IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 December 2023, accepted 7 January 2024, date of publication 24 January 2024, date of current version 12 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3357993

RESEARCH ARTICLE

FNN for Diabetic Prediction Using Oppositional Whale Optimization Algorithm

RAJESH CHATTERJEE[1], MOHAMMAD AMIR KHUSRU AKHTAR[1], DINESH KUMAR PRADHAN[2], FALGUNI CHAKRABORTY[2], MOHIT KUMAR[3], SAHIL VERMA[4], (Senior Member, IEEE), RUBA ABU KHURMA[5,6], AND MARIBEL GARCÍA-ARENAS[7,8]

[1]Faculty of Computing & IT, Usha Martin University, Ranchi 835103, India
[2]Dr. B. C. Roy Engineering College, Durgapur 713206, India
[3]Department of IT, MIT Art, Design and Technology University, Pune 412201, India
[4]Department of Computer Science and Engineering, Chandigarh Group of Colleges, Jhanjeri, Mohali, Punjab 140307, India
[5]Faculty of Information Technology, MEU Research Unit, Middle East University, Amman 11831, Jordan
[6]Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan
[7]Department of Computer Engineering, Automatics and Robotics, University of Granada, 18071 Granada, Spain
[8]Centro de Investigación en Tecnologías de la Información y las Comunicaciones de la Universidad de Granada (CITIC-UGR), 18071 Granada, Spain

Corresponding author: Maribel García-Arenas (mgarenas@ugr.es)

This work was supported in part by the Ministerio Español de Ciencia e Innovación under Grant PID2020-115570GB-C22, Grant PID2022-137461NB-C31, and Grant MCIN/AEI/10.13039/501100011033; in part by Programa Operativo FEDER 2021-2027 under Grant C-ING-027-UGR23; and in part by the Cátedra Fujitsu Tecnología para las Personas (UGR-Fujitsu).

ABSTRACT The medical field is witnessing rapid adoption of artificial intelligence (AI) and machine learning (ML), revolutionizing disease diagnosis and treatment management. Researchers explore how AI and ML can optimize medical decision-making, promising to transform healthcare. Feed Forward Neural Networks (FNN) are widely used to create predictive disease models, cross-validated by medical experts. However, complex medical data like diabetes leads to multi-modal search spaces prone to local minima, affecting optimal solutions. In this study, we focus on optimizing a diabetes dataset from the Pima Indian community, evaluating decision-making performance in diabetes management. Employing multimodal datasets, we compare various optimization algorithms, including the Whale Optimization Algorithm (WOA) and Particle Swarm Optimization (PSO). The test results encompass essential metrics like best-fit value, mean, median, and standard deviation to assess the impact of different optimization techniques. The findings highlight the superiority of the Oppositional Whale Optimization Algorithm (OWOA) over other methods employed in our research setup. This study demonstrates the immense potential of AI and metaheuristic algorithms to revolutionize medical diagnosis and treatment approaches, paving the way for future advancements in the healthcare landscape. Results reveal the superiority of OWOA over other methods. AI and metaheuristics show tremendous potential in transforming medical diagnosis and treatment, driving future healthcare advancements.

INDEX TERMS Feed forward neural network (FNN), oppositional learning, artificial intelligence, meta-heuristic algorithms, whale optimization algorithm (WOA).

I. INTRODUCTION

Machine learning is a branch of science that learns from data and provides insight. One discipline is Artificial Neural Networks (ANN), inspired by neurons in the human brain. The advent of humongous data and excellent computing power helps increase neural network's power and use cases. Out of many different types of neural networks available, feed forward is one of the popular oldest neural networks used today.

In the feed forward neural network, the connection from the input to the hidden to-output layer is one-directional. There might be multiple input nodes, each with a specific weight associated and multiplied individually to sum up the

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

value. Weight is used to assign a certain importance level to each neuron. These are then added with a bias value. The overall summed value is passed to the activation function to obtain the results. Many highly used activation functions include sigmoid, linear function, tanh, softmax, ReLU and leaky ReLU.

Therefore, each neuron is connected with other neurons and helps each other to come up with a given output. The activation value helps neuron to decide if it needs to get fire or not to achieve a given output.

Figure 1, an FNN [1], [2] shows standard input, hidden and output nodes architecture. These nodes are analogues to neurons of Homo sapiens mind. The provided input is the constituent of an input layer, and it captures all the features available as part of the input. We process the input values and go to the hidden layer, which captures the hidden pattern with the non-linearity of data. The output layer is the one that provides the desired result(s) that we want to predict or classify. The most important aspect of an FNN understands the non-linearity, insight available with the data and providing parallelism. There is an increasing number of real-life applications coming up, like classifying pictures [3], analyzing texts [4], and forecasting with time-series analysis.

The potential of a feed forward network arrives with the ability to learn hidden patterns with the data when an FNN is trained well. A well-trained model can solve a given regression or classification problem with optimal fitness value. The training depends on a good model architecture, the selection of hyper parameters, and the choice of the proper optimization algorithm.

The current study focuses on the optimization pillar of these three pillars and understands the impact on the model with a better optimization algorithm. The algorithm aims to converge faster and provide global minima that help in faster model training and prediction with less error rate.

In the literature, several techniques have been proposed to optimize the model. A few notable are gradient descent and meta-heuristic algorithms. When a model starts learning, the issue to address is the scenarios when the model is stuck in local minima and takes longer to converge. There comes the importance of optimization algorithms. It helps find the optimal control parameters that help in reducing the error rate with faster convergence and find the global minima.

Multi-modal search spaces are the kind of issues which do not have one optimum but have many optima which may be a mix of local optima or global optima. In such search spaces, at times the modal stuck in localized minimal value and cannot reach global minimal value. In many studies, optimization techniques are put forward to address aforementioned problem. The variables that dictate faster convergence are optimal metric of weight and bias.

The WOA is a nature-inspired population-based meta-heuristic algorithm based on noteworthy trapping workings called the bubble-net feeding procedure of humpback whales. In recent years, WOA has been applied in different engineering fields. Although the WOA is best suited for training the model mentioned above, it is still stuck in local optima in finding the global minima. This study has incorporated oppositional learning with the WOA to overcome the possibility of getting stuck into local optima and decide the best weight and bias set, which gives the global optima.

A. RESEARCH REVIEW

Metaheuristic algorithms are popular optimization techniques that show encouraging results to solve complex problems that are highly non-linear, have many optima and take time to converge. Some notable optimization techniques are GA (Genetic Algorithm) [5], PSO [6], [7], BAT [8], DE (Differential Evolution) [9], ACO [10], CS (Cuckoo Search) [11], ABC (Artificial Bee Colony) [12], and WOA (Whale Optimization Algorithm) [13]. These optimization techniques are effective during various studies to find optimal control parameters. Thereby, it provides faster convergence and avoids local optima issues, assisting in solving real-world use cases. Like the population-based algorithm, gradient-based back propagation (BP) is popular for training feed forward neural networks and other network architectures. The essence of back-propagation is to propagate the error backwards. It first captures the error or loss incurred in one iteration, and the error is propagated back. This propagation of error helps fine-tune control parameters, i.e. weight and bias. This helps in more generalized learning of the FNN model and thereby helps find the global optima. The process keeps repeating until a satisfied error value is met.

In the study [14], the author has used the back propagation algorithm combined with differential evolution to take multiple parameters like the gross domestic product of gas, total population, total import etc., as the input and rightly predict the energy consumption. The back-propagation is found to have a considerable impact on predicting accurately. The BP works by internally handling the weight and slowly improving to reach closer to predicting correct value. BP is used in gradient descent that mathematically reduces the fitness value (error rate) and helps with learning the model better.

The author Hosseinioun et al. [15], [16] has developed dynamic voltage and frequency scaling (DVFS) technique which is an energy aware method to reduce the usage of energy. This is done applying meta-heuristic technique by mix of Invasive Weed optimization and Culture (IWO-CA). This helps in improved usage of energy.

Malik and Kim [17] has shown the usage of Particle swarm optimization with regeneration based PSO neural network and velocity boost-based PSO NN to manage and minimize the energy consumption. Neural network combined with PSO is found to be effective to get a better accuracy in predicting. In the study from Li et al. [18] has done the research to see the relation between crack depth and surface acoustic wave (SAW). The study uses features generated using laser source scan to the neural network and uses particle swarm optimization as the optimization algorithm. The outcome is very positive, and the model created using neural network with PSO is able to quickly evaluate the crack depths and able correctly use the Scanning Laser Source (SLS) data. Xu et al. [19] did a great work on flood management with accuracy on forecasting rain beforehand.

