Journal of Chemical Health Risks

www.jchr.org

JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

A Modified Moth Flame Optimization Algorithm for Multi-level Classification of COVID-19 from Tuberculosis and Pneumonia Chest X Ray images using Deep learning

***J. Jude Moses Anto Devakanth[1], Dr. R. Balasubramanian[2], Arul Suresh[3]**

[1,3] Research Scholar, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli, TamilNadu.

[2] Professor, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli, TamilNadu.

(Received: 04 August 2023 Revised: 12 September Accepted: 06 October)

KEYWORDS
Lung disease,
COVID-19,
Optimization,
Classification,
Deep learning

ABSTRACT:

Lung diseases refer to a wide range of illness conditions that affect the lungs, including pneumonia, TB, lung cancer, and numerous other respiratory issues. Recently, COVID-19 is a global pandemic infectious disease with high mortality rate which also affects lungs. The earlier lung diseases such as Pneumonia and Tuberculosis are closely related to COVID-19. All these illnesses induce severe respiratory conditions and breathing issues, which ultimately cause death. So, it is necessary to classify these diseases which helps in providing early treatment to save lives. Features plays indispensable role in classification and feature selection or optimization helps to select the most significant features. In this research, a novel optimization algorithm namely "Modified Moth Flame Optimization" (MMFO) is developed for classification of COVID-19 from Tuberculosis and Pneumonia using Chest X Ray images. The proposed MMFO address the issue of population diversity in MFO by introducing an inertia weight to balance the exploitation and global search capabilities from the perspective of diversity. The proposed MMFO algorithm is evaluated using deep learning and the experimental results are compared with state-of-the-art optimization algorithms such as Moth Flame Optimization (MFO), Grey Wolf Optimization (GWO), Crow Search Optimization Algorithm (CSA), Dragonfly Optimization Algorithm (DA), Aquila Optimizer and Whale Optimization Algorithm (WOA). Comparison results proved the superiority of the proposed MMFO algorithm.

1 Introduction

Most real-world situations involve a lot of data and so handling that data become extremely challenging and important task. A dataset has numerous features but not all the features are important and necessary. The performance of the model may be negatively impacted by some features that are unnecessary or redundant. Therefore, the primary goal is to is to lower the size of the original datasets while preserving the performance accuracy of the model [5]. Feature selection helps to reduce the number of features by selecting only the more relevant features. In order to determine the optimum set of features for feature selection issues, techniques such as greedy search, exhaustive search, and random search, etc. have been used. Premature convergence, extreme complexity, and excessive computing cost are problems that plague most approaches. As a result, metaheuristic algorithms receive a lot of attention when dealing with this kind of situation [3].

Meta heuristic algorithms are one of the optimization methods which helps to solve optimization problems by providing optimal solutions [49]. These algorithms are simple, flexible, and capable of avoiding local optima [29]. The behavior of metaheuristic algorithms is stochastic and the algorithms begin the optimization process by producing random solutions. Meta heuristic algorithms has two phases namely exploration phase and exploitation phase [23,38]. The algorithms extensively analyze the promising search space during the exploration phase and the exploitation phase involves the local search of any potential area(s) that were discovered during the exploration stage. Meta heuristic algorithms are mainly categorized into two types based on their categories namely Evolution based and Swarm intelligence-based optimization algorithms. Evolutionary algorithms primarily imitate biological evolution and reproduction concepts. Examples of these algorithms are Genetic Algorithm (GA) and Difference Evolution (DE) etc. Swarm intelligence algorithms take their cues from the cooperative behavior of various creatures including fish, birds, insects, and animals etc. It uses wrapper model for feature selection. Few popular examples of swarm intelligence algorithms are Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Cuckoo Search Algorithm (CSA), Grasshopper Optimization Algorithm (GOA), Whale Optimization Algorithm (WOA) and Moth Flame Optimization (MFO) etc. All these algorithms are utilized in various optimization problems and provides best results. It is observed from the literature survey that, Swarm intelligence algorithms are the important category and it received much attention as it is more flexible and able to provide better optimization results [25]. For solving the optimization problems, most of the researchers developed modified, improved and hybrid versions of both evolutionary [6,14,41,52] and swarm intelligence optimization algorithms [7,21,42,46]. In this research, Moth Flame Optimization algorithm is thoroughly studied and analyzed and a modified version of MFO is developed for optimization or feature selection with the capability to provide better performance than existing algorithms.

2 Related Works

Swarm intelligence algorithms [9] have acquire popularity recently because of its performance. According to the past literatures, a sizable number of novel swarm intelligence algorithms have developed in recent years [8]. Some of the existing swarm intelligence algorithms are studied and presented in this section.

