optimum solutions that gained from Cuckoo search algorithm are more effective than the best solutions obtained by other swarm intelligence algorithms [19]. CS [28] can be used to perform the local search more efficiently.

5. PROPOSED ALGORITHM (PSO-CS) CLASSIFIER

In order to improve the communication ability for the cuckoo birds, a hybrid algorithm is presented between CS and PSO .from the other hand, to increase the probability of the egg's survival, the cuckoo bird migrates by traversing more search

space. It can successfully search better solutions (best nest) by performing levy flight with PSO.

Below the following steps of proposed Algorithm:

(1)In CS: Initialize the Population Size .The population is composed of a certain number of individuals. Also, set the parameters.

(2)To improve the performance of the CS algorithm and avoid falling in the local minima with fixed values of p_a and α , the proposed algorithm uses variables pa and α . The values of pa and α are dynamically changed with the number of generation because, in the early generations, the values of pa and α must be large to increase the diversity of solution vectors. After that, these values should be decreased in final generations to result in a better fine-tuning of solution vectors. In Equations 12-14, where NI and g_n are the number of total iterations and the current iteration, respectively[16]:

$$p_a(gn) = p_{a \max} - \frac{gn}{NI}(p_{a \max} - p_{a \min}) \quad (12)$$

$$\alpha(gn) = \alpha_{\max} \exp(c \cdot gn) \quad (13)$$

$$c = \frac{1}{NI} \ln \left(\frac{\alpha_{\min}}{\alpha_{\max}} \right) \quad (14)$$

The initial fitness value of the population is calculated by using the objective function.

(4) Then cuckoo search selected the best nest among all via levy flight using the Equation (15) as follow[2]

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{levy}(\lambda) + v_i^{t+1} \quad (15)$$

Where x_i^{t+1} is the velocity vector generated from Equation (16) as follow:

$$v_i^{t+1} = v_i^t + \alpha \varepsilon_n + \beta (g^* - x_i^t) \quad (16)$$

Where v_i^t and x_i^t be the position vector for the particle i , α is the learning parameter or accelerating constant , ε_n is the random vector which is draw from N(0, 1).

(5)Cuckoo Search Mode. An individual updates its position by using the process of CS. An individual in CS has no speed and velocity updating formula, whereas an individual in PSO search mode has both position and velocity. Individual velocity in the cuckoo search mode is not updated, and the current velocity of the individual is the velocity updated by PSO search mode.

(6)Discovery Probability. Through the random elimination mechanism in cuckoo search mode, the host has probability Pa of finding foreign eggs.

(7)PSO-CS classifier finds (weight and biases) and passes them to RNN.

(8) Load the training data.

(9) In first epoch, the weights in RNN are calculated.

(10)In the next cycle, PSO-CS classifier will update the weights with the best possible solution and PSO-CS classifier will continue searching the best weights until either the last cycle/epoch of the network is reached or the MSE is achieved.

(11)The amount of error is determined by:[8]

$$E = (T_K - Y_K) \dots \dots \dots (17)$$

The performances index for the network is given as follows:

$$VA(x) = \frac{1}{2} \sum_{k=1}^K (T_k - X_k)^T (T_k - Y_k),$$

$$V_F(x) = \frac{1}{2} \sum_{k=1}^K E^T \cdot E. \quad (18)$$

In the proposed method the average sum of squares is the performance index and it is calculated as follows:

$$VA_{\mu}(x) = \frac{\sum_{j=1}^N V_F(x)}{P_i} \quad (19)$$

where y_r is the output of the network when the k th input net, is presented. The equation $E=(T_k-Y_k)$ is the error for the output layer, $VA_{\mu}(x)$ is the average performance, $VA_F(x)$ is the performance index, and P_i is the number of Cuckoo populations in i th iteration. At the end of each epoch the list of average sums of square errors of i th iteration SSE can be calculated as follows:

$$SSE_i = \{VA_{\mu}^1(x), VA_{\mu}^2(x), VA_{\mu}^3(x), \dots, VA_{\mu}^n(x)\} \quad (20)$$

The Cuckoo Search is replicating the minimum sum of square error (MSE). The MSE is found when all the inputs are processed for each population of the Cuckoo nest. Hence, the Cuckoo Search nest x_j is calculated as follows:

$$x_j = \text{Min}\{VA_{\mu}^1(x), VA_{\mu}^2(x), VA_{\mu}^3(x), \dots, VA_{\mu}^n(x)\} \quad (21)$$

The rest of the average sum of squares is considered as other Cuckoo nests. (Figure3 shows Pseudo code of PSO-CS classifier):

