AN EVOLUTIONARY MODIFIED DRAGONFLY NEURO-FUZZY INFERENCE SYSTEM FOR CLASSIFICATION OF TUBERCULOSIS

D.Arul Suresh[1], J.Jude Moses Anto Devakanth[2], Dr.R.Balasubramanian[3]
[1,2]Research Scholar, [3]Professor
Department of Computer Science and Engineering,
Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli.
***Corresponding Author:** suresharul93@gmail.com

Abstract

Tuberculosis (TB) is an infectious disease which is caused by a bacterium called Mycobacterium tuberculosis and it affects human lungs. The number of TB related deaths worldwide in 2021 was 1.6 million. TB is the second most lethal infectious disease after COVID-19 and the 13th major cause of death globally. World-wide cases of tuberculosis (TB) are predicted to reach 10.6 million in 2021. The interesting fact is that the bacterium which causes TB continues to exist in human body without exhibiting any symptoms. So, it is necessary to predict the occurrence of TB at an early stage. This prediction is possible by performing classification using deep learning model. For classification, features from Chest X ray images need to be extracted. More number of features sometimes leads to less accuracy and this issue can be rectified using feature selection strategy. Genetic Algorithm (GA) and Dragonfly are the evolutionary algorithms which finds its application in feature selection and has the capability to provide promising results. In this research, a novel neuro fuzzy based system to classify tuberculosis namely OMDNFIS (Optimized Modified Dragonfly Neuro Fuzzy Inference System) with DenseNet121 for better classification accuracy. Experimental results are compared with the individual as well as few existing hybrid algorithms and it is proved that proposed OMDNFIS model outperforms the individual deep learning algorithms and few existing deep learning models.

1. Introduction

With every passing day, almost everything is being automated with the help of machines that can think on their own. Healthcare is one of the fundamental aspects of human life [1]. Lung diseases are often associated with excruciating pain and suffering as they affect the breathing pattern of the patient due to suffocation and related symptoms. Tuberculosis (TB) is one such detrimental variant of lung infections that has created a devastating impact on humankind. Tuberculosis is caused by the Mycobacterium tuberculosis bacteria. In general, the lungs are the main target area of this disease, but it can also affect other parts of the body. TB is a contagious disease, i.e., when people infected with TB cough or sneeze, they transmit the disease-causing bacteria in the air. Only a small quantity of these germs is enough to effectively infect a healthy person. Although scientific discoveries and research have been helping to curb the growing influence of TB, the meagre annual medical progress rate in this sector has been unsuccessful in bringing a drastic drop of TB affected patients. According to the Global Tuberculosis Report, 2020, generated by the WHO [2], an estimated 10·6 million people became ill with tuberculosis in 2021, compared with 10.1 million in 2020, and 1.6 million people died from tuberculosis in 2021 (including 187 000 individuals living with HIV). In addition, the incidence rate of tuberculosis increased by 3.6% in 2021 relative to 2020, suggesting a reversal from the trend of nearly 2% decrease per year during the past two decades.

With advances in deep learning, the Convolutional Neural Networks (CNNs) have consistently surpassed other traditional recognition algorithms in achieving super ordinate performance for image-based classification and recognition problems [3]. Deep learning has the ability to extract features from the input images by itself and performs classification. In addition to that, it also accepts manually extracted features for processing. In case of importing manually extracted features, the accuracy may depend on number of features. If the number of features is high or if it includes irrelevant features, the accuracy may be decreased. In such case, feature selection strategy can be utilized to reduce the number of features which can reduce the irrelevant features as feature selection selects only the most relevant features for classification. Hence, feature selection helps to increase prediction or detection accuracy and also aids in reducing the computation time. Finally, a neural network-based approach namely OMDNFIS (Optimized Modified Dragonfly Neuro Fuzzy Inference System) based on Fuzzy Logic and a Deep learning model with feature selection is developed in this research work.

2. Related Work

A novel approach to identify tuberculosis bacteria based on shape and colour was proposed by M.Forero et al. (2004) [4]. Designed algorithm technique was based on combined use of invariant shape features together of bacilli with simple thresholding operation on chromatic channels. This methodology is based on segmentation using K-means clustering which is followed by an identification procedure, for which 110 samples of bacilli was analyzed. Further, the author explored a colour-based edge segmentation technique using derivative operators to all chromatic channels and by using Bayesian decision theory [5]. N. Walia et al. (2015) [6] had designed a fuzzy based decision support system for the identification of tuberculosis. Their framework briefly explains relation between different input attributes and its symptoms. The author concluded that fuzzy basis dependent expert systems can be used during diagnosis. Further, author stated that the developed system can also be utilized for the detection of other chronic obstructive diseases using hybrid neuro systems [7].

An integrated approach for automated detection of early lung cancer and tuberculosis-based X-ray image analysis was demonstrated by K. Lee (2006) [8]. Various symptoms of the disease and finding nodules were focused. The proposed technique uses watershed segmentation approach to isolate a lung X-ray image, and then apply a small scanning window to determine whether any pixel is a part of a disease nodule or not. Additionally, various methods used to detect early signs of cancer and tuberculosis was also explained in this work [9]. Wenbo Li, Yan Kang (2015) [10] using a new adaptive VOI selection method. Twenty-two features were extracted to distinguish nodules, vascular endpoint or vascular cross structure, designed an optimal feature combination selection frame based on improved genetic algorithm and support vector machine. The improved GA algorithm is used to select the optimal feature combination from the feature pool to establish SVM classifier [11]. Asogbon MG (2016) [12] proposed enhanced Neuro-Fuzzy system based on genetic algorithm for medical diagnosis proposed the use of Genetic Algorithm (GA) technique to automatically evolve optimum connection weights needed to efficiently train a built ANFIS model used for Typhoid fever diagnosis. The GA module computes the best set of connection weights, stores them, and later supplies them to the corresponding hidden layer nodes for training the ANFIS [13].