Particle Swarm Optimization was developed by James Kennedy and Russell Eberhart in 1995 [21]. It is developed based on the concepts of fish schooling, swarming and bird flocking theory. This algorithm is tested by applying it for various optimization problems and as a result, it exhibits best performance. The Grey Wolf Optimization (GWO) was introduced by Seyedali Mirjalili et al. [28] which imitates the leadership quality and hunting strategy of grey wolves. For the purpose of imitating the leadership quality, four different types of wolves are utilized. Moreover, three essential components of hunting the prey namely searching, encircling, and attacking are used. The Whale Optimization Algorithm (WOA) imitates the social behavior of whales [26]. The bubble net hunting behavior served as the basis for the WOA algorithm. The Ant Lion Optimizer (ALO) algorithm imitates the natural hunting behaviour of antlions [32]. Salp Swarm Algorithm (SSA) was introduced for the purpose of optimization. The primary inspiration of SSA is the swarming character of salps [26]. Grasshopper Optimization Algorithm (GOA) imitates the basic character of grasshopper swarms in nature and it is also applied to multi-objective problems [33]. Dragonfly Optimization Algorithm (DOA) is another swarm intelligence algorithm which mimics the swarming behavior of Dragonfly. The author also developed binary and multi-objective variants of DOA namely Binary Dragonfly Algorithm (BDA) [24] and Multi-Objective Dragonfly Algorithm (MODA) respectively [30]. In order to conduct global optimization, the Bat Algorithm (BA) is developed which imitates the echolocation behaviour of bats [50]. Moreover, the binary and multi-objective variant of Bat Algorithm namely Binary Bat Algorithm (BBA) [28] and Multi-Objective Bat Algorithm [51] is introduced to solve optimization problems. A novel Gorilla Troops Optimizer (GTO) is developed by Abdollahzadeh et al. [1] which mimics the social activity of Gorilla. Prairie Dog Optimization is the recently developed optimization algorithm by Ezugwu et al. [11]. It imitates the natural

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

activity and habitat of prairie dogs. The foraging and burrow building activities of these dogs are the basic inspiration of this optimization algorithm. All these algorithms are applied using various benchmark functions to check the efficiency of optimization.

An improved or modified version of few swarm intelligence algorithms are also developed to enhance the performance of the algorithms. An improved version of PSO was introduced by incorporating each particles memories to improve the swarm diversity and two novel functions are introduced to maintain balance between the two phases of optimization namely exploration and exploitation [22]. A novel hybrid optimization algorithm that concatenates two algorithms namely PSO and Genetic Algorithm (GA). The GA operators are incorporated with the heuristics of PSO in order to improve the performance of PSO [4]. Hybridization of PSO and ACO is also developed and named as PSO/ACO [17]. Hybrid approach of PSO and GWO is also developed to solve many optimization problems [44]. An improved GWO was introduced namely I-GWO which introduces new movement technique called Dimension Learning based Hunting (DLH). This modification improves the performance of GWO [36]. Another improved version is developed by introducing chaotic tent mapping to enhance the searching ability of grey wolves [18]. A hybrid model of GWO and WOA is developed in which the spiral equation of WOA is utilized in procedures of GWO [43]. WOA is integrated with Differential Evolution (DE) [45] to improve the exploration ability of optimization algorithms [35]. Modified version of Cuckoo Search (CS) optimization algorithm by introducing a new method for determining the step size [47]. An improved model of Binary Dragonfly optimization algorithm is introduced which utilizes various strategies to update the values of the coefficients utilized in BDA [15].

3 Materials and Methodology

The outline of the research work is shown in Figure 1.