- Generate initial population of N host nest for $i=1, \dots, n$
- Set all parameters especially p_a and α by equations (12-14)
- While ($f_{\min} < \text{Max Generation}$) or (stop criterion)
- Get a cuckoo bird randomly
- Choose randomly a nest j among n .
- Move cuckoo bird using Equation (15) and (16)
- Calculate the fitness F_i
- If $F_i > F_j$ Then, Replace j by the new solution. End if
- A fraction (probability (pa)) of worse nest are abandoned and new ones are built. - Keep the best solutions (or nest with quality solutions).
- Rank the solutions and find the current best.
- End while

Figure3. shows Pseudo code of PSO-CS classifier

5.1 Training RNN Proposed Algorithm

1. Each best nest represents a possible solution, which is, the weight and the corresponding biases for RNN.
2. The size of a population represent the quality of the solution.
3. In the first cycle, the weights and biases are initialized with CPSO and then those weights are

passed on to the RNN, and then these weights in RNN are calculated.

4. In the second epoch, CS will update the weights with the best possible solution and PSO-CS will continue searching the best weights until either the last cycle/epoch of the network is reached or the MSE is achieved.

6. SIMULATION RESULTS, ANALYSIS AND DISCUSSION

The simulation experiments are performed on 2.40 GHz Core i5 with 4GB RAM using Matlab 2010a software. For investigating less mean square error (MSE) and to achieve high average classification accuracy on testing data, three benchmark dataset like Irises, Glass and Diabetes dataset are taken from UCI Machine Learning Repository. To evaluate the performance of the proposed algorithm, we compare its computational results with those provided by standard CS and standard PSO. For the CS-PSO classifier algorithm, the number of iterations is 1000 for the three benchmark problems. From the input layer to hidden layer and from hidden to output layer log-sigmoid activation function is used as the transform function. After 20 trails, the results of classification are demonstrated in the following below tables. In the tables, the the PSO-CS classifier performs better than the standard CS and standard PSO. parameter 'MSE' represents mean square error 'SD' represents standard deviation of error mean square. The results show that For the PSO Algorithm, $c1$ and $c2$ were set to 2, w decreased linearly from 0.9 to 0.4, and the initial velocities of particles were randomly generated in $[0, 1]$. For the CS algorithm, the number of populations is 10, epochs 1000, trails 20, whereas other algorithm parameters are $\alpha=0.25$, $pa=0.1$ and $\lambda=1.5$. For The parameters of the PSO-CS classifier are $pa(\min)=0.05$, $pa(\max)=0.5$, $\alpha(\min)=0.01$, $\alpha(\max)=0.5$ and $\lambda=1.5$

6.1 IRIS Classification Dataset

The Iris classification datasets is one of the best pattern recognition multivariate dataset used since 1936[2]. Iris dataset consists of 150 examples from three Iris species, i.e. Iris setosa, Iris virginica and Iris versicolor. The classification of IRIS involves the length and width of sepal and petals from each sample of three selected species. The selected network configuration for Iris classification dataset comprises of 9 inputs, 8 hidden, and 2 outputs nodes. A total of 105 instances are used for training the algorithms, while the rest is used for testing purpose. Table 1 illustrates the MSE, SD, and accuracy of the proposed CPSO algorithm when compared with simple CS and PSO.

Table1: Summary of algorithms performance for IRIS

Algorithms	Accuracy	MSE	SD
CS	90.00	0.050	0.0010
PSO	88.07	0.077	0.0020
PSO-CS	99.03	0.0006	0.0001

Based on the simulation results in Table1, it is clear that the proposed PSO-CS classifier model has better performance in terms of MSE, SD, and accuracy. PSO-CS classifier converged with the MSE of 0.0006 and an accuracy of 99.03, compared to other algorithms. It can be clearly seen that the

proposed algorithm has better performance than the CS, PSO in term of MSE.

6.2 Glass Classification Problem

The Glass dataset is used for separating glass splinters in criminal investigation into six classes taken from UCI Repository or Machine Learning database which consists of float processed or non-float processed building windows, vehicle windows, containers, tableware, or head lamp. This dataset is made of 9 inputs and 6 outputs which consist of 214 examples. The selected feed for wardnet work architecture for this network is set to 9-5-6.

Table2: Summary of algorithms performance for Glass Classification Problem

Algorithms	Accuracy	MSE	SD
CS	91.93	1.8E – 06	0.003
PSO	94.09	0.025	0.009
PSO-CS	99.97	2.2E – 03	2.5E – 05

From the above table it is clear to understand that the proposed algorithm outperform the other algorithms. The proposed PSO-CS classifier algorithm achieve small MSE of $2.20E - 03$, SD of $2.50E - 05$, and high accuracy of 99.97. Meanwhile, the CS and PSO algorithms have large MSE of $1.80E - 06$, 0.025, SD of 0.003, 0.009, accuracy of 91.93, 94.09 which is quite lower than the proposed algorithm.