An approach using LSMT network and PSO helped learning time series data and be better prepared to predict rainfall thereby doing better flood management.

Ilbeigi et al. [20] have proposed method using genetic algorithm in neural network. In the proposal, the author(s) worked to come up with a workable method to optimize the energy consumption of building. First important features are taken for the purpose numerically and this is then put to artificial neural network which is trained and tested to see how its working with varied iterations. Then energy optimization is done using genetic algorithm using those critical variables. It shows a significant improvement and a significant reduction in energy consumption.

In the research from Wang et al. [21], the author used whale optimization technique with artificial neural network to get the best weight and bias for image segregation. Similarly, Li et al [22] showed a whale optimization technique (WOA) which is a modified one which helped in faster global convergence. Kaladevi et al. [23] is able to identify cancer genus cells using meta-heuristic optimization technique.

Musheer et al. [24] employed uses ABC for selecting best features for microarray data. In this study, author has conducted statistical hypothesis test on six cancer datasets to compare the efficiency of the proposed technique to other algorithms. Aziz et al. [25] proposed a new feature selection approach for better classification result in ANNs. Independent component analysis (ICA) is used as an extraction method and ABC as an optimizer. Aziz et al. [26] improved the ICA feature selection using genetic bee colony algorithm.

Zhang et al. [27] show the approach for selecting features of a problem. Feature selection is one of the most important problems in optimization. The binary differential evolution is based on binary operator to focus on feasible optimal spaces and removing the less effective features to help in less computational complexity. The outcome with different dataset shows that it is helping in avoiding local optima and is able to do global exploration. Similarly, Gao et al. [28] show a novel approach using directional permutational differential evolution algorithm to address photovoltaic generation model. The PV generation is a complex process as well as very challenging due to its non-convex, non-linear property with multiple parameters involved. The proposed algorithm is evaluated with six experimental groups on singular, double and triple diode models along with PV models. The PV models are showing a much better result and are able to outperforms the performance accuracy of others.

Lenin et al. [29] used DFO with improved squirrel search optimization (ESSO) algorithm to optimize power losses. In the study [30] a combined WOA and sine cosine algorithm to identify the features of the cellular topology in the chaotic world. The hybrid model is able to work well in getting the performance accuracy and is able to converge fast.

From the outcome of the simulated results, it is observed that proposed approach is more effective than the conventional firefly algorithm.

The Table 1 shows a comparation between different optimization techniques using various factors like convergence rate, avoidance of local optima, dependence on input control parameters, time complexity etc.

Mirjalili et al. [13] developed WOA, an efficient optimizer method. It is based on the preying traits of whales. This follows the phase of exploration and exploitation. Some of the notable points are as follows:

- WOA follows a helix path for hunting the victim.
- This is a distinct feature of the humpback whale that it follows during the exploration phase.
- The next phase is to exploit the target with a randomized value that helps catch the prey in the given phase.
- Overall, WOA does a better job of finding global optima and avoiding local optima.

WOA is increasingly becoming a very popular optimization algorithm, and many use cases are coming up in various studies.

For a more accurate measure of the renewable power resources and the amount of the loss of power, Prakash et al. [31] have implemented WOA. Similarly, Haghnegahdar and Wang [32] have used neural networks with WOA as an optimization algorithm to detect and prevent cyber-attacks for Smart grids, which are intelligent power systems. The author of the paper [33] has used WOA in an ANN model to predict fly rock in 5 different categories, which provides a better result.

The success of meta-heuristic algorithm with oppositional-based learning have inspired to provide a novel optimization algorithm known as OWOA to train FFNN to predict diabetic possibility in a woman based on specific parameters.

The proposed OWOA method is then applied to a neural network with a standard database diabetic problem. The performance is evaluated with other optimization methods like WOA, ABC and PSO. The primary focus of measurement is how effectively these algorithms can avoid local optimum and converge faster. It is good to note that the OWOA optimization algorithm uses the advantage of combining WOA along with OBL. Some well-defined diabetes datasets are considered to evaluate the capability of OWOA to predict diabetes, and outcomes are compared with other standard optimization models. The improved accuracy and the fitness value we can get using OWOA show that it does better than many other meta-heuristic algorithms for diabetic prediction. Using oppositional base learning with the whale optimization algorithm increases the capability of the exploration and exploitation phase to enhance the convergence ability of WOA, which in turn helps in finding the Global Optima more effectively. The efficacy of OWOA has not yet been established for diabetic problems.

TABLE 1. Comparative study of popular meta-heuristic optimization algorithms.

Algorithm.	Description
DE	DE (Differential Evolution) is a metaheuristic algorithm that is known for its fast convergence. However, it can be prone to premature convergence, which means that it may stop searching for the optimal solution too early.
LCA	LCA (Local Competition Algorithm) is a population-based algorithm that is designed to avoid premature convergence. However, it can be computationally expensive.
GA	GA is a metaheuristic technique used for its ability to find global optimal solutions. However, it can be computationally expensive.
SCA	SCA (Stochastic Chaotic Algorithm) is a population-based algorithm that is designed to be computationally efficient. However, it can be less accurate than other algorithms.
PSO	PSO (Particle swarm optimization) is intermediate merging i.e. converging prematurely and cannot reach global maxima. The algorithm combines localized best candidature localized accelerator value, weighing constituent, and globalized accelerator value. It needs lesser no of function checking.
KHA	KHA is an integrated method combining DE and GA. It is known for its fast convergence and accuracy.
DA	DA is nature-inspired technique that is designed to be computationally efficient. However, it can be less accurate than other algorithms.
CSA	CSA is a nature-based technique motivated by the traits of cats. It is known for its fast convergence and accuracy.
FA	FA (Firefly Algorithm) is a nature-inspired technique motivated by traits of candle flies. It is known for its fast convergence and accuracy.
GWO	GWO (Grey Wolf Optimization) is a nature-inspired technique motivated by the social traits of grey wolves. It is known for its fast convergence and accuracy.
WOA	WOA (Whale Optimization Algorithm) is a natured based technique motivated by the traits of humpback whales. It is known for its fast convergence and accuracy.

B. NOVELTY AND CONTRIBUTION

The primary objective for OWOA is to find optimal set of hyper parameters to help with an optimized neural network. By hyper-parameters, we are mainly addressing weights and biases that we can do for WOA, PSO, stochastic gradient descent (SGD), and Adam optimization algorithm (AOA) optimization techniques that are used to optimize control parameters (weights and biases). The proposed OWOA has shown a better performance in the diabetes dataset.

To showcase the performance of OWOA, the algorithm will be assessed with finding global optima, fitness value and some statistical metrics.

Here are the primary contributions of the research:

- To propose a novel optimization technique called Whale Optimization algorithm to identify the optimize weight and bias control parameters.
- To integrate oppositional based learning with WOA algorithm to create larger search space and convergence avoidance.
- The performances of OWOA along with other nature-inspired algorithms are evaluated for faster convergence and time to converge.
- The output is compared and analyzed with other optimization algorithms like CSSO, WOA, and SSO.
- The Diabetes test scenarios are used for the diabetic prediction to carry out.
- The algorithm is evaluated with statistical methods.
- Different metrics are compared to evaluate fitness of the current proposed algorithm.

The rest of the paper is divided into different parts. The mathematical formulation is explained in Section II. Different metaheuristic algorithms are shown in Section III. Section IV shows different oppositional based learning. In section V, we proposed OWOA algorithm. In Section VI, the datasets are described. The experimental setup is discussed in Section 7. Section 8 provides the experimental results obtained from different datasets. Finally, Section 9 discusses the conclusion of the proposed study.

