

RESEARCH ARTICLE

Cervical Cancer Diagnosis Using Intelligent Living Behavior of Artificial Jellyfish Optimized With Artificial Neural Network

DEVIKANNIGA DEVARAJAN¹, D. STALIN ALEX², T. R. MAHESH¹, V. VINOTH KUMAR³, RAJANIKANTH ALUVALU⁴, (Senior Member, IEEE), V. UMA MAHESWARI⁵, (Member, IEEE), AND S. SHITHARTH⁶, (Senior Member, IEEE)

¹Department of Computer Science and Engineering, JAIN (Deemed-to-be University), Bengaluru 562112, India

²Department of Computer Science and Engineering (Data Science), State University of Bangladesh, Dhaka 1205, Bangladesh

³School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu 632014, India

⁴Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad 500075, India

⁵Department of Computer Science and Engineering, KG Reddy College of Engineering and Technology, Hyderabad 500075, India

⁶Department of Computer Science and Engineering, Kebri Dehar University, Somali, Kebri Dehar 3060, Ethiopia

Corresponding author: S. Shitharth (shitharth.s@ieee.org)

ABSTRACT Cervical cancer affects nearly 4% of the women across the globe and leads to mortality if not treated in early stage. A few decades before, the mortality rate was too high when compared to the present statistics. This is achieved as nowadays most of women are aware of this disease and undergo health examination mainly for screening cervical cancer on regular basis. But only the accurate diagnosis can be helpful for further treatment. Many works are carried out for accurate diagnosis and always have some limitations in accurate predictions. In this work, an efficient algorithm is proposed for the accurate diagnosis of cervical cancer. A meta-heuristic called artificial Jellyfish search optimizer (JS) algorithm is combined with artificial neural network (ANN) to tackle this problem. The proposed algorithm is called JellyfishSearch_ANN and is employed to classify the cervical cancer dataset with four type of targets based on the examination. The JellyfishSearch_ANN provides outstanding results among other classifiers taken for comparison and mainly its classification accuracy is found to be above 98.87% for all targets.

INDEX TERMS Artificial neural network, artificial jellyfish search optimizer, cervical cancer, metaheuristic optimization.

I. INTRODUCTION

One among the gynecological cancer is the cervical cancer which occurs in the reproductive organ of the women. Its different sorts are ovarian cancer, uterine cancers, vaginal cancers and others. The lower part of uterus that ends with vagina is the cervix. When the DNA mutations takes place in the healthy cells of cervix, this mostly ends with cancer in cervix. There is a possibility of spreading cancer to various organs that are nearby such as vagina, lungs, liver and others. Mostly middle aged women are highly prone to this disease. It is of various types namely adenocarcinoma, squamous cell carcinoma, and adenosquamous carcinoma. These may

The associate editor coordinating the review of this manuscript and approving it for publication was Shuihua Wang¹.

spread in four different stages as stage1, stage2, stage3 and stage4. This disease is cause due to the sexual transmission of human papillomavirus(HPV) from an infected person to a healthy [1]. As the symptoms are hard to diagnose at the early stage, hence most of the women become a victim for this disease.

Every female should proceed with a normal checkup at a right period of time but the symptoms are hard to analyse. This may be averted by a means of PAP smear screening technique and with the help of getting HPV vaccine. This cancer may be prevented by a correct analysis with proper assistance. This is one of the preventable and treatable if identified early and handled effectively [2], [3]. The proposed approach, JellyfishSearch_ANN handles this purpose.

A number of works carried out on cervical cancer are mentioned below. In [4], the strategies which include filters and wrappers have been used to choose the attributes from the dataset utilized in that work. The lacking and imbalance information are dealt with the use of oversampling, under sampling and other sampling strategies. Among the following classifiers like logistic regression, support vector machine (SVM), neural networks (NN), the Decision tree classifier has yielded 97.5%. The attributes which include patient_age, smoking, hormonal_contraception, first_sexual_intercourse, pregnancy_numbers, and STDs are considered as relevant features [4]. The PAP_smear images of three datasets are classified with fuzzy-cmeans algorithm. The simulated annealing collectively with wrapper filter is applied for feature selection. The maximum accuracy produced for single_cell image dataset is 98.88% [5].

The SVM is utilized in [6], to find the existing condition or stage of the person affected with cervical cancer, images of Magnetic resonance were used. An ensemble classifier [7] is developed by combining the techniques such as linear discriminant, support vector machine, K-nearest neighbor, Boosted and bagging trees. This was considered to be a computer assisted screening system and it was applied for SIPakMed dataset. It is found that it have furnished an accuracy of 98.27% on classifying unhealthy cells from healthy cells. The Long-short-time memory Recurrent neural network (RNN) is optimized using meta-heuristic techniques which include cuckoo_search, genetic algorithm, gravitational search, particle_swarm optimization and gray wolf optimizer. The test is carried out for 668 times and 34 attributes of dataset [11] after preprocessing and a maximum of 96% accuracy is found to be achieved [8].

The deep convolution neural network(CNN) approach is employed classifying the 4 targets of the cervical cancer dataset and it is inferred that the accuracy above 90% is obtained for all the targets [9]. The Gauss Newton representation-based formula (GNRBA) [10] is employed for classifying the cervical cancer data with four targets. This approach has produced 95.35% of accuracy for Hinselmann target variable.

The transfer learning models such as VGG-19, ResNet-50, InceptionV3, AlexNet and SqueezeNet were used in [12] to detect the non-cancerous and cancerous cells in cervix. The preprocessed Pap smear images were used to analyze the structural completeness of these cervical cells. Among all these pretrained models, SqueezeNet has produced the best validation accuracy as 96.9%. In [13], the ResNet-50 model was employed to recognize the Pap smear images to classify the cervical cells as cancerous cells with three classes (superficial squamous, columnar epithelial and intermediate squamous) and non-cancerous cells with four classes (mild, moderate, severe dysplasia and carcinoma -in - situ), using the Pap smear images. In this work, the ResNet has yielded 74.04% accuracy when compared with its competitor CNN whose accuracy was only 44%.

The methods such as Random forest, Decision tree, Logistic regression and Ada boost were applied to the dataset [11], also by including a weighted version [14]. The performance of each model is assessed in terms of both weighted and unweighted version of the dataset. The maximum accuracy observed from this work is below 86%. The Random forest classifier in a combination with a recursive feature elimination technique was applied to the dataset [11] in this work [15]. Along with this, other classifiers such as Naïve Bayes, KNN, SVM and Decision trees were also used. The Performance is accessed using cross validation techniques and it is obtained that Random forest algorithm has produced a highest of 93% accuracy than all others.

The cervical cancer classification was conducted using a total of 915 images with four classes namely normal, adenocarcinoma, precancer and squamous cell cancer. These images were preprocessed and classified using the EfficientNetBD, a pretrained model, which was found to be 94.5% accurate in testing [16]. The dataset was built by using the measurements from the optical measurements in [17]. The machine learning algorithms such as eXtreme Gradient Boosting, Naïve Bayes, Random forest and Convolution neural network were used along with optoelectronic sensor to predict cervical cancer at the early stage. For this optically derived dataset, the accuracy yielded by these machine learning algorithms were observed to produce about 95%.

From the literature, it is found that a few works are carried out so far on the diagnosis of cervical cancer. Over all, it is noted that there are few limitations in achieving the better accuracy and have not focused on lesser computation time. To focus on both, that is to increase the cervical cancer classification accuracy and decrease the computation time, a classifier namely JellyfishSearch_ANN (JSA-ANN) is proposed. The ANN algorithm is firstly combined with it and applied for the discrimination of cervical cancer samples from the normal ones. The Jellyfish optimizer is chosen for this work based on its simpler calculation and quicker response than the other optimizers. The efficiency of this algorithm is compared with that of other similar hybrids and the details are discussed in the following sections.

This paper is structured as follows: a few related works in the literature are discussed above in the introduction section. Section 2 describes about the materials and methods deployed in this work. The experimental settings are elaborated in section 3. The analysis of the results of the proposed work and comparison algorithms used in this work are discussed in section4. The section 5 finally concludes the work.