An artificial intelligent approach for estimation of disease and resource utilization was discussed by E.Papageorgiou et al. (2009) [14]. Fuzzy cognitive map-based tool was used to represent medical diagnosis system concentrated on relating to lungs infections. Due to easy graphical representation approach, the proposed method makes wide use of computer consultation system. The presented system would offer a solution for requirements imposed by the target application, disease symptoms, signs and laboratory tests [15]. A decision based fuzzy expert system was developed to identify the stages of tuberculosis [16]. An efficient fuzzy based detection system was developed to detect tuberculosis and the system uses Mamdani inference method. The authors developed a mobile application in the name of “Fuzzy based Tuberculosis Diagnosis system” and it has been tested with an expert medical officer. [17]. A decision-making system was developed based on the set of rules [18]. The research work demonstrates the risk quotient of TB and its severity based on various parameters as inputs. Further, the accuracy of system is calculated and it came out to be 92%. This system would aid in the diagnosis of the different levels of tuberculosis.

3. METHODOLOGY

The input Chest X Ray (CXR) image is given as input to the proposed system. The input CXR image undergoes preprocessing for noise removal and gray scale transformation. The preprocessed image is given as input to the feature extraction method namely “First and Second Order Statistical (FSOS) feature extraction” which extracts n number of features from the input CXR images. The extracted features are given as input to the proposed “Modified Dragonfly Optimization Algorithm” to get an optimized set of features. Further, the optimized feature set or the reduced feature set are given as input to the deep leaning model namely

"DenseNet 121". At last, fuzzy logic principles are applied to obtain the classification results which will be evaluated using performance metrics. The overall architecture of the proposed OMDNFIS (Optimized Modified Dragonfly Neuro Fuzzy Inference System) is shown in Figure 1.

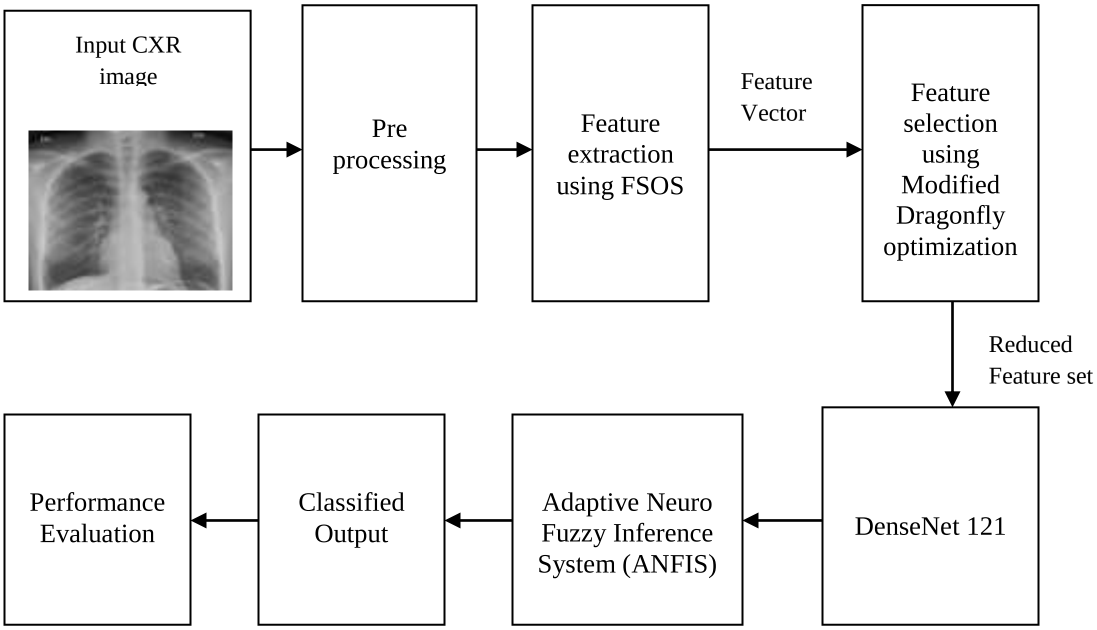

Figure 1. Outline of the work.

3.1. Pre processing

The input image is converted into gray scale image using gray scale transformation. The gray scale image undergoes the process of noise removal. In this research work, the noise from the CXR image is removed by applying median filter.

3.2. Feature extraction using FSOS

The pre-processed image is given as input to the feature extraction namely "First and Second Order Statistical (FSOS)" technique which extracts first and second order features from CXR images for classification. A number of texture features may be extracted from the first and second order statistics. In First order statistics, Gray Level Co-occurrence Matrix (GLCM) are formulated to obtain statistical texture features such as Mean, Energy, Entropy, Skewness, Variance, Uniformity etc. Second order features are extracted based on Gray Level Co-occurrence Matrix, Gray Level Run Length Matrix and Neighborhood Gray Tone Difference Matrix. A total of 44 features are extracted using First and Second Order Statistical (FSOS)" technique.

FSOS = First order statistical features + Second Order Statistical feature (1)

3.3. Proposed Modified Dragonfly Optimization (MDA) Algorithm

Seyedali Mirjalili [19] proposed an optimization algorithm, based on the swarming behavior of dragonflies, and named it Dragonfly Algorithm. The formation of sub-swarms and its back-and-forth flying, over a different area, makes static swarm very useful for exploitation phase. Similarly in dynamic swarm, flying of large number of dragonflies in one direction favors the exploration phase. Dragonfly algorithm uses three basic properties of swarm movement which are as follows.

(i) Seperation

All dragonflies, in a swarm, should maintain some distance from the neighborhood dragonfly to avoid the collision. Reynolds [20] has introduced a method to compute the separation of an individual from its neighboring individuals. Seperation can be calculated using the following equation.

$$S_i = -\sum_{j=1}^{M} X - X_j \tag{2}$$

where X indicates a dragonfly's position in a D-dimensional space (D is the number of decision variables), X_j is the position of the neighbor individual, and M is the number of neighbor individuals.

(ii) Alignment

All dragonflies should maintain nearly the same velocity as their neighbor. This is called Alignment A_i which can be calculated as given

$$A_i = \frac{\sum_{j=1}^{M} V_j}{M} \tag{3}$$

where V_j is the velocity of neighbor individuals, and M is the number of neighbor individuals.

(iii) Cohesion

The trajectory, of all dragonflies in a swarm, should be towards the center of mass of neighborhood. The Cohesion C_i is calculated using the following formula

$$C_i = \frac{\sum_{j=1}^{M} X_j}{M} - X \tag{4}$$

Where X_j indicates the position of the neighbor individual, and M is the total number of dragonflies in the neighborhood.

In addition to this, in order to survive, all individuals should move towards the food source but simultaneously should stay away from the enemy.