II. PROBLEM FORMULATION

In Figure 1, here $x_n = x_1, x_2, x_3, \ldots$ are set of inputs, $w_n = w_1, w_2, w_3, \ldots$ is weight. The output can be shown as the cumulative value of weight and bias as:

$$O_l = \sum\nolimits_{j=1}^{m} x_m w_{mn} + b_n \; where\; n = 1, 2, 3, \ldots \tag{1}$$

This is the value for a given neuron. Such a set of transferred functions we use to get our desired output which is called activation function. Let us show the activation function as:

$$O = f\,(A_t) \tag{2}$$

where, f = method to activate neurons. In our study, sigmoid function is used for that purpose. The sigmoid method can be shown as:

$$f\,(x) = 1/(1 + e^{-t}) \tag{3}$$

This is a special case of single layered network. Likewise, we can extend the concept to multi-layered network, assume. Input nodes are $(I_1, I_2, \ldots, I_m)$, H = $H_1, H_2, \ldots, H_k$ hidden-nodes and X = $X_1, X_2, \ldots, X_s$ are output-nodes as described in the Figure 1, the activation method of a given hidden node say n^{th} hidden node can be shown as:

$$I_n = f\left(\sum\nolimits_{v=1}^{m} I_v r_{vj} + b_j\right) \; where\; j = 1, 2, 3, \ldots k \tag{4}$$

The output of the node t^{th} can be shown as:

$$O_t = f\left(\sum\nolimits_{n=1}^{k} p_n a_{nt} + e_t\right) \; where\; t = 1, 2, 3, \ldots r \tag{5}$$

where r_{vj} is the weight parameter between r^{th} input node and j^{th} hidden node; a_{nt} denotes the weight between n^{th} hidden node and t^{th} output node; b_j represents the bias of j^{th} hidden node and e_t is the bias of t^{th} output node; m is the number

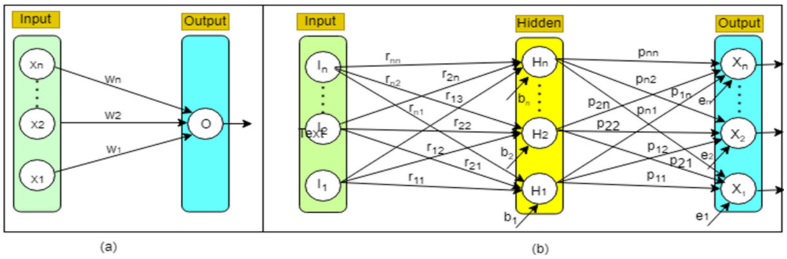


FIGURE 1. Architecture of a) Single layer b) Multilayer FNN.

of input nodes; k is the number of hidden nodes and r is the number of output nodes.

The fitness value be obtained as:

$$F = \sum_{i=1}^{n} \left(O_i^l - S_i^l\right)^2 \tag{6}$$

here n is the samples size and S_i^l refers to i^{th} sample for l^{th} output.

III. OPTIMIZATION TECHNIQUES TO TRAIN AN FNN

The feed forward networks have a very close relationship with the optimization techniques used in computational science. Optimization works to find the control fields that result in minimized fitness value. The FNN model is trained to reduce the error rate in each iteration. There are multiple models used to optimize the fitness function. There are numerous optimization techniques [34] used in literature.

Gradient descent uses the differential function to find the global minima and minimizes the function, whereas the meta-heuristic algorithm uses a population set to find the global minima. The control variables assist the model in learning the underlying model well. The model minimizes control parameters using a better choice of network architecture, a better optimization technique and proper tuning with hyper parameters. Many optimization techniques are proposed in the literature to get minimum control parameters for the best fitness function. The population-based meta-heuristic optimization techniques show promising results with faster convergence and finding global minima. The current literature will discuss the population-based optimization algorithm to train the feed forward neural network.

A. ARTIFICIAL BEE COLONY OPTIMIZATION

Artificial Bee Colony optimization also known as ABC [35], [36] is a popular meta heuristic algorithm used to solve various scientific and real world problems. The algorithm is based on the behavior of bee swarms to intelligently find their food source. The algorithm has three parts as the natural behavior of bees: find all possible location of available food sources which are scout bees, exploiting the sources coming from first set which are employed bees and then evaluate quality of food which are done by onlooker bees.

The application of ABC is seen in multiple areas of health care, telecommunication, infrastructure like monitoring load in smart home, accuracy improvement for cancer diagnosis [37], etc.

The steps to follow for ABC are:

1. To initialize the number of scout bees looking for food origin.
2. To define employed bees that searches new food origin within the neighbor.
3. To define on looker bees to evaluate the quality of food as per the search done by source bees by the employed bees.
4. To scout bees to randomly initiate new solutions.
5. To keep re-iterating steps 2, 3, and 4 to achieve best solution.

B. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a nature inspired algorithm to achieve global. Wang et al. [6] proposed the popular algorithm. This algorithm is developed to mimic the intelligent behavior demonstrated by bird and fishes. Like any other population-based search algorithm, it uses a set of population to search for the best result. The search consists of individual result as well it compares with search from the around candidate. These search agents keep the information about velocity and position which is updated in every iteration. After a number of iterations, it is able to get the best location and position which is considered as the global optimum value.

This can be shown mathematically as:

$$\begin{aligned} f_n^{s+1} &= lf_n^s + d_1.rand_1.\left(qbest_n^s - x_n^s\right) \\ &\quad + d_2.rand_2.\left(qbest - x_n^s\right) \end{aligned} \tag{7}$$

$$x_n^{s+1} = x_n^s + u_n^{s+1} \tag{8}$$

where f_n^s is the velocity of n^{th} particle at iteration s; l is the weight; d_1 and d_2 are acceleration coefficient; $rand_1$ and $rand_2$ are the random numbers between0 and 1; x_n^s is current position of agent n at iteration s, qbest is the n^{th} agent at iteration t and gbest is the overall best solution so far.

C. WHALE OPTIMIZATION ALGORITHM

Mirjalilli et al. [13], [21], proposed a new optimization algorithm called WOA. This is based on the hunting behavior of humpback whale. The hunting behavior is known as bubble-net hunting algorithm. The given whale travels a certain distance original position to new position with helical path by encircling them.

The search agent (individual agents) assumes current position as the optimal solution and shares the current location and based on that updates the optimal solution information. This helps the hump-back whales to reach closer to the prey at each iteration by updating information of the optimal location. This process continues until the whale hunts its trap.

This is a nature inspired algorithm aims at obtaining the global minima with a good number of agents. We initialize the agents with a randomly defined control parameters which are used as the current best position for each of the whale. As we keep iterating, we come close to the optimal solution for all the agents. This is an iterative process that continues until we do not reach close to satisfactory solution. The approach makes WOA a 'goto' algorithm.

In nature, humpback whales hunts fishes using the same method that inspired WOA algorithm. It uses the given method to search for global minima and thereby avoids stuck forever into a local minimum.

The process of hunting a prey is a two-step process, first is to exploit and then to explore. The exploitation step uses reduced spiral path to reduce the distance and exploration continues to find the prey.

The exploitation phase starts when a whale sees a victim and makes spiral path to reach out to the victim. The mathematical expression can be shown as:

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X^*}(i) - X(i)| \tag{9}$$

$$X(i+1) = \overrightarrow{X^*}(i) - \overrightarrow{B}.\overrightarrow{E} \tag{10}$$

i Represents the latest iteration, X∗ Denotes the optimal value, X refers to the latest value, | | represents the absolute value symbol, (\ dot) denotes element-wise multiplication, Y and R constants used in the process, *B* and C variables that determine the shape of the curve in the helical path followed by whales in each iteration to reach the target. B & C are two coefficients that are computed using following equation

$$\vec{B} = 2\vec{\alpha}\vec{r} - \vec{e} \tag{11}$$

$$\vec{C} = 2\vec{r} \tag{12}$$

The value of e is determined by:

$$\alpha = 2 - n * (2/Max\_iter) \tag{13}$$

During each iteration, the parameter α takes values from the set $\{0, 2\}$ and decreases gradually. Additionally, r is a random vector uniformly distributed in the range [0, 1]. The values assigned to B and C directly impact the shape of the curve, leading the whales on their helical path towards the target. Notably, as the value of B decreases during each iteration, the curve continuously tightens and descends further as a result, the algorithm achieves a reduced encirclement in each iteration.

In the context of the current run, denoted by 'n', and with the knowledge of the maximum iteration, represented by 'Max_Iter', the helix shape is constructed based on two crucial values: the current search value 'X' and the best-known value 'X^∗'. These values play a pivotal role in determining the trajectory and overall shape of the path, as described in equation (14).

$$\overrightarrow{X}(i+1) = E'.p^{kr}.\cos(2\pi l)\overrightarrow{X^*}(i) \tag{14}$$

The distance the whale needs to travel to reach its prey is represented by the vector E' = (X∗)(i)-X(i). To ensure the desired shape of the path, a constant 'k' is utilized, which controls the exponential value. Additionally, the variable 'r' takes random values from −1 to 1.