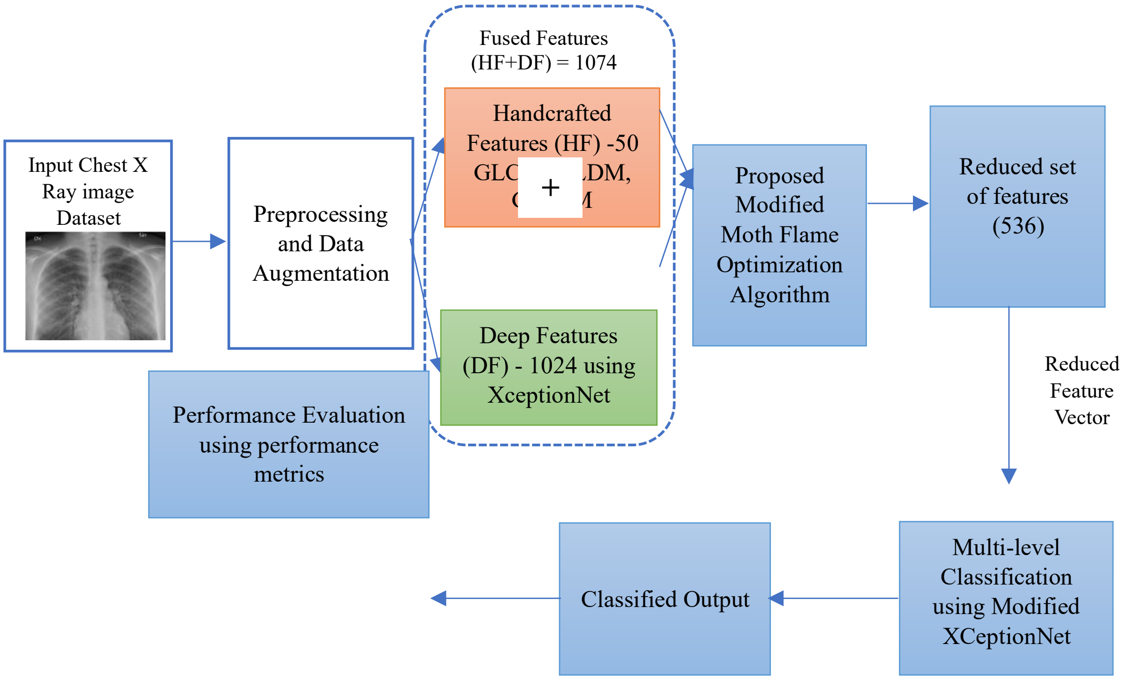


Figure 1. Outline of the Work

This section explains about the steps involved in multi-level classification of COVID-19 from Tuberculosis and Pneumonia Chest X Ray images along with dataset description. The input Chest X Ray (CXR) images are downloaded from NIH-Kaggle database [19]. Then, Preprocessing and Data augmentation is applied to the

Journal of Chemical Health Risks

www.jchr.org

JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

input CXR images for normalization and resizing. 306 features are extracted from the preprocessed image using a feature extraction technique namely "Fusion of Handcrafted and Deep features" (FHD). To reduce the number of features, the extracted features are fed into the proposed optimization technique namely "Modified Moth Flame Optimization" (MMFO) algorithm. Then, the reduced or optimized feature vector is utilized for multi-level classification of lung diseases.

3.1 Dataset Description

Chest X Ray (CXR) images are collected from NIH-Kaggle database that includes categories such as Normal, Tuberculosis, Pneumonia and COVID-19. A total of 16,000 CXR jpeg images of size 512x512 are collected. Few CXR images are shown in Figure 2.