(iv) Food source

Attraction towards the food source is evaluated using the following formula

$$F_i = Xf - X \tag{5}$$

where *Xf* indicates the position of a food source.

(v) Enemy

Distraction from the enemy can be calculated using the formula given below

$$E_i = Xe + X \tag{6}$$

where *Xe* is the position of the enemy.

To upgrade the location of dragonfly in the seek domain and further to simulate its moment, DA [21] has consider two vectors: position (x) and step (Δx).

$$\text{DA optimization =updation of (} \Delta x + x) \tag{7}$$

Here, the step (Δx) vector is similar to the velocity vector in particle swarm optimization (PSO) [22] as, the DA is inspired from PSO. The step (Δx) vector indicates the direction of moment of swarm as is determine by the following equation

$$\Delta X_i(t+1) = (sS_i + \alpha A_i + cC_i + fF_i + eE_i) + w\Delta X_i(t+1) \tag{8}$$

Where *s* refers to the weight of separation, ∝ shows the weight of alignment, *c* shows the weight of cohesion, *f* is the weight of food and *e* is the weight of a enemy. Where w shows the inertia weight, and *t* is representing the current iteration. s, α, c, f, and e are the adjacent coefficients, (t+1+ demonstrate the iteration value and w is inertia weight. As per the DA [19] the value of s and α is 0.1, c is 0.7, and f and e are 1, furthermore, the value of w lies between 0.9 to 0.2.

For stochastic behavior in artificial dragonflies, it is assumed that dragonflies are flying in the search space using random walk also called as Levy fight in case of zero neighborhood. For this, the position of a dragonfly will be updated using the following equation.

$$X_{t+1} = X_t + Levy(d) \times X_t \tag{9}$$

In the standard DA to enhance the randomness and to execute exploration of the dragonfly. Levy flight mechanism is deployed to seek the domain by the random walk, as described in the Equation (9). It is found that during the execution of standard DA for engineering problems that deploying current position with levy flight may stag the solution, as this mechanism may keep the solution near the best find location that may or may not be the global position. In order to overcome this limitation and enhance the potential of DA the Equation (9) is rewritten as

$$X_{t+1} = X_t + Y.Levy\ (d) \tag{10}$$

Where $Y = \frac{S_{max}}{t^2}$

The constant Y value depends on the type of problem, S_{max} refers to the size of population, and the t refers to the maximum number of iterations.

Where d represents the dimension of the position vector. Levy flight [23] can be calculated as

$$Levy(x) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{1/\beta}} \tag{11}$$

Where r1 and r2 are two randomly generated real numbers in the range [0, 1] and β is a constant tuned during the experiment. σ is calculated using the following equation

$$\sigma = \left(\frac{\Gamma(1+\beta) \times Sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{1/\beta} \tag{12}$$

Where $\Gamma(x) = (x-1)!$.

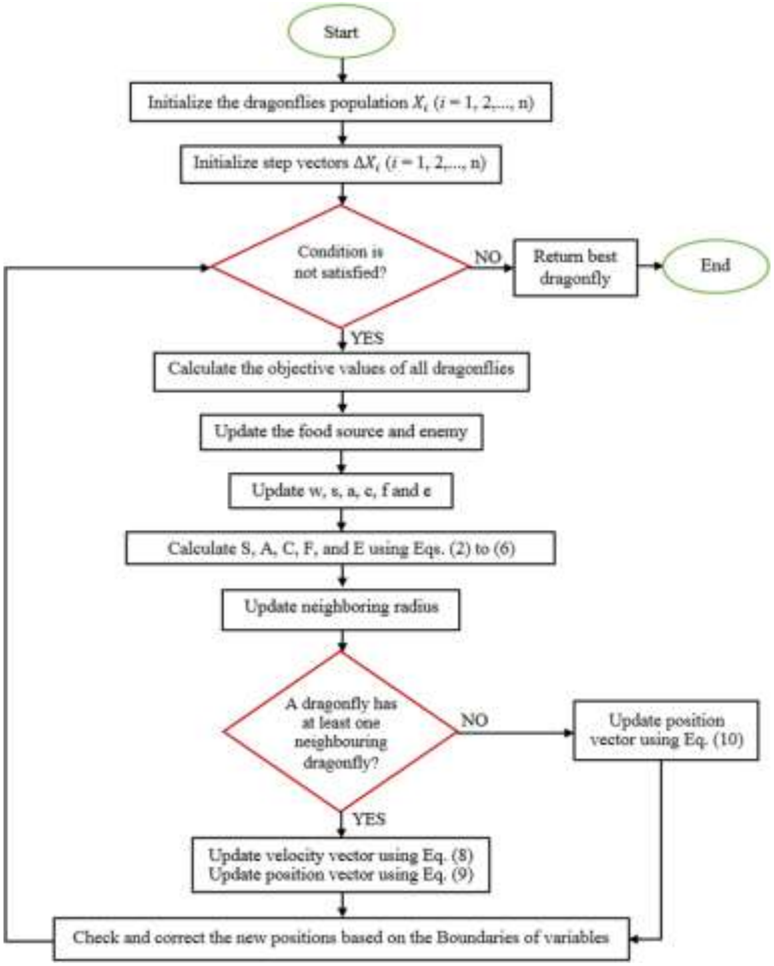

Figure 2. Modified Dragonfly Optimization

3.4 Existing Algorithms for Optimization

There are many existing optimization algorithms exists for the purpose of reducing the number of features for further processing such as classification. Among the existing optimization algorithms, two popular successful evolutionary optimization algorithms namely Genetic Algorithm (GA) [24] and Whale Optimization Algorithm (WOA) [25] are selected for comparison with the proposed Modified Dragonfly Optimization (MDA) algorithm. The detailed working procedure of GA and WOA is explained below.

3.4.1 Genetic Algorithm (GA)

The features extracted using FSOS is given as input into the optimization algorithm namely “Genetic Algorithm” which reduces the number of features for further processing. GA is a stochastic algorithm that mimics natural evolution. The most distinct aspect of this algorithm is that it maintains a set of solutions (called individuals or chromosomes) in a population. As in the case of biological evolution, it has a mechanism of selecting fitter chromosomes at each generation. To simulate the process of evolution, the selected chromosomes undergo genetic operations, such as crossover and mutation. The initial population will be modified to reach a better answer. At each step, the GA selects individuals (chromosomes) from the current population (parents) randomly and uses them to produce the children for the next generation. After several generations, according to essence of the GA, it tries to move to the best solution. The flowchart which represents working of Genetic Algorithm is shown in Figure 3.