Considering an equal probability for choosing between the processes of reducing encirclement and creating a spiral-shaped path, the next value can be computed as follows:

$$\overrightarrow{X}(i+1) = \begin{cases} Reducing\ Spiral\ path, & if\ t = .5 \\ helical\ path, & if\ \neq .5 \end{cases} \tag{15}$$

The algorithm comprises two main components: the first part of the equation represents the bubble-net technique, while the second part defines the helical path.

In the WOA algorithm, similar to other population-based methods, a balance between exploitation and exploration is achieved through the appropriate adjustment of the values 'e' and 's'.

Extensive literature demonstrates that WOA is effective in solving diverse optimization problems. However, one persisting challenge is the potential trapping in local minima.

Stepwise summary involved in WOA algorithm is as mentioned below:

1. To initialize WOA input parameters.
2. To initialize search agent values randomly.
3. To find fitness value of each search agent.
4. The search agents are sorted in descending order and mark best solution as the "*pbest*".
5. Each iteration, search agents are updated with, i.e., searching and encircling the victim. This done by updating the current position and happens in spherical shape.
6. The fitness value of a given search is compared with the previous values and we update if current value is best.
7. The step 4 to 7 is in continuation till exit criteria are not met. We have taken maximum iteration value as 100 for current study.

D. OPPOSICIONAL BASED LEARNING

In evolutionary algorithms, the initial population is randomly generated within the search space, and the aim is to increase the population's values iteratively to reach better solutions at each step. The search process continues until a predefined benchmark is achieved. In cases where prior information is not available, an initial solution is used, leading to random predicted value that creates a distance between these predicted values and the best solution. This distance plays a role in determining the computational time required for executing the search algorithm. When the initial predicted value is closer to the optimal solution, the algorithm converges quickly. If not, it takes more time to converge.

Madhavi et al. [38] introduced a highly efficient computational intelligence concept to increase optimization algorithms, known as oppositional-based learning (OBL). OBL has been incorporated into various evolutionary algorithms for better convergence and enables finding the global optima for optimization problems. The fundamental idea behind OBL is to utilize the current value and compare it with its opposite value. This comparison allows the rapid identification of better solutions. By evaluating the opposite values at the same time, the algorithm has a better chance of converging to positions closer to global solutions quickly. Therefore, OBL is a powerful way to improve the final solution in many optimization tasks. The definition of OBL is mentioned below.

1) OPPOSITE NUMBER

Opposite number x' of a real number x is mathematically defined as:

$$x' = x + y - v \textit{ where } v \in [x, y] \tag{16}$$

2) OPPOSITE POINT

Let X = $x_1 + x_2 + \ldots + x_n$ be a point in n dimensional space, where $x_k \in$ X and $r_k \in [a_k, b_k]$, opposite number can be evaluated as:

$$x_k' = a_k + b_k - x_k \tag{17}$$

Using aforementioned opposite point concept, oppositional based learning (OBL) concept is described.

3) OPPOSITION BASED OPTIMIZATION

Let X = $x_1, x_2, \ldots, x_n$ be possible solution space in n-dim space with g(X) with the function to evaluate fitness and $X^{`}$ is the opposite value of X whose function to evaluate fitness is g($X^{`}$). As per maximization theorem, if g (X)≤ g ($X^{`}$), then X can be replaced by $X^{`}$. The combined data points which are existing space and opposite space can be used to find the better fitness function.

Therefore, oppositional based learning has primarily two important parts (a) how to initialize the opposite space and (b) at each iteration, create opposite points by generation jumps. Here is how we can describe it.

a: OPPOSITION BASED INITIALIZATION OF OPPOSITE SPACE

The initialization is done using algorithm shown in Algorithm 1.

b: OPPOSITION BASED GENERATION BY JUMPING PROBABILITY

In the case of jumping probability, the process is similar to earlier one, here to make the learning more optimized we forcibly changes the present population to the opposite one with each iteration and randomly jump to a new solution value that may be a better solution than the present one.

Algorithm 1 Opposition Value Initialization

```
FOR J = 1 to n_x:
    FOR l = 1 to n_y:
        OP_{j,l} = u_j + v_l − X_{j,l}
    END FORLOOP
END FORLOOP
```

Here, n_x = population of the sample, and n_y = total n° of control variables

TABLE 2. Training data to train the FNN with diabetes dataset.

Dataset	Input Node	Hidden node	Output Node	Total Training Data
Diabetic FNN	**8**	**7**	**1**	**580**

But using randomized value, here we use jumping rate Y_t as described in TABLE 2 for opposite population initialization:

Algorithm 2 Jumping Based Initialization

```
IF randomVal = randomly selected value between 0 and 1
THEN
    FOR J = 1 to n_x:
        FOR l = 1 to n_y:
            OP_{j,l} = u_j + v_l − X_{j,l}
        END FORLOOP
    END FORLOOP
ENDIF
```

Here, n_x = population of the sample, and n_y = total n° of control variables.

IV. PROPOSED OPPOSITIONAL LEARNING BASED WOA FOR FNN

The accuracy and convergence to diagnose a diabetic person using the Pima Indian data set shows a better result when OBL is introduced with the WOA. The model presents the opposite learning space in the whale optimization algorithm and the current learning to improve the model accuracy. The algorithm follows the defined steps herewith:

- **Step 1**: A given number of search agents are chosen using the OBL concepts.
- **Step 2**: To choose an Artificial neural network with a set of input nodes, one hidden layer and one output layer. The network architecture that we use is dependent on input and output values. The hidden layers should be chosen wisely to avoid over fitting or under-fitting problems. The control parameters are weight and bias, which depend on the number of inputs and hidden and output nodes we choose. The weights and biases (control parameters) are randomly initialized.

TABLE 3. Least fitness metric of the proposed algorithms for diabetes dataset.

Control Variables	OWOA	ABC	WOA	PSO
Wgt1	4.150E-02	-2.561E+00	-2.24E+01	-2.26E+00
Wgt2	5.551E-03	1.671E-02	7.61E-01	2.04E-01
Wgt3	1.853E-03	2.580E+01	-8.90E-01	1.04E-01
Wgt4	-5.790E-03	-2.617E-01	3.50E-04	1.46E-03
Wgt5	4.270E+00	4.900E-02	2.31E-01	2.57E-01
Wgt6	-2.583E-01	-4.663E-01	9.62E-01	-6.81E-05
Wgt7	1.401E+01	-2.811E-02	-7.15E-02	6.11E-01
Wgt8	1.680E+00	-1.324E+01	3.05E-04	-2.78E+00
Wgt9	-4.800E+12	-5.468E-02	1.25E+00	1.37E-01
Wgt10	1.552E+03	-8.983E-02	1.15E-01	-3.83E-03
Wgt11	-7.373E+01	-1.070E+00	-7.76E-01	6.06E-01
Wgt12	-7.630E-03	-4.591E-02	1.60E+00	-6.35E-03
Wgt13	4.223E+00	1.583E-01	-7.55E+02	-1.37E+00
Wgt14	-6.750E-02	-7.971E+01	1.51E-02	-3.38E-05
Wgt15	-3.272E-01	7.849E-05	-6.16E-02	6.81E-03
Wgt16	-6.042E-01	2.980E-01	3.85E-03	-3.52E-03
Wgt17	-1.220E+01	9.192E-02	1.82E-03	-3.97E-02
Wgt18	1.114E+01	2.691E-01	-4.26E-03	-7.18E-03
Wgt19	2.783E-01	-8.791E-03	1.36E-02	-3.41E-01
Wgt20	4.060E-07	-4.070E-01	-1.86E-02	-1.28E-01
Wgt21	2.094E+00	-4.983E-01	1.84E-04	1.65E-02
Wgt22	-7.112E-01	-3.631E+01	1.68E+01	1.43E-01
Wgt23	4.420E-01	-2.142E-02	5.36E-01	-1.69E-02
Wgt24	-1.471E+00	-1.281E+00	-9.76E-02	3.06E-01
Wgt25	-4.663E+01	2.514E-01	-2.77E-02	-1.63E-02
Wgt26	-9.773E-03	1.070E-01	7.53E-03	-1.59E-05
Wgt27	-3.050E-01	1.565E-01	-2.36E-01	-2.88E-02
Wgt28	-7.680E-01	-1.648E+01	5.08E+00	-2.38E-03
Wgt29	-2.971E-01	-3.990E-01	-5.75E-04	1.84E+00
Wgt30	1.430E+00	1.817E-01	-1.50E-02	1.76E-01
Wgt31	1.700E-01	1.243E+00	1.97E+01	-9.73E-03
Wgt32	5.700E+00	-1.231E-02	3.19E-02	6.49E-03
Wgt33	9.400E-01	-4.937E-03	-1.66E-01	8.08E-01
Wgt34	-1.292E+00	-6.315E+01	-1.03E-01	-3.81E-05

- **Step 3**: The upper, lower and maximum iteration values are initialized.