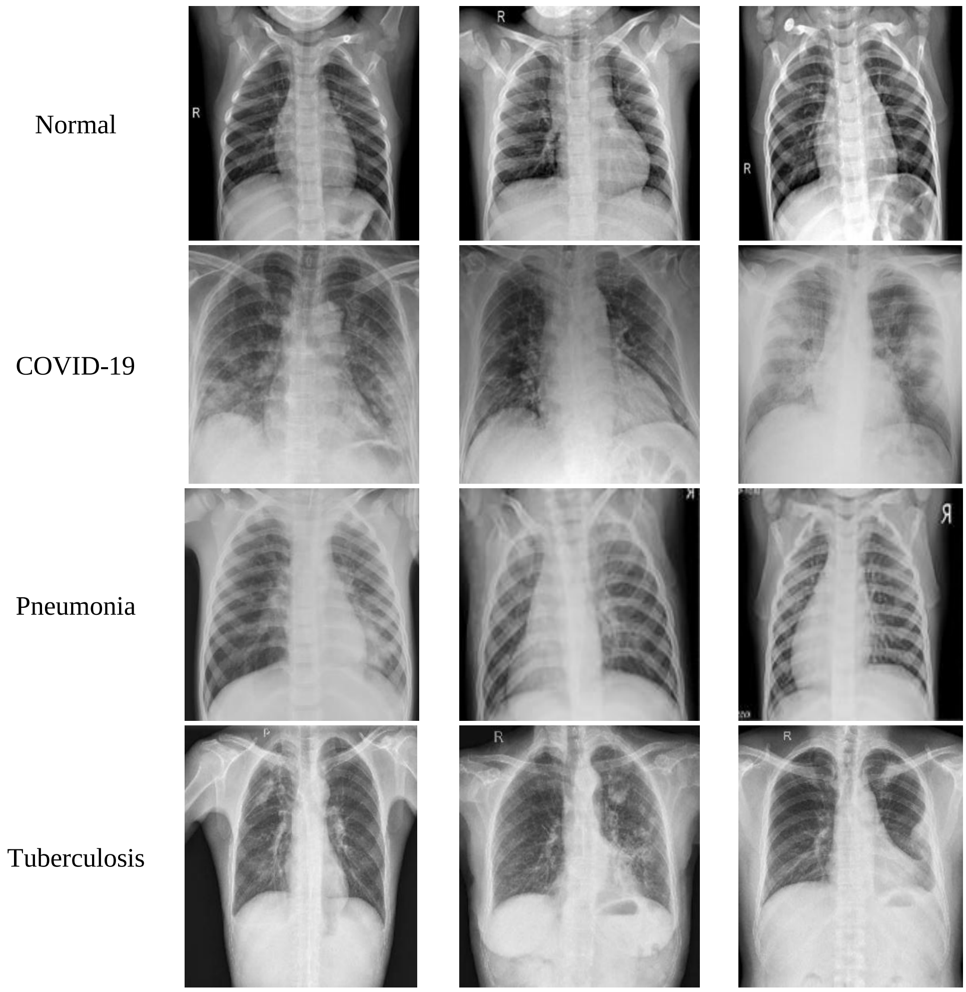


Figure 2. Sample images from the dataset

A total of 16,000 CXR images are collected from the dataset and is split into training data and testing data where 70% is for training and 30% is for testing. The details of the dataset with classes and number of images are shown in Table 1.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

Table 1. Dataset Description

Dataset	Class		No. of images in the class	No. of images for training set	No. of images for testing set
NIH (Kaggle)	Normal		2228	1560	668
	Tuberculosis		5352	3746	1606
	Pneumonia	Bacterial Pneumonia	2856	1999	857
		COVID-19	5564	3895	1669

4 Data Preprocessing and Data Augmentation

In preprocessing, the input CXR images are resized as per the standards of the XceptionNet deep learning model (299X299X3). To improve the sufficiency and diversity, data augmentation techniques are applied to the preprocessed images. Data augmentation techniques helps to improve the classification accuracy [39]. Data augmentation techniques such as Horizontal Flipping, Image rotation and Image zooming are utilized in this research.

5 Feature Extraction using Fusion of Handcrafted and Deep features (FHD) technique

The pre-processed images are fed into feature extraction process where the features are extracted using FHD feature extraction technique. This technique is the fusion of 50 Handcrafted features which are extracted manually using technique such as Gray-Level Co-occurrence Matrix (GLCM) [34], Gray Level Difference Matrix (GLDM) [48] and Gray Level Size Zone Matrix (GLSZM) [40] and 1024 deep features which are extracted automatically using Modified XceptionNet model. In handcrafted features, GLCM helps to extract 20 features whereas GLDM and GLSZM extracts 14 features and 16 features respectively.

6 Proposed Modified Moth Flame Optimization (MMFO) Algorithm

The Moth Flame Optimization (MFO) was invented by Seyedali Mirjalili [31] in 2015. Moths use the moonlight for navigation. By keeping a constant angle with the moon, moths travel along a straight line. In the presence of artificial light, they become disoriented and attempt to maintain a stable angle with it, producing a lethal spiral before collapsing to the light source and dying. This behavior of moths is taken as the inspiration and base for the Moth Flame Optimization (MFO) algorithm. A Modified Moth Flame Optimization (MMFO) is developed in this research work by adding inertia weight in standard MFO. The proposed MMFO is explained in this section.

6.1 Population initialization

At first, population must be initialized and the initial solutions are generated randomly using Eqn (1).

$$MO = R(n,d) \times (ub - lb) + lb \tag{1}$$

Where ub refers to the upper bound and lb represents the lower bound, n is the size of the population, d indicates the number of dimensions or variables and R refers to the random number between (0,1). While generating the initial population, it is assumed that all moths can fly in one dimensional, two dimensional, three dimensional and hyper dimensional space. Moths represents the search agents and the set of moths (MO) are expressed in a matrix using Eqn (2).

$$MO = \begin{bmatrix} MO_{1,1} & MO_{1,2} & \dots & \dots & MO_{1,v} \\ MO_{2,1} & MO_{2,2} & \dots & \dots & MO_{2,v} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ MO_{t,1} & MO_{t,2} & \dots & \dots & MO_{t,v} \end{bmatrix} \tag{2}$$

Where t represents the total number of moths and v refers to the number of variables or dimensions. Each moths contains corresponding fitness values and the array to store the fitness value is also expressed in a matrix MF (moths' fitness) using Eqn (3).