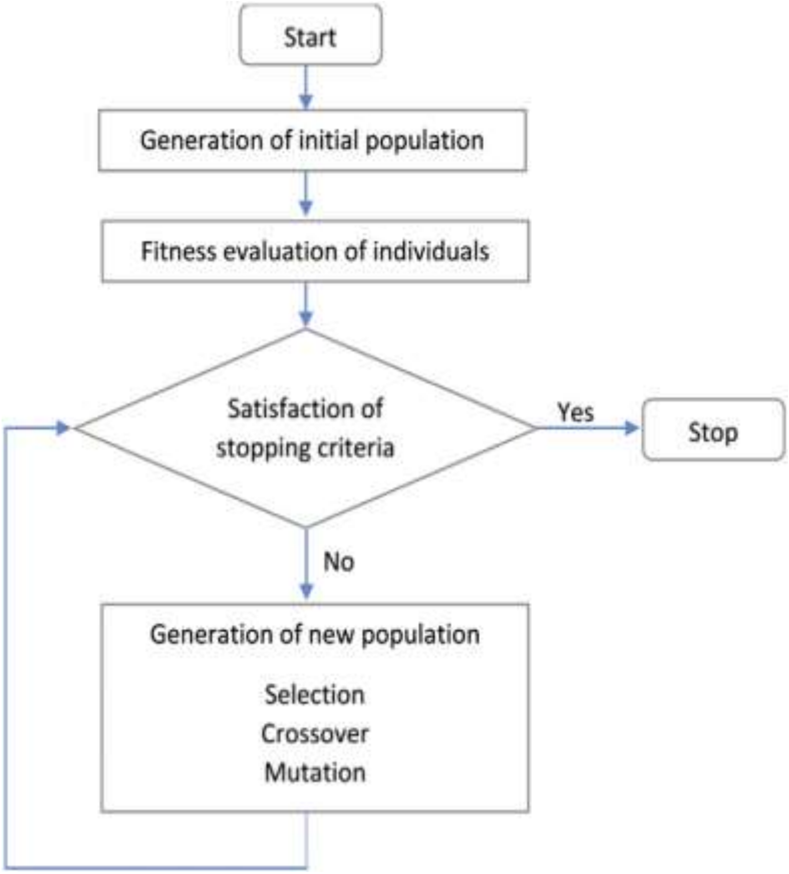

Figure 3. Flowchart of Genetic Algorithm

At each step, the GA uses three main types of rules to create the next generation from the current population. These types of rules are discussed as follows

Selection rules select the individuals' called parents which contribute to the population at the next generation.

Crossover rules combine the chromosomes in order to produce the next generation.

Mutation rules lead the chromosome to change and alter their values.

(i) Fitness function and selection operation

The initial population contains an initial set of features or individuals from which we have to select the best features for classification. In this research, the initial population consists of 44 features. Next step is to evaluate the fitness of each individual or features. The fitness function used here is

$$f(x) = x^2 \tag{13}$$

The next step is selection in which two Parents which has the lowest selection error or highest fitness score will be selected to generate the next offspring, which means those features will be selected for the model training.

(ii) Crossover and Mutation

The best food source or the features is selected for crossover and mutation. One point crossover is utilized in this research work. As a result of crossover and mutation, child chromosomes are generated. Then, the child chromosomes are evaluated using the fitness function and the children which is having the highest fitness value is selected for further process. The whole fitness assignment, updation, selection and mutation process is repeated until a stopping criterion is satisfied.

3.4.2 Whale Optimization Algorithm (WOA)

WOA belongs to the family of stochastic population-based algorithms proposed by Mirjalili and Lewis in 2016 [25]. The algorithm mimics two phases: the first phase (exploitation phase) is encircling a prey and spiral bubble-net attacking method and the second phase (exploration phase) is searching randomly for a prey.

(i) Exploration Phase

Whales can recognize the location of prey and encircle them. Since the position of the optimal design in the search space is not known a priori, the WOA algorithm assumes that the current best candidate solution is the target prey or is close to the optimum. After the best search agent is defined, the other search agents will hence try to update their positions towards the best search agent. This behavior is represented by the following equations:

$$\vec{D} = |\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)| \quad (14)$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} \quad (15)$$

Where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, X^* is the position vector of the best solution obtained so far, $\vec{X}$ is the position vector, | | is the absolute value, and is an element-by-element multiplication. It is worth mentioning here that $X*$ should be updated in each iteration if there is a better solution. The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (16)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (17)$$

Where $\vec{a}$ is linearly decreased from 2 to 0 over the course of iterations and $\vec{r}$ is a random vector in [0,1].

The same approach based on the variation of the $\vec{A}$ vector can be utilized to search for prey. whales search randomly according to the position of each other. Therefore, $\vec{A}$ with the random values greater than 1 or less than −1 is utilized to force search agent to move far away from a reference whale. The position of a search agent in the exploration phase according to a randomly chosen search agent instead of the best search agent found so far. The mathematical model is explained as

$$\vec{D} = |\vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X}| \quad (18)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \quad (19)$$

where $\overrightarrow{X_{rand}}$ is a random position vector (a random whale) chosen from the current population.

(ii) Exploitation Phase

Exploitation phase consists of two steps namely Shrinking encircling mechanism and Spiral updating position.

Shrinking encircling mechanism

Shrinking encircling mechanism is achieved by decreasing the value of $\vec{a}$. Note that the fluctuation range of $\vec{A}$ is also random value in the interval [−a,a] where a is decreased from 2 to 0 over the course of iterations.

Spiral updating position

A spiral equation is then created between the position of whale and prey to mimic the helix-shaped movement of whales as follows

$$\vec{X}(t+1) = \vec{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t) \tag{20}$$

$$\vec{D'} = |\vec{X^*}(t) - \vec{X}(t)| \tag{21}$$

Where $\overrightarrow{D'}$ indicates the distance of the ith whale to the prey, b is a constant for defining the shape of the logarithmic spiral, l is a random number in [−1,1], and is an element-by-element multiplication.

Whales has the probability to choose between either the shrinking encircling mechanism or the spiral model to update the position of whales during optimization. The mathematical model is as follows

$$\vec{X}(t+1) = \begin{cases} \vec{X^*}(t) - \vec{A} \cdot \vec{D} & if \ \ p < 0.5 \\ \vec{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t) & if \ \ p \geq 0.5 \end{cases} \tag{22}$$

The flowchart of Whale optimization algorithm is given in Figure 4.