6.3 Diabetes Classification Problem

This dataset consists of 768 examples, 8 inputs, and 2 outputs and consists of all the information of the chemical change in a female body whose disparity can cause diabetes. The feed forward network topology for this network is set to 8-5-2.

Table3: Summary of algorithms performance for Diabetes Classification Problem

Algorithms	Accuracy	MSE	SD
CS	95.04	0.0112	0.01
PSO	91.73	0.188	0.05
PSO-CS	98.00	0.0003	0.0007

It is evident from Table 3 that the proposed PSO-CS classifier Algorithm show better performance than the CS and PSO algorithms in terms of MSE, SD, and accuracy. From Table 3, it is clear that the proposed algorithm has MSE of 0.0003, and SD 0.0007, and achieved 98.00 percent of accuracy. Meanwhile, the other algorithms such as CS and PSO have MSE of 0.0112 and 0.188, SD of 0.01, and 0.05, and accuracy of 95.04 and 91.73 percent, which is quite lower than the proposed algorithm.

7. CONCLUSION

Training of Recurrent neural networks is a well-known and important hard optimization problem, frequently used for classification purpose. Swarm intelligence metaheuristics have been successfully used for such optimization problems. In this paper, we present the advantages of cuckoo search and PSO algorithm such as achieving faster convergence rate,

minimizing the training error, and increasing the accuracy. So, to improve the performance of the CS algorithm and avoid falling in the local minima with fixed values of p_a and α , the proposed algorithm uses variables p_a and α . The values of p_a and α are dynamically changed with the number of generation. From the other hand, the communication ability for the cuckoo birds has been added by introducing PSO, thus giving cuckoo bird the ability to search the best nest that has higher survival. We used a hybrid algorithms CS and PSO to search for the optimal synaptic weights of the neural network in order to minimize the function errors. The testing was done on three well-known benchmark classification problems (Iris, Glass and Diabetes dataset). Since the number of neurons in hidden layers may strongly influence the performance of recurrent neural networks, so we considered different number of neurons in the hidden layers. Finally, as a comparison between the performance of the proposed PSO-CS classifier algorithm with Standard Cuckoo search algorithm (CS) and PSO Algorithm, it can be concluded that in terms of convergence rate and accuracy, the PSO-CS classifier algorithm outperforms the other individual algorithms.

8. REFERENCES

- [1] Paliwal, M.; Kumar, U.A. Neural networks and statistical techniques: A review of applications. *Expert Syst. Appl.* 2009, 36, 2–17.
- [2] Nazri Mohd Nawi, Abdullah Khan, M.Z. Rehman, Maslina Abdul Aziz, Tutut Herawan, and Jemal H. Abawajy, "Neural Network Training by Hybrid Accelerated Cuckoo Particle Swarm Optimization Algorithm", Springer International Publishing Switzerland 2014 .
- [3] Jeng-Fung Chen , Quang Hung Do , and Ho-Nien Hsieh, "Training Artificial Neural Networks by a Hybrid PSO-CS Algorithm", *Algorithms* 2015, 8, 292-308 .
- [4] C. L. Giles, S. Lawrence, and A. C. Tsoi, "Rule inference for financial prediction using recurrent neural networks ,"in *Proceedings of the IEEE/IAFE Conference on Computational Intelligence for Financial Engineering* ,pp .253–259 ,IEEE, March 1997.
- [5] S.Li,D.C . Wunsch, E. O’Hair ,and M. G. Giesselmann, "Wind turbine power estimation by neural networks with Kalman filter training on a SIMD parallel machine," in *Proceedings of the International Joint Conference on Neural Networks(IJCNN'99)*, pp.3430–3434, Washington, DC, USA, July 1999.
- [6] N. M. Nawi, A. Khan, and M. Z. Rehman, "CSBPRNN: a new hybridization technique using cuckoo search to train back propagation recurrent neural network," in *Proceedings of the First International Conference on Advanced Data and information engineering(2013)vol.285 of lecture notes in Electrical Engineering*,pp.111–118,2014.
- [7] L.Fauseeti, *Fundamental of Neural Network Architecture, Algorithm and Application*, Prentice Hall, Englewood Cliffs, NJ, USA, 1994.
- [8] Nazri Mohd Nawi, Abdullah Khan, M. Z. Rehman, Haruna Chiroma, and Tutut Herawan, "Weight Optimization in Recurrent Neural Networks with Hybrid Metaheuristic Cuckoo Search Techniques for Data Classification", *Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015*, Article ID 868375, 12 pages.