TABLE 3. *(Continued.)* Least fitness metric of the proposed algorithms for diabetes dataset.

Wgt35	-4.851E-03	8.264E+01	1.62E+00	7.21E-01
Wgt36	2.481E-03	1.032E-01	1.12E+00	-2.41E-03
Wgt37	4.100E+00	2.621E-04	3.56E-03	2.21E-01
Wgt38	-8.551E-01	-9.949E-03	1.47E-01	8.60E-03
Wgt39	2.463E-01	-8.284E-01	-4.63E-01	1.03E+00
Wgt40	2.751E+01	-7.241E+00	2.01E-01	1.45E-02
Wgt41	1.973E+00	6.234E-03	-1.27E-01	-8.65E-01
Wgt42	5.721E-01	8.161E-01	7.51E+00	4.00E-03
Wgt43	6.971E-01	7.044E-03	-3.93E-04	4.68E-02
Wgt44	6.351E+00	1.950E-01	6.33E+00	3.54E-01
Wgt45	4.970E+01	-1.623E+00	2.80E-01	1.68E-01
Wgt46	4.711E-02	-5.851E-01	9.89E-03	-4.53E-03
Wgt47	4.791E-02	2.210E+00	-6.41E-03	1.17E+00
Wgt48	2.081E+04	3.481E+01	-2.63E-02	7.40E-05
Wgt49	-2.571E-02	1.124E-02	-2.19E-02	5.42E-02
Wgt50	3.000E+00	-1.851E-01	5.34E-02	-2.24E-03
Wgt51	-6.160E-01	-4.500+00	-2.30E+00	8.61E-02
Wgt52	-2.751E+16	-5.681E-01	2.38E-02	3.74E-02
Wgt53	-1.091E+00	-3.432E-02	3.68E-01	5.09E-04
Wgt54	4.682E-03	8.971E-02	2.91E+00	-2.99E-01
Wgt55	2.751E-02	3.376E-02	3.27E-02	2.43E-01
Wgt56	1.250E-01	-4.090E-03	-7.38E-01	9.26E-02
Wgt57	4.374E-01	2.813E+00	7.00E+00	4.51E-01
Wgt58	3.331E-01	-6.535E+00	1.74E-01	6.03E-02
Wgt59	-2.652E+01	1.390E+01	8.20E-03	1.88E-01
Wgt60	1.860E-02	3.852E-03	-6.10E-01	-4.82E-02
Bs1	5.96E+00	-9.14E-03	1.42E+00	1.35E-02
Bs2	1.12E-01	-3.15E-05	4.50E-02	1.61E-01
Bs3	-1.74E-02	1.19E+00	-2.19E+00	-3.40E-01
Bs4	3.62E+00	-4.60E-01	-9.46E-01	-2.06E+00
Bs5	-7.04E-02	6.20E-07	8.88E-01	4.35E-01
Bs6	9.79E-01	6.11E-01	2.68E-03	6.80E-02
Bs7	-9.65E-01	-5.15E+01	-4.41E+00	2.76E+00
Bs8	1.50E+00	1.16E-01	2.35E+00	2.35E+00
Bs9	1.71E-01	5.26E-02	-3.20E-03	-5.82E-01
Bs10	3.53E-04	-3.10E-02	6.08E-01	-1.34E-01

- **Step 4**: Fitness values are calculated using Eq.6. The lowest fitness value is chosen in each iteration to minimize the overall error.

TABLE 4. Efficiency evaluation of OWOA, ABC, WOA and PSO for diabetic classification.

Algorithm Name	Best-Value	Worst-Value	Mean	Median	Standard Deviation	Variance	Computation Time(s)
OWOA	0.195	0.213	0.1994	0.19733	0.00511	0.000026	12.6
ABC	0.204	0.296	0.23069	0.22194	0.02727	0.000743	13.4
WOA	0.268	0.358	0.29954	0.29441	0.02771	0.000768	13.1
PSO	0.282	0.341	0.30805	0.308055	0.01852	0.000343	14.3

- **Step 5**: The WOA is used for the exploration and exploitation phase. The humpback whale follows a reducing helical path to reach the prey each time.
- **Step 6:** The best value obtained for each iteration is updated as the best solution available globally. With the best solution, the distance is calculated to see how close we are to the prey.
- **Step 7**: Based on the jumping value on every iteration, candidate solution and its opposite points values are calculated and kept as the candidate values for the FNN, and the fitness value of each OWOA solution is evaluated using equation (6).
- **Step 8**: All the fitness values are sorted in ascending order using control parameters for all the available whales (agents).
- **Step 9**: To choose twenty-five best fitness values using OWOA and use those best values for next iteration.
- **Step 10**: The weight and bias values are checked for duplicate occurrence and if a duplicate value is found, it will be replaced with a randomized value.
- **Step 11**: To repeat 3rd to 6th step for maximum iteration to train feed forward neural network.
- **Step 12**: When termination criteria is met, iteration is stopped and go to step 4 for next iteration.

Flow chart of the algorithm is shown in figure 2.

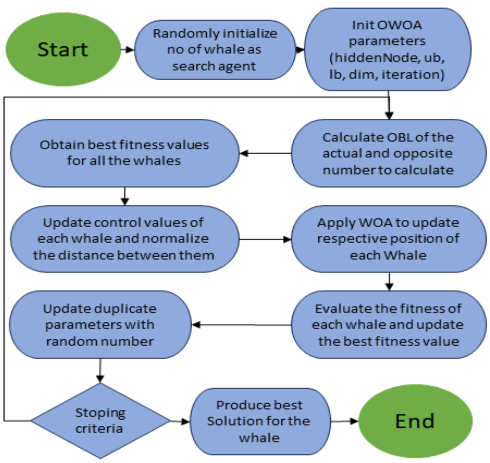


FIGURE 2. Flow chart of the proposed algorithm.

V. EXPERIMENT CONFIGURATION SETUP

In this study, the Pima Indian dataset has been used. The model architecture follows 1 - Y - 1 configuration, with Y = number of hidden nodes. In current setup, Y = 7. The data is used to provide the diagnosis of diabetic occurrence using this dataset. The dataset has 768 total data, 500 are non-diabetic, 268 diabetic cases. The dataset has a total of 8 fields and one output field as diabetic prediction. The research is performed on ladies to evaluate the impact of the given input parameters on diabetes whether it's impacting negatively or positively. The output class is divided into 0 and 1 respectively mean non-diabetic, diabetic. All the optimization algorithms are evaluated in Python-v.3.7.xx versions, 4GB-RAM and 500 Gigs Ram, and Intel Core i52450N CPU. The maximum iteration for the proposed OWOA model is 100, and the total number of candidates is 50.

This dataset is used to train FNN to evaluate the model accuracy. Four standard meta-heuristic algorithms (OWOA, WOA, ABC, and PSO) are used to train a feed forward neural network (FNN) to measure the suitability and supremacy of the proposed OWOA. We have considered the same number of iterations for all the algorithms to check the proposed algorithm's performance unbiasedly.

A. SIMULATION RESULTS AND DISCUSSION

In current setup, we used OWOA to evaluate diabetes detection efficiency with standard dataset. The impact and improvement of the proposed OWOA model are thoroughly examined via a comprehensive comparison with several other models for example OWOA, WOA, ABC and PSO for diabetes detection.

To check the efficiency of the model, Table 3 and Table 5 present a comparison of fitness value. Furthermore, accuracy is measured using another method known as the confusion matrix. This analysis provides valuable insights into the effectiveness of the proposed OWOA technique and its potential advantages over other algorithms for diabetes diagnosis.

The performance evaluation technique used in this study is essential for machine learning classification. With the help of the confusion matrix, various computations for the model can be conducted, including the accuracy of the model.

The accuracy of classification is evaluated using eqn. (18)

$$Accuracy = (X + X^{\sim})/(X + X^{\sim} + A + A^{\sim}) \tag{18}$$

where X (True Positive), $X^{\sim}$ (True Negative), A (False Negative), and $A^{\sim}$ (False Positive).

It provides details about the input-nodes, hidden-nodes, and training data utilized. Notably, the results of TABLE 3 and Table 5 displays the least fitness metric and improved

TABLE 5. Least fitness metric of the proposed algorithms for diabetes dataset.