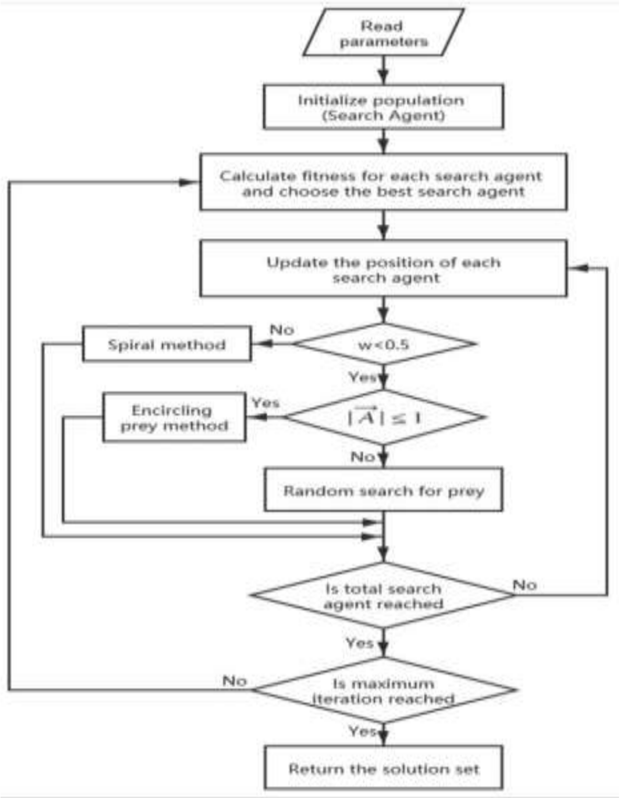

Figure 4. Whale Optimization Algorithm - Flowchart

3.5 DenseNet 121

DenseNet121 (Dense Convolutional Network) is an architecture that focuses on making the deep learning networks go even deeper, but at the same time making them more efficient to train, by using shorter connections between the layers. The DenseNet architecture proposed by Huang et al. [26] provides a rich feature representation while being computationally efficient. The primary reason is that, in each layer of the DenseNet model, the feature maps in the current layer are concatenated with those from all the preceding layers. Because fewer channels are accommodated in the convolutional layers, the number of trainable parameters is diminished, and thus, the model is computationally efficient. Furthermore, the concatenation of the feature maps from the previous layers with the current layer enhances the feature representation.

There are three kinds of blocks in the DenseNet implementation. The first is convolution block, which is a basic block of dense block. The second is dense block, in which convolution blocks are concatenated and densely connected. Dense block is the main component in DenseNet. The last is transition layer, which connects two contiguous dense blocks. Since feature map sizes are the same within the dense block, transition layer reduces the dimensions of the feature map. The technique of bottleneck design is adopted in all the blocks. The architecture of DenseNet121 is shown in Figure 5.

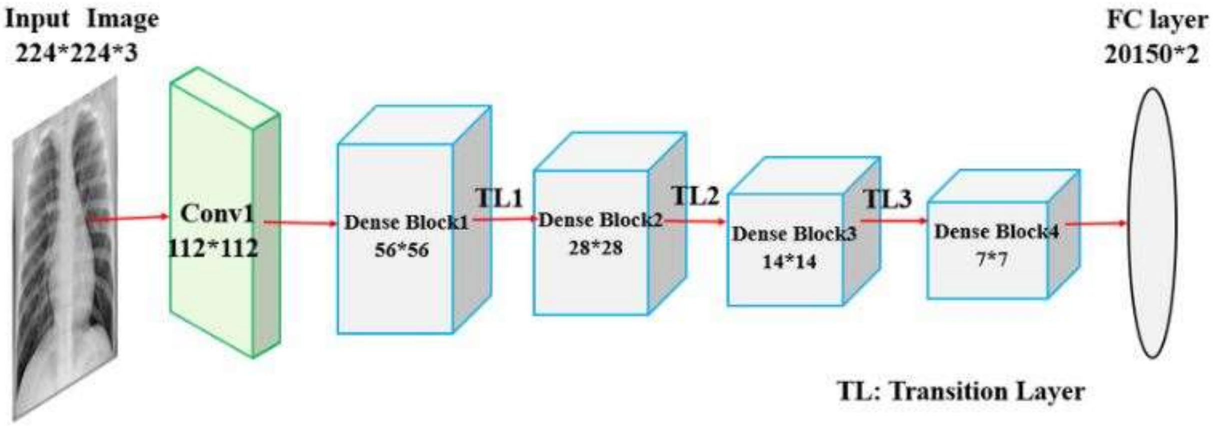

Figure 5. Architecture of DenseNet121

Fully Connected (FC) layer is the final layer which helps in the process of classification of images.

3.6 Adaptive Neuro Fuzzy Inference System

Fuzzy set theory is a powerful tool to deal with the imprecision characteristics in decision-making problems involving uncertainty and vagueness of real-world applications [27]. Fuzzy inference is a process of mapping from a given input to an output dataset using the theory of fuzzy sets. The ANFIS is a fuzzy Sugeno model put in the framework of adaptive systems to facilitate learning and adaptation [28]. Such framework makes the ANFIS modeling more systematic and less reliant on expert knowledge. To present the ANFIS architecture, two fuzzy if-then rules based on a first order Sugeno model are considered:

Rule 1: If (x is A_1) and (y is B_1) then ($f_1 = p_1x + q_1y + r_1$)
Rule 2: If (x is A_2) and (y is B_2) then ($f_2 = p_2x + q_2y + r_2$)

where x and y are the inputs, A_i and B_i are the fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule, p_i, q_i and r_i are the design parameters that are determined during the training process [29].

It is extremely important that the number of membership functions and the number of fuzzy rules hold a critical position in the designing of ANFIS. Adjusting of those parameters is highly crucial for the system because it may lead system to over-fit the data or will not be able to fit the data. This adjusting is made by a hybrid algorithm combining the least-squares method and the gradient descent method with a mean square error method. The lesser difference between ANFIS output and the actual objective means a better (more accurate) ANFIS system [30]. The architecture of ANFIS is shown in Figure 6.