II. MATERIALS AND METHODS

The details of the algorithms used in this proposed work are discussed in this section. The working of ANN classifier, the concepts behind JSA and the procedure of fusing JSA with ANN are elaborated below.

A. THE ANN CLASSIFIER

The ANN is a system that has parallelly distributed process. The architecture is known to have a huge number of less complex densely connected processors. The multi-layer perceptron architecture is widely used for at most of the major applications in the current era. Figure 1 displays the multilayer feed-forward network with three layers. This architecture shows the unidirectional connection of each neuron with the all other neurons of its adjacent layer strictly without forming a loop. That is, the input information is propagated in one direction that is “forward direction”. Each of the connection bears a weight that are adjusted accordingly based on the learning rule, at the time of backward error propagation [18], [19].

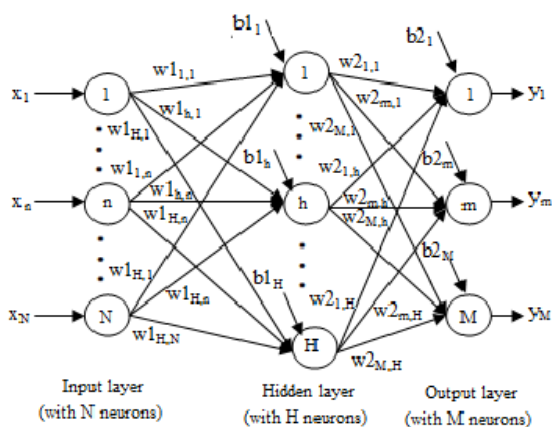


FIGURE 1. Architecture of artificial neural network.

Mostly, ANNs are trained dataset using a gradient descent optimizer called as back propagation algorithm (BP), which is gradient descent optimizer to minimize the factors with respect to error model that are generated during each training phases. During the training procedure, a non-linear mapping of the input variables to the output variables is attained. Thus, the non-linear relationship these variables are recognized by adjusting the weights using BP [20]. The steps for training ANN using BPN for classification is given as follows,

- 1: Load the training data, TD = {X, T} with total number of S samples. Where X={X₁,X₂,...X_S} is the source input and each X_i is N-dimensional data such that X_i={x₁,x₂, ... x_N}. T={T₁,T₂,... T_S} is target output. Each T_i ∈ {0,1} for binary classification dataset.
- 2: Synaptic weights and biases are initialized randomly At each iteration j, j=1,2... S
- 3: Calculate hidden layer
- 3.1: Net input at hidden layer: This is calculated for each hidden neuron h by using (1) as,

$$net_h^j = \sum_{n=1}^N x_n w_{1hn} + b_{1h}, h = 1 \dots H \quad (1)$$

where w_{1hm} synaptic weight of hidden neuron h and input neuron n. N and H are respectively the size of

input layer and hidden layer. b_{1h} bias of hidden neuron h.

- 3.2: Evaluation of Output at Hidden layer Activation function f is calculated from each hidden neuron h from (2) as,

$$O_h^j = f \left(net_h^j \right) \quad (2)$$

In case of sigmoid function, it is calculated as in (3)

$$O_h^j = sigmoid \left(net_h^j \right) = \frac{1}{1 + e^{(-net_h^j)}} \quad (3)$$

- 4: Output layer Computation

- 4.1: Net input at output layer: This is calculated for every output neuron m is calculated using (4),

$$net_m^j = \sum_{h=1}^H O_h^j w_{2mh} + b_{2m}, m = 1 \dots M, \quad (4)$$

where w_{2mh} synaptic weight of hidden neuron h and input neuron n. b_{2m} bias of output neuron m. M – number of output nodes.

- 4.2: Actual output calculation The network output given by each output neuron m is calculated using (5) as,

$$y_m^j = \left(net_m^j \right) \quad (5)$$

- 5: Error value estimation:

The subtraction of Network output from Target output is the error value obtained using (6) as,

$$E = \sum_{m=1}^M \left(T_m^j - y_m^j \right)^2 \quad (6)$$

T_m^j and y_m^j are the target and network output respectively from jth training data.

- 6: Weight Adjustment: Minimize the error and improve accuracy.
- 6.1: weight update: Equation (7) gives the synaptic weight modification from output layer to layer as,

$$W_{2mh}^{j+1} (new) = w_{2mh}^j + \Delta w_{2mh}^j \quad (7)$$

where Δw_{2mh}^j = ηδ_m^jO_h^j weight between output layer (m) neurons and hidden(h) layer neuron, learning rate (η) constant and sensitivity (δ_m^j) of the output (m) node.

- 6.2: weight update: Equation (8) gives the synaptic weight modification from hidden layer to output layer as,

$$w_{1hn}^{j+1} (new) = w_{1hn}^j + \Delta w_{1hn}^j \quad (8)$$

where Δw_{1hn}^j = ηδ_h^jx_n^j weight between input layer (n) neurons and hidden (h) layer neuron. Sensitivity (δ_h^j) of the hidden (h) node.

- 7: The steps from step 2 to step 6 are iterated for all training data till minimization of error takes place and it indicates training process completion.

From step 1 to step 4, the signals flow in the forward direction, after errors are calculated at step 5 and are propagated in backward path for weight adaptation. The set of weights biases that exists after the end of each training process is used to discriminate any unseen data. This final model is achieved based on the minimization of the error which is calculated at every iteration. The BP algorithm is mostly easily prone to trap in local minima and slow convergence. Reference [21], [22]. The metaheuristic algorithms are used to train ANN to overcome these problems.

B. THE JSA CONCEPTS

The Jellyfish search optimizer is a recently developed algorithm by Chou and Truong in 2020 [23]. This approach is devised based on the inspiration of artificial jellyfish movement and its behavior in ocean current. The interesting facts and the structure of model of the algorithm is briefly discussed in this section to express its contribution to this work. The jellyfish adapts itself for any scales of temperature and various shapes, sizes and colours in this world. Their preying behavior also differs. Their bottom part is like umbrella which helps this movement by pushing the water outward in order to move forward. The mass of the jellyfish is also represented as the Jellyfish bloom. This swarm formation happens when there are favorable situations like availability of oxygen, food and temperature.

Three idealized rules defined for this algorithm are:

- The movement of the jellyfish is either based on following the ocean current or jellyfish swarm. The time control mechanism governs these movements.
- The jellyfish move towards abundant food in inside ocean current.
- The location of the jellyfish and the corresponding objective function determines the quality of food.

The jellyfish population (JF) is given as,

$$JF = \{JF_1, JF_2, \dots, JF_{NP}\} \tag{9}$$

where NP is the jellyfish population size. Each jellyfish is given as a D-dimensional vector as follows,

$$JF_i = (if_{i,1}, if_{i,2}, \dots, if_{i,D}) \tag{10}$$

Each element $if_{i,d}$ is the d^{th} dimension of i^{th} jellyfish where $d=1,2, \dots, D$. The population of the jellyfish is not initialized with random values in order to avoid slow convergence. The simplest chaotic maps such as logistic map are used for initializing the population. It is given as,

$$JF_{i+1} = \omega JF_i (1 - JF_i), 0 < JF_0 < 1 \tag{11}$$

where ω : 4.0, JF_i : location of i^{th} jellyfish based on logistic chaotic value and JF_0 : jellyfish, $JF_0 \in [0, 1]$ of initial population.

The jellyfish attracts to the abundant nutrients available in ocean current. The ocean current's direction is given by (12) as,

$$\vec{O}_{dir} = JF^* - A_c \frac{\sum JF_i}{NP}$$

$$\vec{O}_{dir} = JF^* - rand(0, 1) * \gamma * \mu \tag{12}$$

where JF^* : the jellyfish with current best, A_c : attraction governing factor, μ : mean location of jellyfish population, γ : distribution coefficient > 0 and experimentally found to be equal to 3.