$$MF = \begin{bmatrix} MF_1 \\ MF_2 \\ \vdots \\ MF_t \end{bmatrix} \tag{3}$$

The next key element in MFO are Flames. The matrix for flames (FS) like moth matrix is shown in Eqn (4).

$$FS = \begin{bmatrix} FS_{1,1} & FS_{1,2} & \dots & \dots & FS_{1,v} \\ FS_{2,1} & FS_{2,2} & \dots & \dots & FS_{2,v} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ FS_{t,1} & FS_{t,2} & \dots & \dots & FS_{t,v} \end{bmatrix} \tag{4}$$

The matrix FF to store the corresponding fitness values for flames is shown in Eqn (5).

$$FF = \begin{bmatrix} FF_1 \\ FF_2 \\ \vdots \\ FF_t \end{bmatrix} \tag{5}$$

The difference between the moths and flames is the way to update them in each iteration. Moths are the search agents which moves around the search space where flames are the best solutions that obtain so far. As a result, every moth investigates the search area around flames and updates it whenever a better solution is discovered.

6.2 Position Update Mechanism

This mechanism imitates the firefighting activity of moths. In accordance with the sorted flames, the search agents or moths fly in a spiral pattern around the flames which is mathematically described in Eqn (6).

$$MF_i = S(MF_i, FS_j) \tag{6}$$

$$S(MF_i, FS_j) = D_i \times e^{kr} \times \cos(2\pi r) + FS_j \tag{7}$$

Where MF_i refers to the updated position of ith moth, S is the logarithmic spiral curve, FS_j refers to the jth flame, k is the constant and r is the random integer between [a,1], where a decreases linearly from 1 to -2 when the number of iterations increases. D_i refers to the distance between ith moth and jth flame.

$$n(moths) \leq n(flames) \tag{8}$$

In MFO, the position update mechanism is categorized into two types. The first type is described using Eqn (8).

Where n(moths) refers to the number of moths and n(flames) represents the number of flames. In the above case, each moth fly around the respective flame and updates its position. The second type of position update mechanism is described in Eqn (9).

$$n(moths) > n(flames) \tag{9}$$

As per Equation 9, if number of moths is greater, then all moths update their position based on only one flame. This results in poor diversity of the population for optimization. Population with high diversity is necessary to search for larger range in the search space. To increase the diversity and to improve the performance of optimization, Eqn (7) is modified by adding

Journal of Chemical Health Risks

www.jchr.org

JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

an inertia weight [37] to improve the diversity of search space [12] in MFO. The modified equation is utilized in the proposed MMFO and is provided below.

$$S(MF_i, FS_j) = W \times D_i \times e^{kr} \times \cos(2\pi r) + FS_j \tag{10}$$

$$W = e^{-c} \times \left(1 - \frac{T}{T_{max}}\right) \tag{11}$$

$$C = \frac{1}{N \times L} \times \sum_{i=1}^{N} \times \sqrt{\sum_{j=1}^{dim} \left(M_{i,j}{}^{T} - \bar{M}_j{}^{T}\right)^2} \tag{12}$$

Where W is the inertia weight, T indicates the current iteration and T_{max} refers to the maximum iteration. The value of C is determined by the difference in positions of the moths in different dimensions and e refers to exponential. In Eqn (12), N refers to the population number, dim represents dimension, $M_{i,j}{}^{T}$ refers to i^{th} moth in j^{th} dimension at iteration T and $\bar{M}_j{}^{T}$ indicates the mean value of moths in j dimension at T^{th} iteration. L is described as maximum diagonal distance of the search space and is expressed using Eqn (13).

$$L = \sqrt{\sum_{j}^{dim} \left(ub_j - lb_j\right)^2} \tag{13}$$

Where ub_j indicates the upper bound and lb_j refers to the lower bound of the j^{th} dimension in the search space. D_i refers to the distance from the i^{th} moth to the j^{th} flame which is calculated using Eqn (14).

$$D_i = \left|FS_j - MF_i\right| \tag{14}$$

To explore the optimal solution and to increase the convergence rate of the algorithm, an adaptive reduction mechanism for the number of flames is adopted in traditional MFO algorithm and is expressed in Eqn (15).

$$FN = round(M_F - i_c * \frac{M_F - i_c}{i_m}) \tag{15}$$

Where FN refers to the Flame Number, M_F represents the maximum number of flames, current number of iterations and maximum number of iterations are represented by i_c and i_m respectively. The addition of inertia weight W in the proposed MMFO leads to increase in population diversity and expansion of search space. Moreover, there is a chance for the search agents to jump out of the trap to search for global optimal solutions even the solutions fall into local optima because of the greater search space. Therefore, in later iterations, the value of W decreases because of the gradual approach of $1 - \frac{T}{T_{max}}$ to zero. At this time, significantly a greater number of moths can identify the best flame and is mainly based on the modifications carried out in the proposed MMFO which is expressed in Eqn (10).