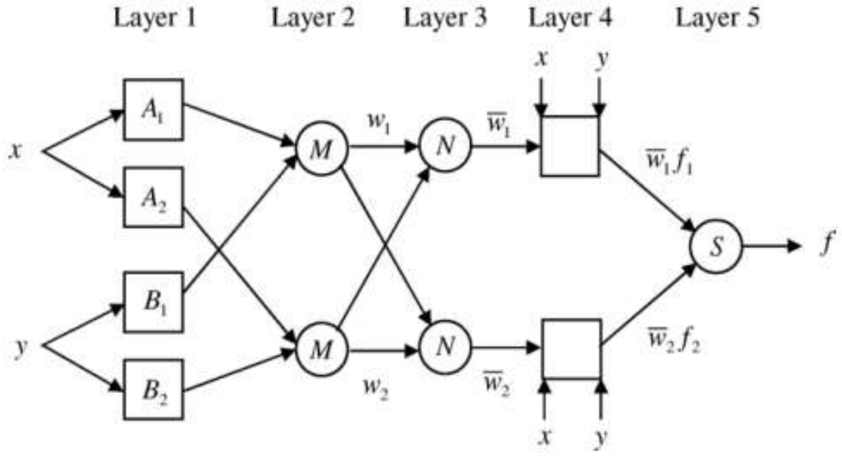

Figure 6. ANFIS Architecture

Layers of ANFIS

Layer 0: It consists of plain input variable set.

Layer 1: Each node in this layer generates a membership grade of a linguistic label. For instance, the node function of the ith node may be a generalized bell membership function

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)\right]^{b_i}} \tag{23}$$

where x is the input to node i; A_i is the linguistic label (small, large, etc.) associated with this node; and $\{a_i, b_i, c_i\}$ is the parameter set that changes the shapes of the membership function. Parameters in this layer are referred to as the premise parameters.

Layer 2: The function is a T-norm operator that performs the firing strength of the rule, e.g., fuzzy conjunctives AND & OR.

$$w_i = \mu A_i(x)\ \mu B_i(y) \tag{24}$$

Where i=1,2

Layer 3: Every node in this layer is fixed and determines a normalized firing strength. It calculates the ratio of the j^{th} rule's firing strength to the sum of all rules firing strength.

$$\overline{w}_i = \frac{w_i}{w_1 + w_2} \tag{25}$$

Where i=1,2.

Layer 4: The nodes in this layer are adaptive and are connected with the input nodes (of layer 0) and the preceding node of layer 3. The result is the weighted output of the rule j.

$$\overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i) \tag{26}$$

where w_i is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are referred to as the consequent parameters.

Layer 5: This layer consists of one single node which computes the overall output of the model.

$$\text{Overall output} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{27}$$

In this research, the convolution layers in the DenseNet121 architecture are combined with ANFIS model for the purpose of multiclass classification. The process of extracting feature is performed using convolution and pooling layers and the extracted features are sent to the fully connected layer of DenseNet121 which consists of n number of neurons. At the last stage, features obtained from neurons are given as input to the ANFIS model to classify tuberculosis along with its level. In this research, four different levels are modelled namely A,B,C and D. Level A represent as an Normal TB. Level B represents as Primary TB or Exposure includes the patients who are having 10% to 25% infection. Level C represent as Latent TB refers to the patients who are having 26% to 50% infection. Level D as represents Active TB the patients who are having 51% to 100% of tuberculosis infection respectively.

In fuzzy system, the format of this rule is given as

$$\text{IF } x \text{ is } A \text{ and } y \text{ is } B \text{ THEN } Z = f(x,y)$$

Here, AB are fuzzy sets in antecedents and $z = f(x,y)$ is a crisp function in the consequent.

4. Experimental Result Analysis and Performance Evaluation

Experiments are carried out in CXR images for the proposed Optimized Modified Dragonfly Neuro Fuzzy Inference System (OMDNFIS). Experiments are carried out using k fold cross validation where the value of k is set to 10. During experimentation, the number of epochs is set to 100. The input images are pre-processed and features are extracted from the pre-processed image using the feature extraction technique namely “First and Second Order Statistical” feature extraction. A total of 44 features are extracted using FSOS technique. The extracted features are given as input to an optimization algorithm called Modified Dragonfly Algorithm. The proposed Modified Dragonfly optimization algorithm selects 10 features out of 44 features for further processing which are shown in Table 1.

Table 1. Features selected by Modified Dragonfly Algorithm

Sl.No.	Feature Label	Features
1	m	Mean
2	Enr	Energy
3	SD	Standard Deviation
4	CRN	Correlation
5	SENT	Sum Entropy
6	SVAR	Sum Variance
7	RP	Run Percentage
8	LGRE	Low Gray-Level Run Emphasis
9	CA	Coarseness
10	TS	Texture Strength

The selected 10 features are used for classification using DenseNet121. At last, fuzzy inference system is used for better classification result.

4.1. Performance Evaluation

To evaluate the performance of the proposed OMDNFIS, performance metrics are used. The proposed model is compared with few existing models.

4.1.1. Performance Metrics

The performance of proposed method is evaluated with the existing models using performance metrics such as Accuracy, Precision, Recall and F1 score.

(i) Accuracy

Accuracy is the simplest intuitive performance metric, because it is just a ratio of accurately predicted observations and total observations.

$$Accuracy = \frac{True\ Negatives\ + True\ Positive}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \tag{28}$$

(ii) Precision

Precision is defined as the proportion of accurately predicted positive observations to the total number of expected positive observations.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{29}$$

(iii) Recall

Recall is defined as the ratio of accurately predicted positive observations to all observations in the actual class.

$$Recall = \frac{TruePositive}{True\ Positive + False\ Negative} \tag{30}$$

(iv) F1 Score

The F1 Score is the weighted average of Precision and Recall. As a result, this score accounts for both false positives and false negatives.

$$F1\ Score = 2 * \frac{precision * recall}{precision + recall} \tag{31}$$

4.1.2 Performance Analysis of proposed OMDNFIS

The performance of proposed OMDNFIS is evaluated using the performance metrics and the experimental results are discussed in this section. Features are extracted using FSOS feature extraction technique.

From Table 1, it is observed that 10 most significant features are selected by Modified Dragonfly Algorithm. The performance of proposed Modified Dragonfly Algorithm is compared with existing optimization algorithm namely Genetic Algorithm (GA), Whale Optimization Algorithm (WOA) and Dragonfly optimization algorithm (DA) [31] along with DenseNet121 in terms of performance metrics at various stages is shown in Table 2.