The new location of i^{th} jellyfish at time $t+1$ is given in (13).

$$\begin{aligned} JF_i^{t+1} &= JF_i^t + rand(0, 1) * \vec{O}_{dir} \\ JF_i^{t+1} &= JF_i^t + rand(0, 1) * (JF^* - rand(0, 1) * \gamma * \mu) \end{aligned} \tag{13}$$

The jellyfish performs two motions namely passive motion or Type A motion and active motion or TypeB. Initially they start with passive motion, and then over the time it is updated with active motion. The passive motion is based on its own location and the updated location at time $t+1$ is given as (14).

$$JF_i^{t+1} = JF_i^t + \alpha * rand(0, 1) * (UBSS - LBSS) \tag{14}$$

where UBSS : upper bound search space, LBSS: lower bound search space and α : motion coefficient > 0 which is experimentally found as 0.1. The TypeB motion is based on the quantity of food that is found by jellyfish. The food quantity is decided by the objective function, f calculated using the corresponding jellyfish. A jellyfish JF_j is randomly selected to find the direction of movement of the jellyfish_of_interest JF_{int} . If the better food is found at the location of jellyfish JF_j , then the jellyfish JF_{int} moves towards it, else it moves away. This movement of jellyfish to better direction helps for effective exploitation of local search space. The new location of JF_{int} is given in (15).

$$JF_{int}^{t+1} = JF_{int}^t + \vec{M} \tag{15}$$

where

$$\vec{M} = rand(0, 1) * JF_{dir}^{\vec{}} \tag{16}$$

The jellyfish direction,

$$JF_{dir}^{\vec{}} = \begin{cases} JF_j^t - JF_{int}^t, & \text{if } (f(JF_{int}) \geq f(JF_j)) \\ JF_{int}^t - JF_j^t, & \text{otherwise} \end{cases} \tag{17}$$

The time control mechanism helps the jellyfish to decide its type of motion in the jellyfish swarm over the time and also its movement towards an ocean current. This mechanism includes a time control function $t_c \in [0..1]$ and a constant t_0 .

$$t_c = \left| \left(1 - \frac{t}{iter_{max}} \right) * (2 * rand(0, 1) - 1) \right| \tag{18}$$

where t and $iter_{max}$ are respectively the iteration number and maximum number of iterations. If $t_c > t_0$, the jellyfish follows the ocean current, else they move inside the swarm. Then the function $(1 - t_c)$ determine TypeA or TypeB, that is its movement inside the swarm. The $rand(0,1) > (1 - t_c)$, then TypeA motion else TypeB motion. The boundary conditions are calculated and checked for the location update of the jellyfish based on the earth demographics.

TRAINING**Algorithm 1 JSA_ANN()**

Input: Train the Dataset $D = \{D_1, D_2, \dots, D_S\}$, with size (S)
 The training data ($D_{tr} = \{D_1, D_2, \dots, D_{Tr}\}$) and testing data ($D_{ts} = \{D_1, D_2, \dots, D_{Ts}\}$) of size Tr and Ts are respectively taken.
 The size of input layer (N), hidden layer ($H = (2 * N) + 1$) and output layer (M) are considered for building ANN.

Output: JSA-ANN trained classifier model

Algorithm 1.a JSA_Algorithm

Input: Algorithm initialized using Logistic map, Mean square error (fitness of jellyfish "jellyfish_fitness"), these are the values retrieved as output from ANN_A

Output: Selection of Best Jellyfish

```

1: Initializing the algorithm:
1.1: NP is the population size of jellyfish. Every jellyfish is considered
    as set of biases and synaptic weights. Dimension of each jellyfish,
    Dim = (N * H) + (H * M) + H + M, Distribution coefficient ( $\gamma$ ) = 3,
    Motion coefficient ( $\alpha$ ) = 0.1 and maximum iteration (iter_max)
1.2: Initialize the population, Jellyfish swarm (JF) using Logistic map.
1.3: Calculate the fitness value of Jellyfish swarm using ANN_A()
1.4: The food quantity available at every  $i^{\text{th}}$  jellyfish location is calculated
    (objective value) using  $f(JF_i)$ .
1.5: The jellyfish with current best food is found (JF*)
1.6: Time, t is initialized with 1
    2: for generation = 1 to iter_max
    3: for  $i = 1$  to NP
3.1: time control function  $t_c$  is calculated using (18)
3.2: if ( $t_c \geq 0.50$ )
3.3:   The ocean_current direction calculated using (12)
3.4:   New location of jellyfish i is determined by (13)
3.5: else
    The jellyfish i moves inside the jellyfish swarm
3.6: if ( $\text{random}(0, 1) > (1 - t_c)$ )
    The jellyfish will exhibit a Passive motion (TypeA)
3.7: New location of jellyfish i is calculated using (14)
3.8: else
    The jellyfish exhibits the Active motion (TypeB)
3.9: The direction of jellyfish i is evaluated using (16)
3.10: New location of jellyfish i is calculated using (15)
3.11: end if
3.12: end if
3.13: The boundary conditions are checked
3.14: Food quantity at new location is estimated
3.15: Location of jellyfish i,  $JF_i$  is updated
3.16: The jellyfish with current best food is updated (JF*)
3.17: end for
    4: Increment t,  $t = t + 1$ 
    5: Check for stopping criteria,  $t > \text{iter\_max}$ 
    6: end for
    7: Display the training MSE, best fitness value and computation time
  
```

C. CLASSIFIER JELLYFISHSEARCH_ANN (JSA_ANN)

The optimization algorithms are used in the neural networks for various tasks such as feature selection, tuning of hyper parameters, optimizing the synaptic weights and biases, and many others. In this work, the JSA algorithm is employed for optimizing the synaptic weights and the bias values of ANN. This is clearly defined in Algorithm 1 (JSA_ANN) based on training and testing of the algorithm.

Algorithm 1.b ANN_A()

Input:

Load training Initialise training data D_{tr} . Every sample $D_i = \{X_i, T_i\}$, $i = 1, 2, \dots, Tr$.

The training input data, $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\}$ has n features and the training target data, T_i mentions the labels as either 0 (healthy) or 1 (cervical cancer). Weights and bias values are fetched from every iteration of JSA_algorithm.

Output: Training performance, best solution MSE_training and computation (Ct) time.

```

Jellyfish_fitness = ANN_A(H, JF, n, D_tr, M, NP)
1: for i = 1 to NP
2:   for j = 1 to Tr
3:     compute the hidden layer output with (2)
4:     compute the predicted output with (5)
5:     calculate  $E(j) = (T(j) - y(j)) + E(j-1)$ 
6:   end for j
7: end for i
8: calculate  $MSE = \frac{1}{N} \sqrt{E^2}$ 
9: Jellyfish_fitness = MSE
10: Return (Jellyfish_fitness)
11: Report training computation time and training MSE
  
```

TESTING

```

1: Load testing data  $D_{ts}$  and the best_solution.
2: Evaluate the network output  $y_{test}$ 
3: Compare target output and network output
4: Evaluate classifier's performance in terms of Misclassification rate (MR), Accuracy (Acc),  $G_{mean}$ , F-score, specificity, precision, sensitivity.
5: Return JSA-ANN classifier's performance.
  
```

D. THE JSA-ANN CLASSIFIER METHODOLOGY

The ANN is loaded with the training data and the jellyfish population is initialized with Logistic maps. Also, the hyper-parameters like attraction factor, distribution and motion coefficient are also initialized. The fitness (MSE) of every jellyfish is calculated by ANN and also each jellyfish in the population is a vector that is considered as a set of synaptic weights and biases. Hence every jellyfish is used to optimize ANN. The fitness is calculated for all jellyfish through a fixed number of iterations. The Framework of JSA-ANN classifier is shown in figure 2.