The Pseudocode of the proposed MMFO is provided below.

Journal of Chemical Health Risks

www.jchr.org

JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

Pseudocode of the proposed method

```
Initialize the parameters for moth flame
Initialize Moth's position MF_i randomly
    for i=1 to t do
      Calculate the fitness function FF_i
    end for
while T ≤ T_max
    Update the position of MF_i
    Calculate the number of flames using Equation (15)
    Evaluate the fitness function FF_i
    if T==1 then
      FS=sort (MO) and FF=sort (MF)
   else
     FS=sort(MF_{n-1}, MF_n) and FF= sort(MF_{n-1}, MF_n)
  end if
  for i=1 to t do
    for j=1 to v do
      Update the values of k and r
      Calculate the value of D_i respect to its corresponding moth using Equation (14)
        if n(moths)<n(flames)
          Update the position of moth using Equation (7)
        else
          Update the position of moth using Equation (10)
        end if
      end for
   end for
end while
Return the best solution FS[0]
```

7 Multi level Classification

In this research, multi-level classification comprises two phases namely first level classification and second level classification. The first level represents triclass classification which contains three classes namely Normal, Tuberculosis and Pneumonia. The second level classification refers to the binary classification which has two classes namely Bacterial Pneumonia and COVID-19. The CXR images which are identified as Pneumonia in the first level is taken as the input for the second level classification. For classification, Modified XceptionNet [10] is utilized which contains an additional convolutional layer and max pooling layer. Multi-level classification is shown in Figure 3.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

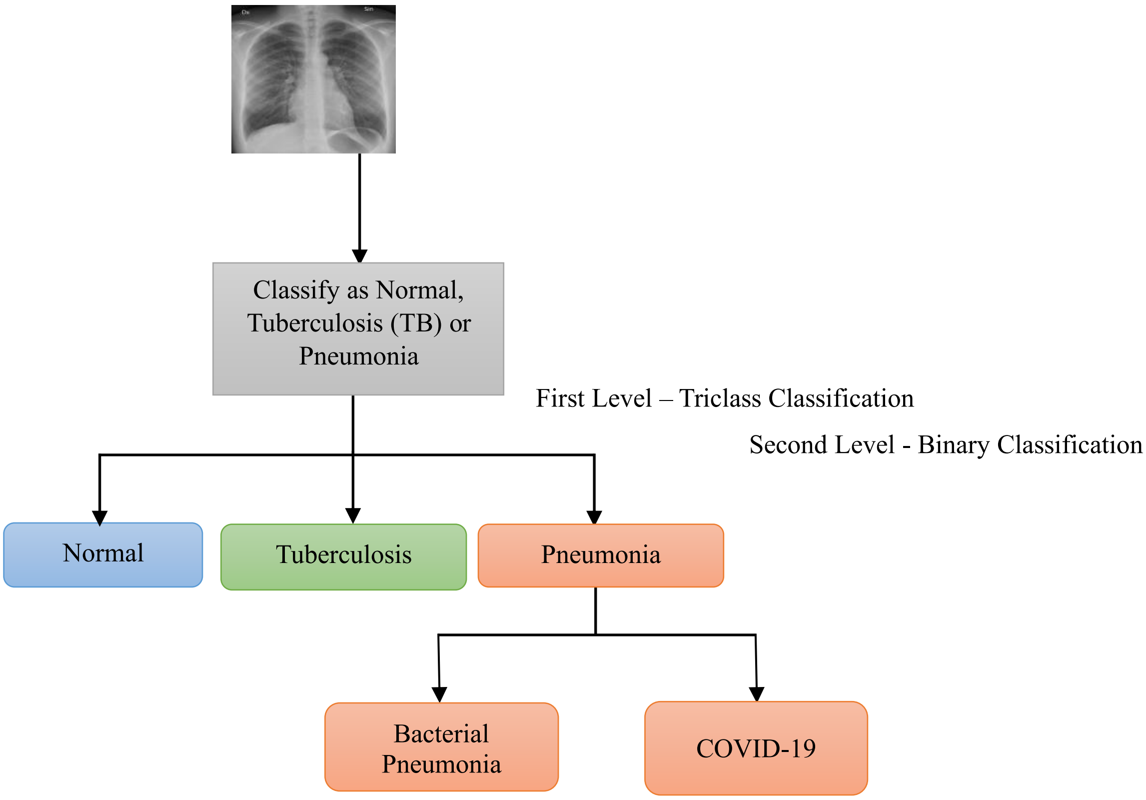


Figure 3. Multiclass classification

Multiclass classification is performed using Modified XceptionNet and the architecture of Modified XceptionNet utilized in this research work is shown in Figure 4.