Control Var.	OWOA	ABC	WOA	PSO
W1	3.14E+01	-6.59E+01	4.24E-02	-7.06E+00
W2	5.95E-01	-4.26E-01	-1.23E+00	9.68E-02
W3	1.96E+00	-3.61E+00	6.47E-02	3.00E-03
W4	-1.90E+00	-1.19E-03	-1.44E-01	-7.23E-01
W5	-3.00E+01	2.37E+00	-2.65E-01	-3.15E-01
W6	-8.01E-02	-4.03E-02	-1.59E+01	-9.07E+00
W7	7.73E-03	2.52E-01	-9.56E+00	6.29E+00
W8	4.40E-01	-1.09E+00	1.49E-01	-3.80E-01
W9	-4.25E+00	-1.74E+00	-4.83E+00	-1.65E+00
W10	-9.91E-05	-3.38E+00	-2.62E+00	7.54E+00
W11	-2.61E+00	8.63E+00	3.03E+00	1.02E+00
W12	-5.04E+00	-3.12E-01	3.05E-01	4.98E+00
W13	-7.68E+00	1.62E+00	-2.12E+01	-1.47E+01
W14	-1.52E-01	-1.30E+00	-5.03E+00	-4.90E+01
W15	1.50E+00	-3.27E+00	-1.68E+00	9.93E-01
W16	-3.50E-01	-1.53E+00	2.06E-02	3.90E+01
W17	6.78E+00	3.13E-01	-4.63E-03	-7.68E-01
W18	2.38E-03	6.85E+00	-1.48E+01	-5.78E+00
W19	2.00E-01	-9.40E+00	1.59E-01	-3.54E+00
W20	-2.91E+00	-4.07E+00	1.78E+00	-2.12E-01
W21	-1.86E-01	-7.40E-02	2.14E-01	-6.02E+00
W22	-1.01E+02	-2.30E+00	-1.31E+00	-2.97E-01
W23	5.18E-01	6.99E+00	-7.12E+00	-1.38E-01
W24	-2.74E+00	-2.18E+00	-1.51E-02	-2.42E+01
W25	-4.15E+01	-2.70E+00	-4.17E+00	-1.27E-02
W26	8.52E-01	4.68E-01	-4.34E-01	1.77E+00
W27	-2.08E+00	5.31E-01	-8.86E-01	-3.68E+00
W28	-7.97E-01	-1.06E+00	-3.97E-03	-8.73E+01
W29	-2.85E+00	-1.59E-01	-2.78E-01	-5.94E-02
W30	-3.03E-01	-2.42E-05	-3.08E+00	2.89E+00
W31	-1.72E+00	-2.19E+00	-2.97E-01	4.03E+00
W32	2.22E-02	5.12E-01	-6.87E+00	-2.28E+00
W33	1.32E-01	-1.31E+01	-3.28E-02	-1.70E+00
W34	3.76E-01	1.68E-01	1.55E-02	-8.71E+00
W35	1.74E-03	-7.82E-01	-3.76E+00	-2.86E+01
W36	-1.66E-03	-7.83E-01	-5.95E+00	-5.49E+00
W37	-3.16E+00	-1.97E+00	1.06E+00	2.44E-01
W38	-3.75E+01	4.89E-01	-4.89E+01	-3.24E+00
W39	4.62E-01	2.30E+00	3.44E+00	-6.00E-01
W40	2.95E+00	-7.08E+00	-2.37E-01	1.02E+00
W41	-5.25E-01	5.27E-01	-2.99E+01	1.95E-01
W42	-1.65E+00	-2.29E-01	5.71E-01	4.87E+00
W43	1.02E-04	2.40E+00	-8.27E+00	-3.87E+00

TABLE 5. *(Continued.)* Least fitness metric of the proposed algorithms for diabetes dataset.

W44	4.59E+01	3.05E-01	-8.86E-01	-4.39E+00
W45	-7.84E+00	1.16E+00	-6.60E-01	2.63E+01
W46	1.80E+01	9.23E-01	4.39E+00	3.52E-01
W47	4.10E-01	1.17E-01	-1.78E+00	-2.49E-01
W48	1.82E-02	-4.75E-01	-5.09E-03	1.79E-01
W49	-3.64E+00	-5.02E+00	-5.98E-01	1.53E+00
W50	-1.28E+02	1.07E+00	-4.67E+00	-1.44E-02
W51	-1.78E+01	9.56E-01	1.14E-01	-4.02E+00
W52	6.93E-01	8.06E+00	2.95E-01	5.19E-01
W53	2.59E-01	1.44E+00	1.02E-02	-2.64E+00
W54	3.24E-02	3.54E-02	-5.86E-01	7.50E-01
W55	-2.51E-01	-7.61E+00	3.76E-01	-5.81E+00
W56	-2.72E+00	-1.55E+00	-1.35E+01	-8.03E+00
W57	-2.46E+00	-9.10E-01	-1.74E+01	-1.08E+02
W58	3.27E+00	-9.96E-01	-1.15E+01	7.09E-01
W59	-2.22E+00	2.27E+00	-4.55E-02	4.12E-03
W60	2.13E+00	-4.09E-01	9.68E-02	-3.78E+00
B1	-2.35E-02	-1.14E-02	5.46E+00	-8.49E-01
B2	-1.74E+01	2.22E+00	-1.88E+00	-1.10E+01
B3	3.03E-02	4.00E-01	-3.89E-01	-1.52E+00
B4	-1.23E-01	-2.89E+00	-1.04E+00	-2.64E-01
B5	-6.77E+00	-8.37E-01	-6.61E+00	-5.24E-01
B6	2.22E+00	9.90E-01	6.98E-02	-2.84E+00
B7	-2.51E-03	2.63E-01	2.47E-01	2.52E-01
B8	2.04E+00	3.06E+00	8.30E-02	-4.73E-01
B9	-5.26E+00	-1.54E+00	1.98E-01	-8.39E-01
B10	-2.14E+02	-1.73E+01	-8.80E+00	-2.14E+00

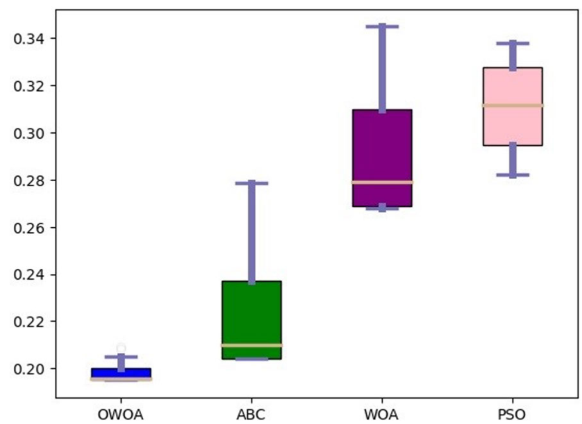


FIGURE 3. Boxplot of the statistical results for OWOA, ABC, WOA and PSO algorithms using Diabetes dataset. The results obtained for the OWOA are clearly better than those obtained for the rest of the tested algorithms.

efficiency which indicate the OWOA algorithm outperforms the other algorithms (OWOA, ABC, WOA and PSO).

Algorithm 3 Algorithm for OWOA

Initialize input parameters of WOA like population size (n_x), number of control variables (n_y), maximum iteration cycles (n_{max}, jumping rate, etc.)
Create initial population of WOA as follows:
FOR *i* in range (1, n_x): do
  FOR *j* in range(1, n_y): do
    $F(i,j) = x_{i,min} + \text{rand} *(x_{i,max}\text{-}x_{i,min})$
  END FORLOOP
END FORLOOP
Generate opposite initial population as follows:
FOR *i* in range (1, n_x): do
  FOR *j* in range(1, n_y): do
    $OP(i,j) = x_{i,min} + (x_{i,max}) + F(i,j)$
  END FORLOOP
END FORLOOP
Evaluate fitness value of current population and opposite population of all whale search agent using (4)
Based of fitness values, (M_p) a number of best solutions are selected.
J = 1
WHILE J < J_{max} do
  FOR *j* in range(1,m_x):do
    Update j^{th} whale particle using two phases namely searching phase and encircling phase.
  END FORLOOP
  Chaotic map is used to generate non-repetitive set of random numbers to cover all possible states in search domain.
  FOR *j* in range(1,m_x):do
    Generate opposite particle of the j^{th} whale particle using non-repetitive chaos based jumping rate.
  END FORLOOP
  Calculate fitness value of current population and opposite population of all whale agent using (4).
  Based on fitness values, M_p number of best solutions are selected from current and opposite population.
  J = J +1
END WHILELOOP
Display optimal results.