The jellyfish movement towards a location is based on the availability of food quantity. Its movement is calculated based on time control function that is either along with ocean current or into the jellyfish swarm. Then the location of each jellyfish is updated. These steps contribute for one iteration and the process is iterated to maximum. It is clear that the JSA generates ANN's synaptic weights and biases in the forward direction and in reverse direction, the MSE calculated by ANN is returned as fitness value of jellyfish to JSA. The optimal solution is best jellyfish calculated from the entire

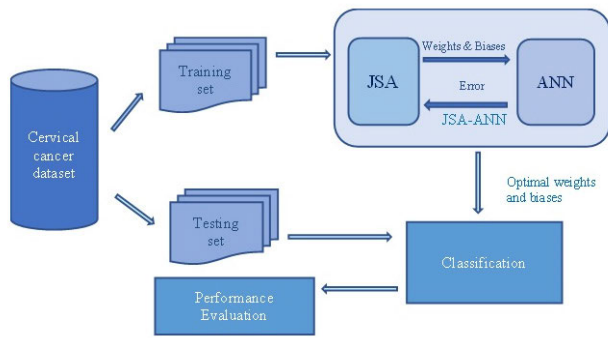


FIGURE 2. Framework of JSA-ANN classifier.

process. This jellyfish is deployed with ANN to test the unpracticed data. The JSA uses logistic map for population initialization to avoid random population. This helps JSA to escape from slow convergence and avoid getting trap in local minima. When compared with other algorithms, the JSA follows very simpler calculations.

It is found that the computational complexity of proposed JSA-ANN classifier with population size, n is found to be $O(n)$.

III. EXPERIMENTAL SETTING

All the algorithms used in this work are built with the MATLAB 2020a. This is installed in the system with the configuration as follows, processor (Intel Core i5) with 2.7GHz speed and RAM (8GB). The randomness in the results is handled by conducting the experiments with 10-fold cross validation with 10 independent trials. The average of the output results is used for discussion. For comparison of JSA-ANN's performance, a few classifiers are built from scratch on combining ANN with different nature inspired metaheuristic optimization algorithms such as Bacterial foraging optimization (BFO) [24], Genetic algorithm (GA) [25], Particle swarm optimization (PSO) [26], Invasive weed optimization (IWO) [27] and Firefly optimization (FA) [28][29], thus to obtain Bacterialforaging-ANN (BFO-ANN), Genetic-ANN (GA-ANN), ParticleSwarm-ANN (PSO-ANN), Invasive Weed-ANN (IWO-ANN) and Firefly-ANN (FA-ANN) classifier respectively. The performance is evaluated based on their capability of learning and its ability of generalization of Cervical cancer dataset with four targets. The fitness function used in this work is mean square error.

A. DATASET DETAILS

The cervical cancer dataset [11] taken from the UCI repository is employed during this study. This dataset contains 858 records with thirty two input attributes and 4 target variables namely Hinselmann, Schiller, Cytology and Biopsy. Hinselmann is the methodology to check the existence cervical cancer in an exceedingly patient by diagnosing their cells on a device named colposcope. Schiller is another test

that create use of applied iodine to the cervix. Then if the iodine turns brown color, this can be the indication of healthy cells else if it seems to be yellow or white and conjointly remains to be unstained, then this can be noted because the presence of abnormal cells. Cytology, a microscopic anatomy of a cervical cell which is done with Pap-smear examination below a microscope. Biopsy is the surgical primarily based extraction that is accustomed collect the tissues or cell sample for additional examination to find the existence of the abnormality. This dataset contains more missing details and as this could have an effect on the performance of a classifier in generating biased results. Therefore, the records are preprocessed and the samples with missing values are unheeded. As a results of preprocessing, 2 attributes are ignored (considered only 30 attributes) and a complete of 443 samples are taken for experimentation. The details of variety of healthy and abnormal samples thought-about for four targets are given in Table 1.

TABLE 1. Samples vs targets of cervical cancer dataset.

	Target_1 (Hinselm ann)	Target_2 (Schiller)	Target_3 (Cytology)	Target_4 (Biopsy)
Abnormal samples	18	39	28	27
Normal Samples	425	404	415	416

Binary classification is conducted for four target variables separately.

B. NETWORK SETTINGS

It is known that the ANN is also called as universal approximator. The function approximation is the mapping of input with output by updating the hyperparameters of the networks such as synaptic weights and bias values in a required number of iterations. Huge number of hidden neurons increases complexity that causes overfitting and less numbers affects the function approximation which in turn decreases its ability of generalization.

It is known from the literatures that for most of the function approximations, a 3-layered ANN is sufficient. The attributes present in the dataset decides the number of nodes in input layer and output layer. The hidden layer neurons are taken as $2N \pm 1$ [30], [31]. Hence for this work, number of nodes taken for input layer, hidden layer and output layer are taken as 30, 61 and 1 respectively, that is the architecture of ANN architecture is 30-61-1. Sigmoidal function is used in all nodes of the network as the activation function.

C. NETWORK SETTINGS

For any sample j , error value for each output neuron m is evaluated as the difference in target_output (T_m^j) and network_output (y_m^j) and it got using (19),

$$E^j = \sum_{m=1}^M (T_m^j - y_m^j)^2 \quad (19)$$

This is mean square error (MSE) [32] for each training sample. Hence error is got for the whole training data of size S which gives overall performance of classifier. Fitness function (F) of metaheuristic optimization algorithm will be average MSE of all training records. This is formulated and given in (20)

$$\text{Minimize } F = \sum_{j=1}^S \frac{E^j}{j} \quad (20)$$

Minimization of MSE is measured as fitness value in ANN training.

D. SEARCH AGENTS ENCODING

The number of jellyfish in a population is taken as NP and every search agent in the jellyfish population, Sa_i is encoded as a vector of dimension, $\text{Dim} = (N * H) + (H * M) + H + M$, where the size of input layer, hidden layer and output layer are respectively the N, H and M . Dim is number of elements present in each search agent, that constitutes the weights and biases of ANN. Hence, each search agent Sa_i is represented below using (21),

$$Sa_i = \left\{ \vec{W}1, \vec{B}1, \vec{W}2, \vec{B}2, \right\}, i = 1, 2, \dots, NP \quad (21)$$

where $\vec{W}1 = \{w_{111}, w_{112}, \dots, w_{1HN}\}$ and $\vec{W}2 = \{w_{211}, w_{212}, \dots, w_{2HN}\}$ are weight vectors constituting weights between hidden layer - input layer and output layer & hidden layer respectively. $\vec{B}1 = \{b_{111}, b_{112}, \dots, b_{1HN}\}$ and $\vec{B}2 = \{b_{211}, b_{212}, \dots, b_{2HN}\}$ are the bias vectors respectively for hidden layer neurons and output layer nodes.

E. HYPERPARAMETER DETAILS

The details of the hyperparameters of MHOAs (discussed in the Section 3) used in this work along with that of JSA algorithm are tabulated in Table 2. The population size and the maximum number of generations are set as 50 and 250 respectively, for all the experiments conducted. The experiments are conducted for the maximum generations.

IV. RESULT ANALYSIS

The Jellyfishsearch_ANN classifier is analyzed by comparing its performance with other comparison algorithms considered for this study. The results are analyzed in this section in terms of classifiers' learning capability and generalization capability.

A. CLASSIFIERS' LEARNING CAPABILITY

The proposed algorithm and its comparison algorithms are trained. The details of learning of these algorithms are visually depicted as convergence graphs using the figures Fig. 3(a-d). The convergence graphs that are used have following representations, number of generations in X-axis and the fitness values given in terms of MSE that are generated in every generation is marked towards Y-axis. The Table 3 provides the learning details in terms of Fitness

TABLE 2. Control parameter value details.