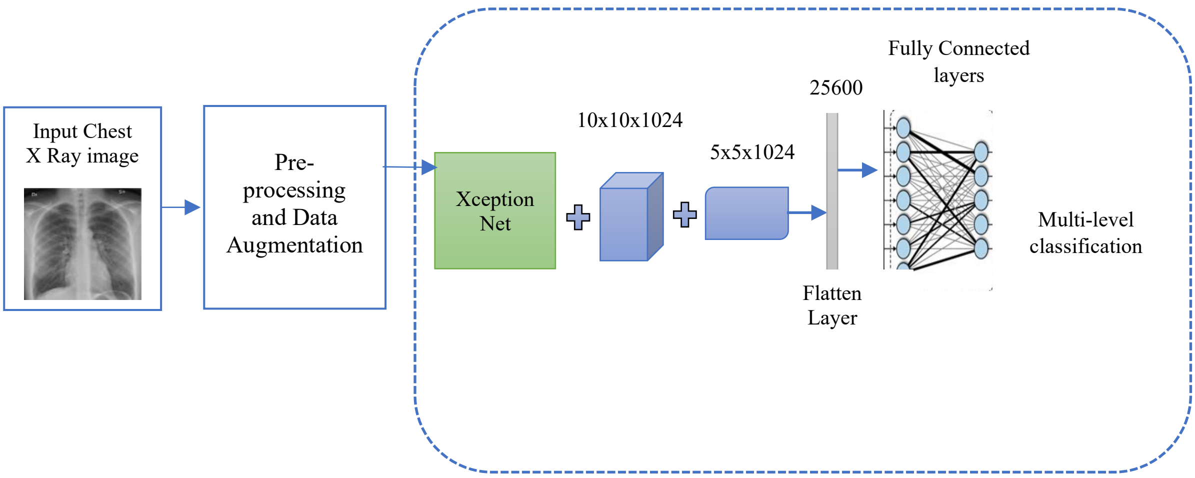


Figure 4. Modified XceptionNet based Multi-level Classification

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

The parameters utilized in Modified XceptionNet is listed in Table 2.

Table 2. Parameters for classification using Modified XceptionNet

Parameters	Value
Image Shape	299X299X3
Data Augmentation	RandAugment
Base Model	Modified XceptionNet
Classifier	Softmax
Optimizer	Nadam
Loss function	Categorical Crossentropy
Class	Triclass and Binary class
Learning rate	0.01
Dropout	0.5
Epoch	25 to 100
Batch Size	32

It is inferred from Table 2 that Categorical Cross entropy loss function and Nadam optimizer is used to train the modified XceptionNet. To optimize the training process of modified XceptionNet, Nadam optimizer is used. Nadam helps to increase the accuracy and minimize the training cost of the neural network. Modified XceptionNet is trained with 100 epochs in each training phase. The learning rate and the batch size was set to 0.01 and 32 respectively.

8 Experimental Result Analysis and Performance Evaluation

Experiments are carried out using CXR images to verify the efficiency of proposed Modified Moth Flame Optimization (MMFO) algorithm. The input images are pre-processed and features are extracted from the pre-processed image using the feature extraction technique namely "Fusion of Handcrafted and Deep features" (FHD). A total of 1074 features are extracted using FHD feature extraction technique which includes 50 handcrafted features and 1024 deep features. The extracted features are given as input to the proposed MMFO algorithm. The proposed MMFO optimization algorithm selects 536 features which includes 10 handcrafted and 526 deep features for multi-level classification. The selected 10 handcrafted features are listed in Table 2.