Table 2. Performance analysis of proposed OMDNFIS

Models	Number of features selected	Performance Metrics			
		Accuracy	Precision	Recall	F1 Score
Genetic Algorithm (GA) + DenseNet121	21	94.96	92.53	92.44	93.67
Whale Optimization Algorithm (WOA) + DenseNet121	16	96.74	94.30	94.12	94.20
Dragonfly Algorithm (DA) + DenseNet121	13	97.10	96.24	96.12	97.18
Modified Dragonfly Algorithm (MDA) + DenseNet121	10	98.58	97.66	97.42	97.66

It is inferred from Table 2 that, MDA with DenseNet121 provides an accuracy of 98.58% whereas the combination of DenseNet121 with GA, WOA and DA provides an accuracy of 94.96%, 96.74% and 97.10% respectively. The proposed MDA along with DenseNet121 provides the greater accuracy than other two optimization algorithms.

4.1.3 Performance comparison of proposed OMDNFIS and existing models

The performance of proposed OMDNFIS is compared with few existing deep learning models with and without fuzzy. The performance comparison is shown in Table 3.

Table 3. Performance comparison of proposed GNFIS and existing model

Reference	Methods	Accuracy (%)
Al-Timemy et al. [32]	ResNET-50	93.60
Kukker and sharma [33]	Modified Fuzzy Q Learning (MFQL)	91.60
Farouk Hassan et al. [34]	CNN	94.10

Gao et al. [35]	3D block based residual deep learning network	85.29
Mizan et al. [36]	MobileNet	89.90
	Xception	87.10
	Inception V3	89.90
Oh et al. [37]	Res Net-18	91.90
Heidari et al. [38]	VGC16 based CNN	94.50
Proposed OMDNFIS	Modified Dragonfly + ANFIS	98.58

It is observed from Table 3 that the proposed OMDNFIS model provides the highest accuracy of 98.58%. The proposed OMDNFIS model provides better accuracy as it combines the advantages of DenseNet121 and Adaptive Neuro Fuzzy Inference System. The second highest accuracy is provided by VGG16 based CNN model which is 94.50%. The performance of OMDNFIS is also evaluated for various number of epochs which is shown in Figure 7.

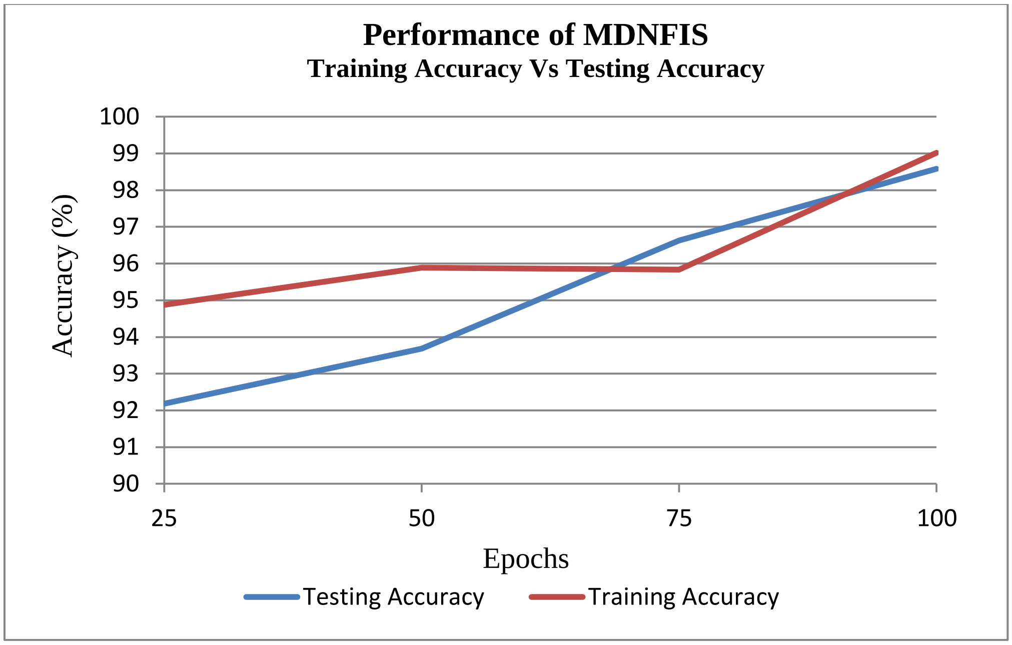

Figure 7. Performance of proposed GNFIS

Figure 7 represents the classification accuracy during training and testing phase. From the figure it is observed that the proposed OMDNFIS model provides an accuracy of 98.58% when the epoch is set to 100.

CONCLUSION

A novel Optimized Modified Dragonfly Neuro-Fuzzy Inference System is developed in this research work. The proposed OMDNFIS is the combination of Modified Dragonfly optimization, DenseNet121 and Adaptive Fuzzy Inference System (ANFIS). Modified Dragonfly optimization algorithm is utilized for efficient feature selection which selects 10 significant features out of 44 features which is extracted using FSOS feature extraction technique. MDA is also compared with GA, WOA, DA and from the results it is proved that MDA performs better for feature selection. Deep learning models are found to be best performing model for the purpose of classification. Convolutional Neural Network (CNN) is identified as the best model which contains DenseNet121 as its type. DenseNet121 model is a deep learning model which has the capability to provide best results for classification. At last, fuzzy logics are applied by utilizing Adaptive Neuro Fuzzy Inference System. The performance of the proposed OMDNFIS is analyzed and evaluated using performance metrics. The proposed OMDNFIS is also compared with few existing deep learning models and it is observed that OMDNFIS outperforms all the existing deep learning models with an accuracy of 98.58%. The proposed OMDNFIS is not only applicable for the detection of tuberculosis, it can also be used for prediction or detection of any other diseases. In future, a deep learning model can be developed to improve the classification accuracy.