Algorithm	Parameter	Value
Jelly fish search	Distribution coefficient (γ)	3
	Motion coefficient(α)	0.1
Invasive weed	Standard deviation (σ_{initial}) - Begin	2
	Standard deviation (σ_{final}) - End	0.01
	Modulation index (n) - Non-linear	3
Particle swarm	Constants [C_1, C_2] - Acceleration	[2,2]
	Weight (w) - Inertia	0.9
Bacterial foraging	Number_Chemotactic (Nc) steps	25
	Number_swim (Ns) steps	4
	Number_reproductive (Nre) steps	4
	Level of elimination and dispersal (Ned)	2
Genetic	Type of selection	Roulette_wheel
	ProbabilityIndex_crossover	1
	Mutation probability	0.01
Firefly	Absorption_Coefficient(γ)	1
	Attractiveness_initial_value(β_0)	1
	Mutation_Coefficient(α)	0.2

value (mean MSE \pm SD, best_fitness) and training computation time(Tr_Ct), that is obtained at each generation. It is observed that the learning capability of the proposed JSA_ANN improves on training, as the best fitness value is achieved. In this work, the minimization of MSE or the fitness value without getting stuck in local minima is considered as convergence and this is achieved as expected after some iterations.

It is observed from Table 3, that the JellyfishSearch_ANN has converged without getting into the local minima with the fitness values \pm Standard deviation(SD) for the targets namely target_1, target_2, target_3 and target_4 as 0.00125 ± 0.00032 , 0.00129 ± 0.00039 , 0.00131 ± 0.00044 and 0.00119 ± 0.00050 respectively. These are the minimum mean square error produced by the proposed algorithm when compared with all other techniques. It can also be noted that both the algorithms namely Genetic_ANN and JellyfishSearch_ANN are found to be competitive as they have produced relatively similar fitness for the targets. ParticleSwarm_ANN comes subsequent to Genetic_ANN besides for target_1 as Bacterialforaging_ANN got the chance. The Invasive Weed_ANN has contributed a worst fitness value in terms of target_1 and Firefly_ANN has produced a worst fitness for target_2, target_3 and target_3 respectively. For target_1, target_2, target_3 and target_4, JellyfishSearch_ANN has yielded the best fitness of 0.00063, 0.00060, 0.00064 and 0.00059 respectively.

The JellyfishSearch_ANN has taken a lesser computation time than all others, as 167.85s, 171.83s, 171.4s and 174.61s for the target_1, target_2, target3 and target_4 respectively. Next, a slightly higher time is utilized by Genetic_ANN. The Firefly_ANN has taken a largest time but failed to converge with optimal solution. From the convergence graph in

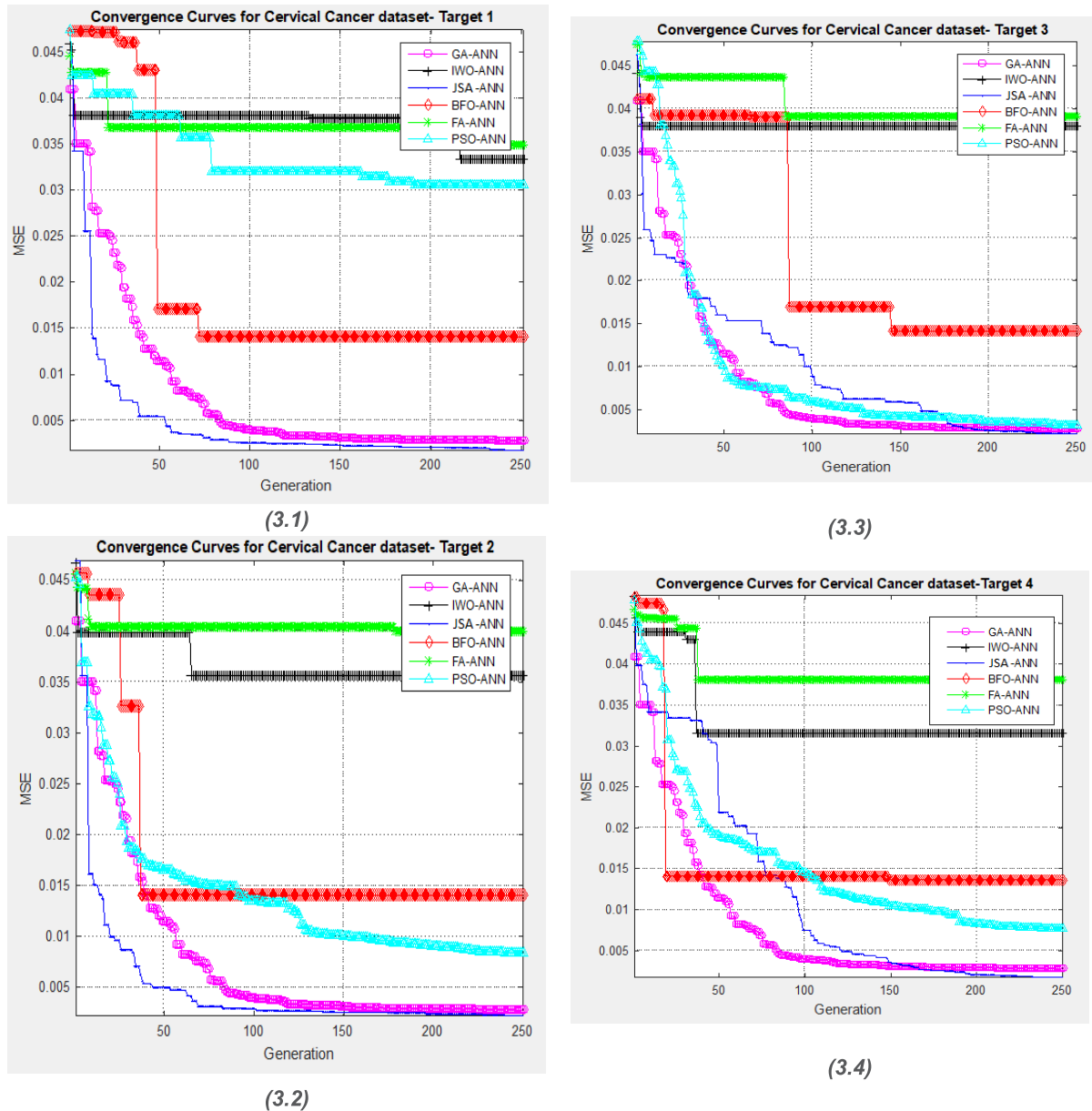


FIGURE 3. Convergence graphs showing learning outcome of classifiers for (3.1) Target_1 (3.2) Target_2 (3.3) Target_3 and (3.4) Target_4.

Fig. 3(a-d), it is graphically determined that the algorithms namely Bacterialforaging_ANN, Invasive Weed_ANN and Firefly_ANN have not shown any improvements after some generations and they struck in local minima and lost their ability of transiting from exploration to exploitation.

B. CLASSIFIERS' GENERALIZATION CAPABILITY

After the training of a classifier, its generalization functionality is constantly evaluated. The generalization functionality of classifier is its cap potential to categorize any unpracticed data. The testing details of all the classifiers is presented in Table 4 for all four targets. The testing output produced by the classifiers are analyzed in terms of the performance metrics such as specificity ($S_{pec}\%$), sensitivity

($S_{ens}\%$), precision ($P_{rec}\%$), MCR ($MCR\%$) and accuracy ($Acc\%$). As the dataset is imbalanced, the metrics such as G_{mean} measure and F_{-score} degree are also calculated to evaluate the classifiers' performance [32]. The confusion matrix representing the performance of the classifiers in the testing phase towards all the targets is displayed with the Figures 4–7.

It is inferred from the Table 4, that the Jellyfish-Search_ANN has yielded the accuracy of 99.324 %, 99.097 %, 98.871 % and 99.323 % for target_1, target_2, target_3 and target_4 respectively. The next higher value of accuracy for all targets is achieved by the competitive algorithm, Genetic_ANN. The proposed Jellyfish-Search_ANN has produced (specificity ($Spec\%$), sensitivity

TABLE 3. Details of classifiers' learning capability.