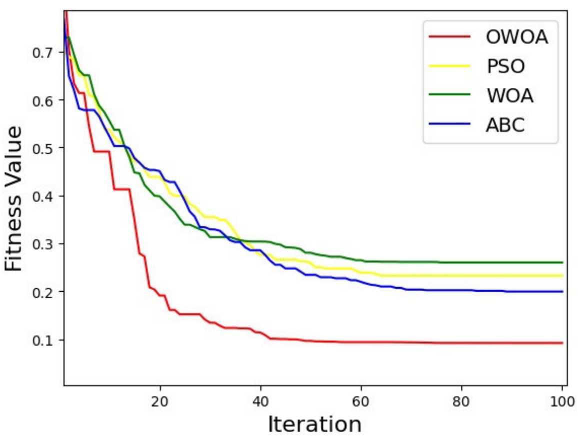


FIGURE 4. Convergence graph of OWOA, WOA, ABC and PSO for diabetes dataset.

To assess the advantage of the suggested OWOA model for improvement, the statistical values of OWOA are compared with ABC, WOA and SSO in Table 4 and Table 6. Figure 3 presents a boxplot of statistical variables like mean, median, standard deviation, best value, and least value. These statistical results clearly demonstrate that the best, mean, median, and worst fitness values obtained by the OWOA method are closer and therefore assures the advantage of the OWOA method. The tuning efficiency of the proposed OWOA has been measured in terms of convergence mobility. The convergence graph of the proposed algorithm in comparison with the other algorithms are shown in the figure 4. This graph shows that the proposed algorithm can more easily escape the local optima than other algorithms.

TABLE 6. Efficiency evaluation of OWOA, ABC, WOA and PSO for diabetic classification.

Statistical Metric	OWOA	ABC	WOA	PSO
Best	0.030418	0.03564	0.043935	0.052911
Average	0.038004	0.045889	0.066539	0.076884
Median	0.033218	0.039183	0.058102	0.059852
Standard Deviation	0.019673	0.017687	0.05699	0.068326

VI. CONCLUSION

In this study, a hybrid OWOA is developed to train the FNN to predict diabetic patients based on some parameters. This study combines the oppositional learning technique to address two significant bottlenecks observed in the original WOA: slow convergence and the tendency to get stuck in local minima. By incorporating OBL, the proposed approach achieves smoother convergence, resulting in an improved convergence rate and enhanced search capabilities. The study applies the OWOA technique to solve a diabetic dataset problem. The results show the efficacy of OWOA as convergence accuracy and ability to find global optima, which makes it a robust algorithm for various scientific and research domains.

REFERENCES

[1] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometric Intell. Lab. Syst.*, vol. 39, no. 1, pp. 43–62, Nov. 1997, doi: 10.1016/S0169-7439(97)00061-0.
[2] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, Oct. 1994, doi: 10.1109/45.329294.
[3] B. Zhang, W. Huang, L. Gong, J. Li, C. Zhao, C. Liu, and D. Huang, "Computer vision detection of defective apples using automatic lightness correction and weighted RVM classifier," *J. Food Eng.*, vol. 146, pp. 143–151, Feb. 2015, doi: 10.1016/j.jfoodeng.2014.08.024.
[4] M. Arora and V. Kansal, "Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis," *Social Netw. Anal. Mining*, vol. 9, no. 1, p. 12, Mar. 2019, doi: 10.1007/s13278-019-0557-y.
[5] J. Zhang, H. Li, Z. Tang, Q. Lu, X. Zheng, and J. Zhou, "An improved quantum-inspired genetic algorithm for image multilevel thresholding segmentation," *Math. Problems Eng.*, vol. 2014, pp. 1–12, Jan. 2014.
[6] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, "A systematic literature review on particle swarm optimization techniques for medical diseases detection," *Comput. Math. Methods Med.*, vol. 2021, pp. 1–10, Sep. 2021, doi: 10.1155/2021/5990999.
[7] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007, doi: 10.1007/s11721-007-0002-0.

[8] W. H. Bangyal, A. Hameed, J. Ahmad, K. Nisar, M. R. Haque, A. A. A. Ibrahim, J. J. P. C. Rodrigues, M. A. Khan, D. B. Rawat, and R. Etengu, "New modified controlled bat algorithm for numerical optimization problem," *Comput., Mater. Continua*, vol. 70, no. 2, pp. 2241–2259, 2022.

[9] K. V. Price, "Differential evolution," in *Handbook of Optimization*. Berlin, Germany: Springer, 2013, pp. 187–214, doi: 10.1007/978-3-642-30504-7_8.

[10] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[11] X.-S. Yang and S. Deb, "Cuckoo search: Recent advances and applications," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 169–174, Jan. 2014, doi: 10.1007/s00521-013-1367-1.

[12] D. Karaboga, "Artificial bee colony algorithm," *Scholarpedia*, vol. 5, no. 3, p. 6915, Mar. 2010, doi: 10.4249/scholarpedia.6915.

[13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[14] Y.-R. Zeng, Y. Zeng, B. Choi, and L. Wang, "Multifactor-influenced energy consumption forecasting using enhanced back-propagation neural network," *Energy*, vol. 127, pp. 381–396, May 2017, doi: 10.1016/j.energy.2017.03.094.

[15] P. Hosseinioun, M. Kheirabadi, S. R. K. Tabbakh, and R. Ghaemi, "A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm," *J. Parallel Distrib. Comput.*, vol. 143, pp. 88–96, Sep. 2020, doi: 10.1016/j.jpdc.2020.04.008.

[16] N. Manikandan, N. Gobalakrishnan, and K. Pradeep, "Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment," *Comput. Commun.*, vol. 187, pp. 35–44, Apr. 2022, doi: 10.1016/j.comcom.2022.01.016.

[17] S. Malik and D. Kim, "Prediction-learning algorithm for efficient energy consumption in smart buildings based on particle regeneration and velocity boost in particle swarm optimization neural networks," *Energies*, vol. 11, no. 5, p. 1289, May 2018, doi: 10.3390/en11051289.

[18] K. Li, Z. Ma, P. Fu, and S. Krishnaswamy, "Quantitative evaluation of surface crack depth with a scanning laser source based on particle swarm optimization-neural network," *NDT & E Int.*, vol. 98, pp. 208–214, Sep. 2018, doi: 10.1016/j.ndteint.2018.05.011.

[19] Y. Xu, C. Hu, Q. Wu, S. Jian, Z. Li, Y. Chen, G. Zhang, Z. Zhang, and S. Wang, "Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation," *J. Hydrol.*, vol. 608, May 2022, Art. no. 127553, doi: 10.1016/j.jhydrol.2022.127553.

[20] M. Ilbeigi, M. Ghomeishi, and A. Dehghanbanadaki, "Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm," *Sustain. Cities Soc.*, vol. 61, Oct. 2020, Art. no. 102325, doi: 10.1016/j.scs.2020.102325.

[21] J. Wang, J. Bei, H. Song, H. Zhang, and P. Zhang, "A whale optimization algorithm with combined mutation and removing similarity for global optimization and multilevel thresholding image segmentation," *Appl. Soft Comput.*, vol. 137, Apr. 2023, Art. no. 110130, doi: 10.1016/j.asoc.2023.110130.

[22] M. Li, X. Yu, B. Fu, and X. Wang, "A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems," *Neural Comput. Appl.*, vol. 2023, pp. 1–14, Jan. 2023, doi: 10.1007/s00521-023-08287-5.

[23] P. Kaladevi, V. V. Punitha, D. Muthusankar, and R. Praveen, "Breast cancer diagnosis using orca predation optimization algorithm," *J. Intell. Fuzzy Syst.*, vol. 45, no. 3, pp. 3855–3873, Aug. 2023, doi: 10.3233/jifs-231176.

[24] R. A. Musheer, C. K. Verma, and N. Srivastava, "Novel machine learning approach for classification of high-dimensional microarray data," *Soft Comput.*, vol. 23, no. 24, pp. 13409–13421, Dec. 2019, doi: 10.1007/s00500-019-03879-7.

[25] R. Aziz, C. K. Verma, M. Jha, and N. Srivastava, "Artificial neural network classification of microarray data using new hybrid gene selection method," *Int. J. Data Mining Bioinf.*, vol. 17, no. 1, p. 42, 2017.