REFERENCES

1. Antony, B., & NB, K. (2017). Lung tuberculosis detection using x-ray images. *International Journal of Applied Engineering Research*, *12*(24), 15196-15201.
2. Bagcchi, S. (2023). WHO's Global Tuberculosis Report 2022. *The Lancet Microbe*, *4*(1), e20.
3. Guo, R., Passi, K., & Jain, C. K. (2020). Tuberculosis diagnostics and localization in chest x-rays via deep learning models. Front. Artif. Intell, 3, 583427.
4. Forero, M. G., Sroubek, F., & Cristóbal, G. (2004). Identification of tuberculosis bacteria based on shape and color. Real-time imaging, 10(4), 251-262.
5. C.A. Robert, J.E. Buikstra, The Bio-Archaeology of Tuberculosis: A Global Perspective on a Re-Emerging Disease, University Press of Florida, Gainesville, Fl, 2009.
6. Walia, N., Singh, H., Tiwari, S. K., & Sharma, A. (2015). A decision support system for tuberculosis diagnosability. International Journal on Soft Computing, 6(3), 1-13.
7. David M. Morens, Gregory K. Folkers, Anthony S. Fauci, The challenge of emerging and re-emerging infectious diseases, Nature 430 (2004) 242–249.
8. Le, K. (2006, September). Automated detection of early lung cancer and tuberculosis based on X-ray image analysis. In Proc. WSEAS International Conference on Signal, Speech and Image Processing (pp. 1-6).
9. American Thoracic Society, Diagnostic standards and classification of tuberculosis in adults and children, Am. J. Respir. Crit. Care. Med. 161 (2000) 1376–1395.
10. Sun, S., Li, W., & Kang, Y. (2015, October). Lung nodule detection based on GA and SVM. In 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI) (pp. 96-100). IEEE.
11. Syed Tasleem, Gulraiz Ali, Farzana Mahdi SikandarHasan, Molecular diagnosis of tuberculosis: a new primer design, Iran. J. Clin. Infect. Dis. 4 (2) (2009) 105–108.

12. Asogbon, M. G., Samuel, O. W., Omisore, M. O., & Awonusi, O. (2016). Enhanced neuro-fuzzy system based on genetic algorithm for medical diagnosis. J Med Diagn Meth, 5(205), 2.
13. Young S. Kim, Donald E. Shelton, Gregg Barak, studying juror expectations for scientific evidence: a new model for looking at the CSI myth, J. Am. Judges Assoc. 47 (1–2) (2011) 8–18.
14. Papageorgiou, E. I. (2011, June). Review study on fuzzy cognitive maps and their applications during the last decade. In 2011 IEEE international conference on fuzzy systems (FUZZ-IEEE 2011) (pp. 828-835). IEEE.
15. S.A. Oke, A literature review on artificial intelligence, Int. J. Inf. Manage. Sci. 19 (4) (2008) 535–570.
16. Sharma, R., & Kochher, R. (2017, April). Fuzzy decision support system for tuberculosis detection. In 2017 International Conference on Communication and Signal Processing (ICCSP) (pp. 2001-2005). IEEE.
17. Angbera, A., Esiefarienrhe, M., & Agaji, I. (2016). Efficient fuzzy-based system for the diagnosis and treatment of tuberculosis (EFBSDTTB). International Journal of Computer Applications Technology and Research, 5(2), 34-48.
18. Tyagi, P. K., Gupta, N. K., & Gupta, S. (2016, November). Diagnosis of pulmonary tuberculosis using fuzzy inference system. In *2016 Second International Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH)* (pp. 3-7). IEEE.
19. Mirjalili, S.: Dragonfy algorithm: a new meta-heuristic optimization technique for solving single objective, discrete, and multi-objective problems. Neural Comput. Appl. 27, 1053–1073 (2016).
20. Reynolds, C.W.: Flocks, herds, and schools: a distributed behavioral model, in computer graphics.ACM SIGGRAPH Comput. Graph 21, 25–34 (1987)
21. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput. Appl. 2016, 27, 1053–1073.
22. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
23. Yang, X.-S.: Nature-inspired metaheuristic algorithms (2010)
24. Deb, K., & Agrawal, S. (1999). Understanding interactions among genetic algorithm parameters. Foundations of genetic algorithms, 5(5), 265-286.
25. Mirjalili, S. and A. Lewis, The Whale Optimization Algorithm. Advances in Engineering Software, 2016. 95: p. 51-67.
26. Huang G., Liu Z., Van Der Maaten L. & Weinberger K. (2017). Densely connected convolutional networks. Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition. pp. 4700-4708.

27. Mousavi, S. J., Ponnambalam, K., & Karray, F. (2007). Inferring operating rules for reservoir operations using fuzzy regression and ANFIS. Fuzzy sets and systems, 158(10), 1064-1082.
28. Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics, 23(3), 665-685.
29. Guler, I., & Übeyli, E. D. (2005). Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. Journal of neuroscience methods, 148(2), 113-121.
30. Karaboga, D., & Kaya, E. (2019). Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey. Artificial Intelligence Review, 52, 2263-2293.
31. Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27, 1053-1073.
32. Chandra, T. B., Verma, K., Singh, B. K., Jain, D., & Netam, S. S. (2021). Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting-based classifier ensemble. Expert systems with applications, 165, 113909.
33. Nunez-Fernández, D., Ballan, L., Jiménez-Avalos, G., Coronel, J., & Zimic, M. (2020). Using Capsule Neural Network to predict Tuberculosis in lens-free microscopic images. arXiv preprint arXiv:2007.02457.
34. Patterson, B., Koch, A., Gessner, S., Dinkele, R., Gqada, M., Bryden, W., ... & Wood, R. (2020). Bioaerosol sampling of patients with suspected pulmonary tuberculosis: a study protocol. BMC infectious diseases, 20(1), 1-7.
35. Gao, X. W., James-Reynolds, C., & Currie, E. (2020). Analysis of tuberculosis severity levels from CT pulmonary images based on enhanced residual deep learning architecture. Neurocomputing, 392, 233-244.
36. Mizan, M. B., Hasan, M. A. M., & Hassan, S. R. (2020, November). A Comparative Study of Tuberculosis Detection Using Deep Convolutional Neural Network. In 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT) (pp. 157-161). IEEE.
37. Oh, Y., Park, S., & Ye, J. C. (2020). Deep learning COVID-19 features on CXR using limited training data sets. IEEE transactions on medical imaging, 39(8), 2688-2700.
38. Heidari, M., Mirniaharikandehei, S., Khuzani, A. Z., Danala, G., Qiu, Y., & Zheng, B. (2020). Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms. International journal of medical informatics, 144, 104284.