METHODS	FUNCTION: FITNESS			TR_CT (s)
	MEAN	S _D	BEST FIT	
TARGET_SET 1				
JELLYFISHSEARCH_ANN	0.00125	0.00032	0.00063	167.85
GENETIC_ANN	0.00385	0.00089	0.00091	253.41
PARTICLESWARM_ANN	0.03981	0.06512	0.03286	274.98
BACTERIALFORAGING_ANN	0.01911	0.00252	0.01817	286.92
INVASIVE WEED_ANN	0.04759	0.13299	0.04232	288.88
FIREFLY_ANN	0.04544	0.14349	0.03612	377.7
TARGET_SET 2				
JELLYFISHSEARCH_ANN	0.00129	0.00039	0.0006	171.83
GENETIC_ANN	0.0032	0.00086	0.00081	197.04
PARTICLESWARM_ANN	0.01089	0.00419	0.00859	266.81
BACTERIALFORAGING_ANN	0.01416	0.07189	0.01313	236.98
INVASIVE WEED_ANN	0.04869	0.1237	0.04109	251.83
FIREFLY_ANN	0.05819	0.13306	0.05309	281.4
TARGET_SET 3				
JELLYFISHSEARCH_ANN	0.00131	0.00044	0.00064	171.4
GENETIC_ANN	0.00415	0.00076	0.00119	199.5
PARTICLESWARM_ANN	0.00771	0.01319	0.00629	267.94
BACTERIALFORAGING_ANN	0.03409	0.06252	0.02252	239.16
INVASIVE WEED_ANN	0.0516	0.14145	0.03685	250.61
FIREFLY_ANN	0.05714	0.17322	0.0771	280.92
TARGET_SET 4				
JELLYFISHSEARCH_ANN	0.00119	0.0005	0.00059	174.61
GENETIC_ANN	0.00311	0.00087	0.00154	208.74
PARTICLESWARM_ANN	0.01009	0.01529	0.0022	273.65
BACTERIALFORAGING_ANN	0.01899	0.07139	0.01214	243.62
INVASIVE WEED_ANN	0.02339	0.10345	0.03699	254.25
FIREFLY_ANN	0.0473	0.17269	0.04395	287.43

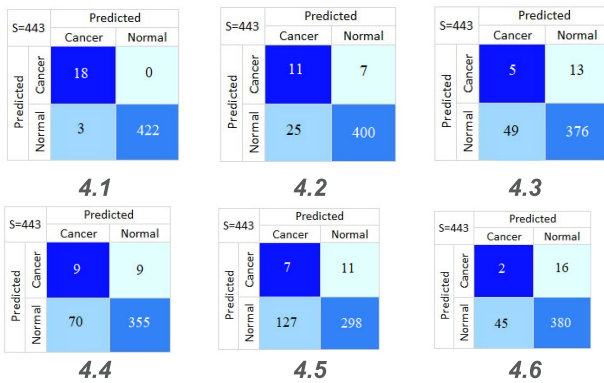


FIGURE 4. Confusion matrix of classifiers, JellyfishSearch_ANN(4.1), Genetic_ANN(4.2), ParticleSwarm_ANN(4.3), Invasive Weed_ANN(4.4), Firefly_ANN(4.5) and Bacterialforaging_ANN(4.6) for the Target_1.

(Sens%), precision (Prec%)) of (99.529/ 100 / 90), (99.505 / 94.872 / 94.872), (99.036 / 96.429 / 87.097) and (99.76 / 92.593 / 96.154) for target_1, target_2, target_3 and target_4 respectively. This is the maximum score produced by the proposed algorithm among all other comparison methods. Genetic_ANN produces the next higher score in terms of all measures. The above discussed performance metrics gives the overall performance of the algorithm irrespective of the classes.

To analyze the performance of classifiers with respect to the classes, the two metrics namely Gmean and F-score measures are included. This is because of the skewness found

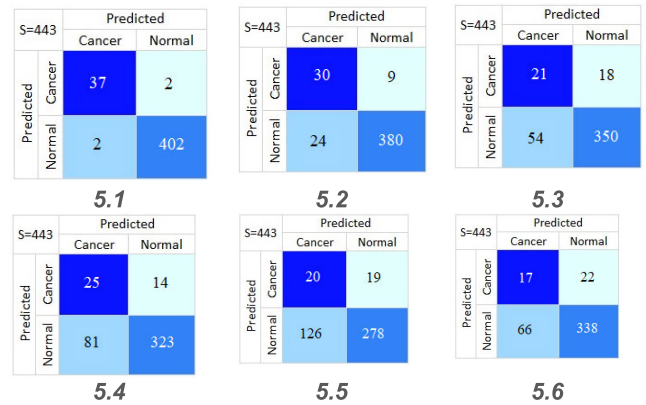


FIGURE 5. Confusion matrix of classifiers, JellyfishSearch_ANN(5.1), Genetic_ANN(5.2), ParticleSwarm_ANN(5.3), Invasive Weed_ANN(5.4), Firefly_ANN(5.5) and Bacterialforaging_ANN(5.6) for the Target_2.

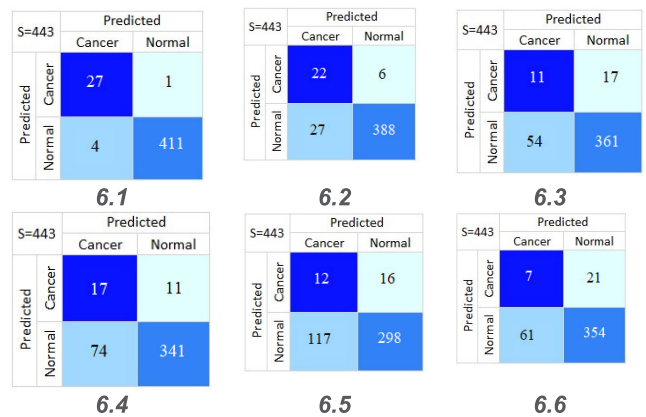


FIGURE 6. Confusion matrix of classifiers, JellyfishSearch_ANN(6.1), Genetic_ANN(6.2), ParticleSwarm_ANN(6.3), Invasive Weed_ANN(6.4), Firefly_ANN(6.5) and Bacterialforaging_ANN(6.6) for the Target_3.

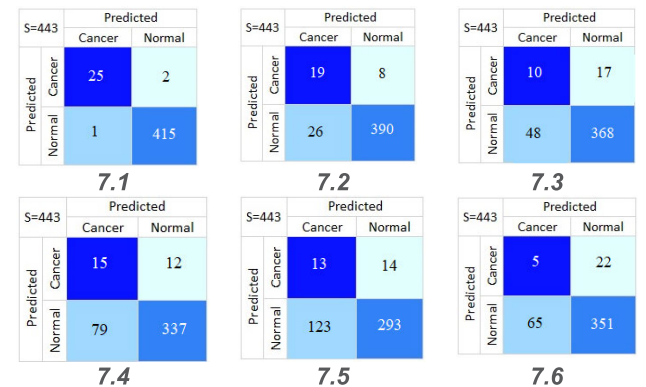


FIGURE 7. Confusion matrix of classifiers, JellyfishSearch_ANN(7.1), Genetic_ANN(7.2), ParticleSwarm_ANN(7.3), Invasive Weed_ANN(7.4), Firefly_ANN(7.5) and Bacterialforaging_ANN(7.6) for the Target_4.

in the dataset as the number of normal samples are more when compared to unhealthy samples. This condition mostly prevails in datasets normally built for binary classification. In a condition such as the classifier might yield higher accuracy values though it wrongly classifies the unhealthy samples

TABLE 4. Details of classifiers’ generalization capability.