[26] R. Aziz, C. K. Verma, and N. Srivastava, "Artificial neural network classification of high dimensional data with novel optimization approach of dimension reduction," *Ann. Data Sci.*, vol. 5, no. 4, pp. 615–635, Dec. 2018, doi: 10.1007/s40745-018-0155-2.

[27] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.

[28] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, "A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models," *Energy Convers. Manage.*, vol. 230, Feb. 2021, Art. no. 113784, doi: 10.1016/j.enconman.2020.113784.

[29] K. Lenin, "Real power loss reduction by duponchelia fovealis optimization and enriched squirrel search optimization algorithms," *Soft Comput.*, vol. 24, no. 23, pp. 17863–17873, Dec. 2020, doi: 10.1007/s00500-020-05036-x.

[30] M. S. Turgut, H. M. Sağban, O. E. Turgut, and Ö. T. Özmen, "Whale optimization and sine–cosine optimization algorithms with cellular topology for parameter identification of chaotic systems and Schottky barrier diode models," *Soft Comput.*, vol. 25, no. 2, pp. 1365–1409, Jan. 2021, doi: 10.1007/s00500-020-05227-6.

[31] D. B. Prakash and C. Lakshminarayana, "Optimal siting of capacitors in radial distribution network using whale optimization algorithm," *Alexandria Eng. J.*, vol. 56, no. 4, pp. 499–509, Dec. 2017, doi: 10.1016/j.aej.2016.10.002.

[32] L. Haghnegahdar and Y. Wang, "A whale optimization algorithm-trained artificial neural network for smart grid cyber intrusion detection," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9427–9441, Jul. 2020, doi: 10.1007/s00521-019-04453-w.

[33] H. Guo, J. Zhou, M. Koopialipoor, D. Jahed Armaghani, and M. M. Tahir, "Deep neural network and whale optimization algorithm to assess flyrock induced by blasting," *Eng. Comput.*, vol. 37, no. 1, pp. 173–186, Jan. 2021, doi: 10.1007/s00366-019-00816-y.

[34] S. H. Yeung and K. F. Man, "Multiobjective optimization," *IEEE Microw. Mag.*, vol. 12, no. 6, pp. 120–133, Oct. 2011, doi: 10.1109/MMM.2011.942013.

[35] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014, doi: 10.1007/s10462-012-9328-0.

[36] S. Ghosh and D. Chatterjee, "Artificial bee colony optimization based non-intrusive appliances load monitoring technique in a smart home," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 77–86, Feb. 2021, doi: 10.1109/TCE.2021.3051164.

[37] P. Stephan, T. Stephan, R. Kannan, and A. Abraham, "A hybrid artificial bee colony with whale optimization algorithm for improved breast cancer diagnosis," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13667–13691, Oct. 2021, doi: 10.1007/s00521-021-05997-6.

[38] S. Mahdavi, S. Rahnamayan, and K. Deb, "Opposition based learning: A literature review," *Swarm Evol. Comput.*, vol. 39, pp. 1–23, Apr. 2018, doi: 10.1016/j.swevo.2017.09.010.

RAJESH CHATTERJEE received the M.C.A. degree from Jawaharlal Nehru University. He is currently pursuing the Ph.D. degree with Usha Martin University, Ranchi, Jharkhand, India. He is currently an Assistant Vice President in an international bank. He has published extensively and notably in the field of data science and machine intelligence. His research interests include mobile meta heuristic algorithm, natural language processing, machine learning, deep learning, and computer vision.

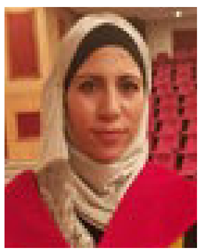
MOHAMMAD AMIR KHUSRU AKHTAR received the M.Tech. and Ph.D. degrees from the Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India, in 2009 and 2015, respectively. He is currently an Associate Professor with the Faculty of Computing and Information Technology, Usha Martin University, Ranchi. He is the author of over 20 peer-reviewed publications. His research interests include mobile ad-hoc networks, natural language processing, machine learning, deep learning, metaverse, and feature engineering.

DINESH KUMAR PRADHAN received the Ph.D. degree in computer science and engineering from NIT Durgapur. He is currently an Assistant Professor in information technology with the Dr. B. C. Roy Engineering College, Durgapur, India, where he has been a Faculty Member, since 2009. He has published extensively and notably in the field of data science and machine intelligence. Outside of professional interests, he travels widely, reads, writes, sails, and believes in happy living.

FALGUNI CHAKRABORTY received the M.Tech. degree in operations research from NIT Durgapur, West Bengal, India, in 2014, and the Ph.D. degree from the Department of Computer Science and Engineering, NIT Durgapur, in 2021. He is currently an Assistant Professor with the Master of Computer Application Department, Dr. B. C. Roy Engineering College, Durgapur, West Bengal. He has 14 years of experience in academics and three years IT industry. He is the author of over ten peer-reviewed publications in reputed journals like *Applied Soft Computing* (Elsevier) and *Soft Computing* (Springer). His research interests include computer vision, evolutionary algorithms, AI & machine learning, and deep learning.

MOHIT KUMAR received the Ph.D. degree from Jaipur National University. He has published articles in reputed top-cited journals like IEEE Transactions on Industrial Informatics, IEEE Transactions on Network Science and Engineering, *HCIS* (Springer), *Journal of Information Science and Engineering*, *Sensors* (MDPI), and *Symmetry*. He also has been granted two patents. He is currently guiding four Ph.D. students. He has also worked as a consultant in government funded projects. His research interests include WSN, the IoT, optimization, ML, AI, DL, and data science. He is a reviewer of top-cited journals.

SAHIL VERMA (Senior Member, IEEE) received the B.Tech. and M.Tech. degrees in computer science and engineering from Maharishi Markandeshwar University, Mullana, Ambala, India, in 2012, and the Ph.D. degree in computer science and engineering from Jaipur National University, Jaipur, India, in 2017. He is currently with the Department of Computer Science and Engineering, Chandigarh Group of Colleges, Chandigarh University, Mohali, India. He has authored or coauthored articles in reputed top-cited journals, including the IEEE Transactions on Industrial Informatics, IEEE Internet of Things Journal, *ACM Transactions on Internet Technology*, and IEEE Access. He is a Reviewer of top-cited journals, including the IEEE Transactions on Intelligent Transport Systems, IEEE Transactions on Network Science and Engineering, IEEE Access, *Neural Computing and Applications* (Springer), *Human-Centric Computing and Information Sciences* (Springer), *Mobile Networks and Applications* (Springer), *Journal of Information Security and Applications* (Elsevier), *Mobile Information Systems* (Hindawi), *International Journal of Communication Systems* (Wiley), and *Security and Communication Networks* (Hindawi). He is a Professional Member of ACM and IAENG. He has many research contributions in the area of cloud computing, the Internet of Things, vehicular ad-hoc networks, WSN, and MANET. Some of his research findings are published with top-cited publishers, including IEEE, Wiley, Springer, ACM, Elsevier, MDPI, and various international conferences of repute. He has chaired many sessions in international conferences. He is also a member of editorial board of many journals, a TPC member, and an organizing committee member in various international conferences. He has also participated as an author in various international conferences. He is involved in many research and development activities. He has visited two international universities as VSB, Technical University of Ostrava, Ostrava, Czech Republic; and University of Milan, Milan, Italy, for research collaborations. He has guided 14 PG students, three Ph.D. students, and six are ongoing.



RUBA ABU KHURMA received the bachelor's and master's degrees (Hons.) in computer science from Yarmouk University, in 2004 and 2007, respectively, and the Ph.D. degree (Hons.) in computer science from the University of Jordan, in 2021. She was an Assistant Professor with the Computer Science Department, Al-Ahliyya Amman University. She was a Researcher in cybersecurity and machine learning with the Department of Cybersecurity and Digital Forensics, Al-Balqa Applied University. She has published many articles in reputable journals, book chapters, and international conferences, such as CEC, ICPRAM, ECTA, ITISE, PICIT, and Evo-applications.

MARIBEL GARCÍA-ARENAS received the Ph.D. degree in computer science from the University of Granada, in 2003. From 2003 to 2007, she was with the University of Jaén, teaching computer and telecommunication engineering. She has been with the University of Granada, since 2007, where she is currently a Senior Lecturer with the Architecture and Technology of Computers Department, teaching in computer science. Her main research interests include evolutionary computation and high-performance computing. She has worked on several European, national, regional projects being the head researcher for three projects, and two of them related to mobility and traffic at national and regional level. She also has more than 25 publications in international journals and a higher number in conferences.