- [9] Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: Learning internal representations by error propagation, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Rumelhart, D. E. McClelland, J. L., Eds., MIT Press, Cambridge, 45. (1986) .
- [10] Goldberg, E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison Wesley: Boston, MA, USA, 1989.
- [11] Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
- [12] Dorigo, M.; Maniezzo, V.; Golomi, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybernet.* 1996, 26, 29–41.
- [13] Travis Desell, Sophine Clachar, James Higgins, Brandon Wild, "Evolving Deep Recurrent Neural Networks Using Ant Colony Optimization", *The 15th European Conference on Evolutionary Computation in Combinatorial Optimization* .
- [14] Beatriz A. Garroand Roberto A.Vázquez, "Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms", *Computational Intelligence and Neuroscience* Volume 2015, Article ID 369298, 20 page .
- [15] Ala'a Abu-Srhan and Essam Al Daoud, "A Hybrid Algorithm Using a Genetic Algorithm and Cuckoo Search Algorithm to Solve the Traveling Salesman Problem and its Application to Multiple Sequence Alignment", *International Journal of Advanced Science and Technology* Vol.61, (2013), pp.29-38 .
- [16] Ehsan Valian, Shahram Mohanna and Saeed Tavakoli, "IMPROVED CUCKOO SEARCH ALGORITHM FOR FEEDFORWARD NEURAL NETWORK TRAINING", *International Journal of Artificial Intelligence & Applications (IJAI/A)*, Vol.2, No.3, July 2011 .
- [17] Nazri Mohd. Nawi, Abdullah, M. Z. Rehman, "A NEW OPTIMIZED CUCKOO SEARCH RECURRENT NEURAL NETWORK (CSRNN) ALGORITHM", *Springer-Verlag Berlin Heidelberg* 2011 .
- [18] Peng Xiao, Ganesh K. Venayagamoorthy, and Keith A. Corzine, "Combined Training of Recurrent Neural Networks with Particle Swarm Optimization and Backpropagation Algorithms for Impedance Identification", *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)* .
- [19] Yang XS, Deb S, (2009) Cuckoo search via Lévy flights, *Proceedings of World Congress on Nature & Biologically Inspired Computing*, India, pp 210-214.
- [20] Yang XS, Deb S, (2010) Engineering Optimisation by Cuckoo Search, *Int. J. Mathematical Modelling and Numerical Optimisation*. Vol. 1, No. 4, pp 330–343.
- [21] H. Jaeger (2002): Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. GMD Report 159, German National Research Center for Information Technology, 2002 (48 pp.) .
- [22] Aseel Ismael Ali, Ruba Talal, "UCTP based on Hybrid PSO with Tabu Search Algorithm using Mosul University Dataset", *International Journal of Computer Applications (0975 – 8887)* Volume 91 – No.9, April 2014 .
- [23] S. Kitayama, K. Yamazaki, M. Arakawa "adaptive range particle swarm optimization". Springer Science + Business Media, Journal: *Optimization and Engineering* ISSN: 13894420 Year: 2009 Volume: 10 Issue: 4 pages:575-597 DOI:10.1007/s11081-009-9081-7.
- [24] D. Bratton, J. Kennedy "defining a standard for particle swarm optimization", *IEEE, Journal: 2007 IEEE Swarm Intelligence Symposium* ISBN: 1424407087 Year: 2007 Pages: 120-127 Provider: IEEE Publisher: IEEE DOI: 10.1109/SIS.2007.368035 .
- [25] D. P. Rini, S. M. Shamsuddin, S. S. yuhaniz "Particle Swarm Optimization: Technique, System and Challenges" *International Journal of Computer Applications (0975 – 8887)* Volume 14– No.1, January 2011.
- [26] Engelbrecht A. P., (2007): "Computational Intelligence An Introduction", Second Edition, John Wiley & Sons Ltd, West Sussex, England .
- [27] E.Valian, S. Mohanna and S. Tavakoli, "Improved Cuckoo Search Algorithm for Global Optimization", *International Journal of Communications and Information Technology, IJCIT*, vol. 1, no. 1, (2011), pp. 1-62 .
- [28] R. G. Babukartik and P. Dhavachelvan, "Hybrid Algorithm using the advantage of ACO and Cuckoo Search for Job Scheduling", *International Journal of Information Technology Convergence and Services (IJITCS)*, vol. 2, no. 4, (2012).