METHOD	ACCU RACY	MCR	SENS	SPEC	PREC	GME AN	F- SCORE
TARGET_SET 1							
JELLYFISHSEAR CH_ANN	99.32	0.67	100	99.30	90.01	99.7	0.923
GENETIC_ANN	92.78	7.22	61.11	94.12	30.56	0.76	0.407
PARTICLESWARM_ANN	86.01	14.00	27.78	88.47	9.26	0.51	0.139
BACTERIALFORAGING_ANN	86.23	13.77	11.11	89.41	4.26	0.32	0.062
INVASIVE WEED_ANN	82.17	17.83	50.01	83.53	11.39	0.65	0.186
FIREFLY_ANN	68.85	31.15	38.89	70.12	5.22	0.52	0.092
TARGET_SET 2							
JELLYFISHSEAR CH_ANN	99.1	0.9	94.87	99.51	94.87	0.97	0.949
GENETIC_ANN	92.55	7.45	76.92	94.06	55.56	0.85	0.645
PARTICLESWARM_ANN	83.75	16.25	53.85	86.63	28.11	0.68	0.368
BACTERIALFORAGING_ANN	80.14	19.87	43.59	83.66	20.48	0.6	0.279
INVASIVE WEED_ANN	78.56	21.45	64.11	79.95	23.59	0.72	0.345
FIREFLY_ANN	67.27	32.73	51.28	68.81	13.7	0.59	0.216
TARGET_SET 3							
JELLYFISHSEAR CH_ANN	98.87	1.13	96.43	99.04	87.1	0.98	0.915
GENETIC_ANN	92.55	7.45	78.57	93.49	44.9	0.86	0.571
PARTICLESWARM_ANN	83.97	16.03	39.29	86.99	16.92	0.59	0.237
BACTERIALFORAGING_ANN	81.49	18.51	25	85.3	10.29	0.46	0.146
INVASIVE WEED_ANN	80.81	19.19	60.71	82.17	18.68	0.71	0.286
FIREFLY_ANN	69.98	30.02	42.86	71.81	9.3	0.56	0.153
TARGET_SET 4							
JELLYFISHSEAR CH_ANN	99.32	0.68	92.59	99.76	96.15	0.96	0.943
GENETIC_ANN	92.33	7.68	70.37	93.75	42.22	0.81	0.528
PARTICLESWARM_ANN	85.33	14.67	37.04	88.46	17.24	0.57	0.235
BACTERIALFORAGING_ANN	80.36	19.64	18.52	84.38	7.14	0.4	0.103
INVASIVE WEED_ANN	79.46	20.54	55.56	81.01	15.96	0.67	0.248
FIREFLY_ANN	69.07	30.93	48.15	70.43	9.56	0.58	0.16

due to the negligible number, then Gmean and F-score will give the value as zero. It is explicitly found from Gmean and F-score produced by JellyfishSearch_ANN that the classifier has given an outstanding performance on perfectly discriminating the samples of both the classes. The proposed method has given equal preference in classifying both the classes as its Gmean and F-score is greater than 96 and 91 respectively.

Again Genetic_ANN has generated the next higher values of Gmean and F-score. Then comes the classifiers Invasive Weed_ANN and Firefly_ANN with some reasonable discrimination of samples. But ParticleSwarm_ANN classifier and Bacterialforaging_ANN classifier have only concentrated in classifying correctly the healthy samples. Also, these classifiers have produced a lower values for all other matrices, thus with an equal performance. Firefly_ANN is the lowest performer when compared with all others.

The accuracy produced by the traditional ANN from the recent works carried out in [33] and [34] for the dataset [11] are found to be 97.67% and 95% respectively. This is inferred to be lesser than our proposed approach.

V. CONCLUSION

The intelligence living behaviour of artificial jellyfish is employed to optimize ANN. This classifier conjointly referred to as JellyfishSearch_ANN is applied for accurate and efficient classification of cervical cancer dataset that has four targets. To validate the efficiency of the proposed work, five similar type of ANN classifiers are implemented using other metaheuristic algorithms such as PSO, IWO, GA, FA and BFO. The learning capability is analysed in terms of mean fitness, best fitness and computation time. It is found only the Genetic_ANN has produced some reasonable results but lesser than the proposed approach. But other comparison algorithms are found to be struck with local minima without further improvements when the population size and generations were increased. But a considerable progress is seen in the proposed approach. The performance evaluation metrics such as precision, specificity, sensitivity, Gmean, MCR, F-score and accuracy are used to calculate classifier’s generalization capability. Hence from the experimentation results, it is confirmed that JSA-ANN has yielded the outstanding results for all the targets than all other comparison classifiers. Among all the targets, it is also observed that the proposed classifier has achieved best result for the target1 that is target values based on Hinselmann test. The limitation of this project is that a particular feature selection algorithm is not used. In future, it is planned to use a suitable feature selection algorithm with this work to further increase the accuracy. This algorithm can also be used for the effective diagnosis of other diseases that are challenging the medical domain.

REFERENCES

- [1] WHO, *Comprehensive Cervical Cancer Control—A Guide to Essential Practice*, 2nd ed., World Health Organization, Geneva, Switzerland, 2014.
- [2] C. Haie-Meder, P. Morice, and M. Castiglione, “Cervical cancer: ESMO clinical practice guidelines for diagnosis, treatment and follow-up,” *Ann. Oncol.*, vol. 21, pp. v37–v40, May 2010.
- [3] P. L. Martin-Hirsch and N. J. Wood, “Clinical evidence handbook: Cervical cancer,” *Amer. Family Physician*, vol. 85, pp. 1186–1187, Mar. 2012.
- [4] J. J. Tanimu, M. Hamada, M. Hassan, H. Kakudi, and J. O. Abiodun, “A machine learning method for classification of cervical cancer,” *Electronics*, vol. 11, p. 463, 2022, doi: 10.3390/electronics11030463.
- [5] W. William, A. Ware, A. H. Basaza-Ejiri, and J. Obungoloch, “Cervical cancer classification from Pap-smears using an enhanced fuzzy C-means algorithm,” *Informat. Med. Unlocked*, vol. 14, pp. 23–33, Feb. 2019.
- [6] K. Sneha and C. Arunvinodh, “Cervical cancer detection and classification using texture analysis,” *Biomed. Pharmacol. J.*, vol. 9, no. 2, pp. 663–671, Aug. 2016.
- [7] K. P. Win, Y. Kitjaidure, K. Hamamoto, and T. Myo Aung, “Computer-assisted screening for cervical cancer using digital image processing of pap smear images,” *Appl. Sci.*, vol. 10, no. 5, p. 1800, Mar. 2020.

- [8] A. Q. B. Silva, "Predicting cervical cancer with metaheuristic optimizers for training LSTM," in *Computational Science—(ICCS)* (Lecture Notes in Computer Science), vol. 11540, J. Rodrigues, Ed. Cham, Switzerland: Springer, 2019, pp. 642–655.
- [9] D. Zahras and Z. Rustam, "Cervical cancer risk classification based on deep convolutional neural network," in *Proc. Int. Conf. Appl. Inf. Technol. Innov. (ICAITI)*, Padang, Indonesia, Sep. 2018, pp. 149–153.
- [10] Z. Rustam, V. A. W. Hapsari, and M. R. Solihin, "Optimal cervical cancer classification using Gauss-Newton representation based algorithm," in *Proc. AIP Conf.*, vol. 2168, 2019, Art. no. 020045.
- [11] M. F. Ijaz, M. Attique, and Y. Son, "Data-driven cervical cancer prediction model with outlier detection and over-sampling methods," *Sensors*, vol. 20, no. 10, p. 2809, May 2020, doi: [10.3390/s20102809](https://doi.org/10.3390/s20102809).
- [12] Z. K. Şentürk and S. Uzun, "An improved deep learning based cervical cancer detection using a median filter based preprocessing," *Eur. J. Sci. Technol.*, vol. 32, pp. 50–58, Jan. 2022.
- [13] B. J. Priyanka, "Machine learning approach for prediction of cervical cancer," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 8, pp. 3050–3058, 2021.
- [14] Q. Yin, "The application of machine learning in cervical cancer prediction," in *Proc. 6th Int. Conf. Mach. Learn. Technol.*, New York, NY, USA, Apr. 2021, pp. 12–19.
- [15] M. Arora, S. Dhawan, and K. Singh, "Improved performance of machine learning algorithms for prognosis of cervical cancer," *Adv. Comput. Des.*, vol. 6, no. 3, pp. 191–205, 2021.
- [16] L. W. Habtemariam, E. T. Zewde, and G. L. Simegn, "Cervix type and cervical cancer classification system using deep learning techniques," *Med. Devices: Evidence Res.*, vol. 15, pp. 163–176, Jun. 2022.
- [17] M. Kruczkowski, A. Drabik-Kruczkowska, A. Marciniak, M. Tarczewska, M. Kosowska, and M. Szczerska, "Predictions of cervical cancer identification by photonic method combined with machine learning," *Sci. Rep.*, vol. 12, no. 1, p. 3762, Dec. 2022, doi: [10.1038/s41598-022-07723-1](https://doi.org/10.1038/s41598-022-07723-1).
- [18] H. White, "Learning in artificial neural networks: A statistical perspective," *Neural Comput.*, vol. 1, no. 4, pp. 425–464, 1989.
- [19] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.
- [20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.
- [21] M. Gori and A. Tesi, "On the problem of local minima in backpropagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 1, pp. 76–86, Jan. 1992.
- [22] T. L. Fine, *Feedforward Neural Network Methodology*. Springer, 1999.
- [23] J.-S. Chou and D.-N. Truong, "A novel Metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl. Math. Comput.*, vol. 389, Jan. 2021, Art. no. 125535.
- [24] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Mar. 2002.
- [25] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
- [26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.
- [27] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecol. Informat.*, vol. 1, no. 4, pp. 355–366, Dec. 2006.
- [28] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. London, U.K.: Luniver Press, 2008.
- [29] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications (SAGA)* (Lecture Notes in Computer Sciences), vol. 5792, O. Watanabe and T. Zeugmann, Eds. Berlin, Germany: Springer, 2009, pp. 169–178.
- [30] V. Kárková, "Kolmogorov's theorem is relevant," *Neural Comput.*, vol. 3, no. 4, pp. 617–622, 1991.
- [31] T. Hegazy, O. Moselhi, and P. Fazio, "Developing practical neural network application using backpropagation," *J. Microcomput. Civil Eng.*, vol. 9, no. 2, pp. 145–159, 1994.
- [32] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2001.
- [33] B. Sundarambal, C. Karthikeyini, R. M. Bommi, S. Subramanian, and V. Jacintha, "Comparison of machine learning and deep learning models for cervical cancer prediction," in *Proc. 6th Int. Conf. Devices, Circuits Syst. (ICDCS)*, Apr. 2022, pp. 135–138, doi: [10.1109/ICDCS54290.2022.9780705](https://doi.org/10.1109/ICDCS54290.2022.9780705).
- [34] M. Mehmood, M. Rizwan, M. Gregus MI, and S. Abbas, "Machine learning assisted cervical cancer detection," *Front Public Health*, vol. 9, Dec. 2021, Art. no. 788376, doi: [10.3389/fpubh.2021.788376](https://doi.org/10.3389/fpubh.2021.788376).



DEVIKANNIGA DEVARAJAN received the B.E. degree in information technology from the Sethu Institute of Technology, Madurai Kamraj University, India, in 2003, the M.Tech. degree in advanced computing from Tifac Core, SASTRA University, India, in 2005, and the Ph.D. degree in computer science and engineering in machine learning domain from Anna University, India, in 2019. She has published several research and review articles in reputed journals. Her research interests include the development of various machine learning algorithms optimized with metaheuristics optimization algorithms for various medical applications and others, and also several projects on computer vision. She has received the Best Teacher Award, for her excellence in teaching and mentoring.



D. STALIN ALEX received the Master of Technology degree from Manonmaniam Sundaranar University, Tirunelveli, and the Ph.D. degree from Anna University, Chennai, India, in 2017. He has 15 years of teaching experience and eight years of the Head of the Department experience in India and abroad. He is expert in data science, artificial intelligence, machine learning, and image processing. He has published 26 research articles indexed in SCOPUS/SCI/UGC Care journals. He filed two patents. His research interests include image processing, video processing, machine learning, artificial intelligence, and data science.



T. R. MAHESH is currently working as an Associate Professor and a Program Head of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be-University), Bengaluru, India. He has to his credit more than 40 research papers in Scopus and SCIE indexed journals of high repute. His research interests include image processing, machine learning, deep learning, artificial intelligence, the IoT, and data science. He has

been the editor for books on emerging and new age technologies with publishers like Springer, IGI Global, and Wiley. He has served as a reviewer and a technical committee member for multiple conferences and journals of high reputation.



V. VINOTH KUMAR is an Associate Professor with the School of Information Technology and Engineering, VIT University, Tamilnadu, India. He is the author or coauthor of papers in international journals and conferences, including SCI indexed papers. He has published as over than 35 papers in IEEE ACCESS, Springer, Elsevier, IGI Global, and Emerald. His current research interests include wireless networks, the Internet of Things, machine learning, and big data applications. He is also an Associate Editor of *International Journal of E-Collaboration (IJeC)* and *International Journal of Pervasive Computing and Communications (IJGCC)*, and an editorial member of various journals.



RAJANIKANTH ALUVALU (Senior Member, IEEE) received the Ph.D. degree in cloud computing as specialization. He is currently working as an Associate Professor with the Department of IT, Chaitanya Bharathi Institute of Technology, Hyderabad, India. Formerly, he had held positions, including a Professor and the Head of the Department of CSE, Vardhaman College of Engineering, Hyderabad; the Vice-Chair of the Entrepreneurship and Startup Committee; and a Treasurer and the Secretary of the IEEE Computer Society, Hyderabad Section. He is having more than 17 years of Teaching Experience. He has organized various international conferences and delivered keynote addresses. He has published more than 60 research papers in various peer-reviewed journals and conferences. He is a Life Member of ISTE and a member of ACM and MIR Labs. He was a recipient of the Best Advisor Award from the IEEE Hyderabad Section as well as the IUCEE Faculty Fellow Award (2018). He has also been the Co-ordinator of the AICTE 'Margadarshan' Scheme, a Google Cloud Facilitator, an Editor of *IJDMMM* journal published by Inderscience, and the Academic Editor of *Computer Science (PeerJ)* journal, in addition to being the reviewer for several Scopus indexed and SCI indexed journals



V. UMA MAHESWARI (Member, IEEE) received the Ph.D. degree in image analytics and data science from Visveswaraya Technological University, Belgaum. She is currently working as an Associate Professor, the Head of CSE (AI and ML), KG Reddy College of Engineering and Technology, Hyderabad. She has published more than 20 research papers in SCI, ESCI, WoS, DBLP, and SCOPUS indexed journals and conferences. She has also published four Indian patents on facial expression analysis in the fields of medical, e-commerce, education, and security. She has done an enormous study and given contributions in facial expression analysis and applications. She constructed feature vector for a given image based on the directions and introduced dynamic threshold values while comparing the images, which helps to analyze any image. She has researched the similarity of images in a given database to retrieve the relevant images. She also worked with convolutional neural networks by giving the pre-processed input image to improve the accuracy. It has been proved that the maximum edge intensity values are enough to retrieve the required feature from the image instead of working on total image data. She is the Co-ordinator for TEDxVCE. She has organized various technical programs and served as a technical committee member and a reviewer for various conferences. She has delivered sessions in various capacities. She received the Best Faculty Award under the innovation category from the CSI Mumbai Chapter for the year 2019



S. SHITHARTH (Senior Member, IEEE) received the B.Tech. degree in information technology from the KGiSL Institute of Technology, Coimbatore, India, in affiliation with Anna University, Chennai, India, in 2012, the M.E. degree in computer science and engineering from the Thiagaraja College of Engineering, Madurai, India, in affiliation with Anna University, in 2014, and the Ph.D. degree from the Department of Computers Science and Engineering, Anna University. He is currently pursuing the Ph.D. degree with the University of Essex. He worked in various institutions with a teaching experience of seven years. He is also working as an Associate Professor with Kebri Dehar University, Ethiopia. He has published more than 45 international journals along with 20 international and national conferences. He has even published four patents in IPR. His current research interests include cyber security, blockchain, critical infrastructure and systems, network security, and ethical hacking. He is also an Active Member of IEEE Computer Society and five more Professional Bodies, and also a member of the International Blockchain Organization. He is a Certified Hyperledger Expert and a Certified Blockchain Developer. He is an active researcher, reviewer, and editor for many international journals.

...