Table 2. List of selected Handcrafted features

Sl.No.	Label	Features	Formula
1	ENT	Entropy	$ENT = -\sum_{m=1}^{C_g}\left\{\sum_{n=1}^{C_g}\{P_{m,n}.ln[P_{m,n}]\}\right\}$
2	SAVE	Sum Average	$SAVE = \sum_{k=2}^{2-C_g} k.P_{i+j}\,(k)$
3	MP	Maximum Probability	$MP = \underset{m,n}{max}\,\{P_{m,n}\}$
4	GLN	Gray level Non-uniformity	$GLN = \frac{\sum_{m=1}^{D_g}\left(\sum_{n=1}^{D_d} P(m,n)\right)^2}{D_z}$

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

5	DV	Dependence Variance	$DV = \sum_{m=1}^{D_g}\sum_{n=1}^{D_d} P(m,n)\,(n - \left(\sum_{m=1}^{D_g}\sum_{n=1}^{D_d} nP(m,n)\right)$
6	HGLE	High Gray Level Emphasis	$HGLE = \frac{\sum_{m=1}^{D_g}\sum_{n=1}^{D_d} P(m,n)m^2}{D_z}$
7	LAE	Large Area Emphasis	$LAE = \frac{\sum_{m=1}^{S_g}\sum_{n=1}^{S_s} P(m,n)n^2}{S_z}$
8	GLV	Gray Level Variance	$GLV = \sum_{m=1}^{S_g}\sum_{n=1}^{S_s} p(m,n)(m - \left(\sum_{m=1}^{S_g}\sum_{n=1}^{S_s} p(m,n)m\right))^2$
9	ZP	Zone Percentage	$ZP = \frac{N_z}{N_p}$
10	LALGLE	Large Area Low Gray Level Emphasis	$LALGLE = \frac{\sum_{m=1}^{S_g}\sum_{n=1}^{S_s} \frac{P(m,n)n^2}{m^2}}{S_z}$

In Table 2, Features ENT, SAVE and MP are the features extracted using GLCM. In GLCM, $P_{m,n}$ refers to the GLCM matrix where m and n are the gray level values of the image respectively. C_g represents the number of discrete gray levels in the image. Then, the features GLN, DV and HGLE are the features extracted using GLDM. In GLDM, $P_{m,n}$ refers to the dependence matrix, p(m,n) represents the normalized dependence matrix and $(m,n)^{th}$ element describes the frequency of voxels with gray level *m* dependent voxels *n* in its neighbourhood present in the image. Finally, the features LAE, GLV, ZP and LALGLE are extracted using GLSZM. In GLSZM, $P_{m,n}$ refers to the size zone matrix, p(m,n) is the normalized size zone matrix and $(m,n)^{th}$ element refers to the amount of zones which has gray level m and size n in the image. In GLCM,

$$P_{i+j}(k) = \sum_{m=1}^{C_g}\sum_{n=1}^{C_g} P_{mn} \quad \text{where m+n=k, k=2,3,.........}2C_g \tag{16}$$

In GLDM, the input image is labelled as D. Let D_g refers to the number of discrete intensity values in D, D_d represents the discrete dependency sizes in D and D_z refers to the amount of dependency zones in D. Number of dependency zones (D_Z) in the input image (D) is calculated using following Equations.

$$p(m,n) = \frac{P(m,n)}{D_z} \tag{17}$$

$$D_Z = \sum_{m=1}^{D_g}\sum_{n=1}^{D_d} P(m,n) \tag{18}$$

In GLSZM, Consider S as the input image. Let S_g be the number of discrete intensity values in S, S_s be the number of discrete zone sizes in S and S_z refers to the number of zones in S.

$$p(m,n) = \frac{P(m,n)}{S_z} \tag{19}$$

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

$$S_Z = \sum_{m=1}^{S_g} \sum_{n=1}^{S_s} P(m,n) \tag{20}$$

8.2 Performance Evaluation

To evaluate the performance of the proposed MMFO along with Modified XceptionNet, performance metrics are used. Using the performance metrics, the proposed MMFO is compared with few existing optimization algorithms.

8.1.1 Performance Metrics

The performance metrics used in this research are Accuracy, Precision, Recall, F1 score Kappa and Error rate. The formula to calculate the performance metrics is explained in this section where TP refers to True Positive, TN denotes True Negative, FP represents False Positive and FN indicates False Negative.

(i) Accuracy

Accuracy is the simplest but significant performance metric which is just a ratio of accurately predicted observations to total observations.

$$Accuracy = \frac{TN + TP}{TP + FP + TN + FN} \tag{21}$$

(ii) Precision

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

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

(iii) Recall

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

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

(iv) Specificity

Specificity is defined as the ratio of negative observations to all observations in the actual class.

$$Specificity = \frac{TN}{TN + FP} \tag{24}$$

(v) 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.

$$F1\ Score = 2 * \frac{precision * recall}{precision + recall} \tag{25}$$

(vi) Kappa

Kappa is a statistical metric which can be computed using accuracy and the formula is given below.

$$Kappa = \frac{Accuracy - Expected\ Accuracy}{1 - Expected\ Accuracy} \tag{26}$$

Where the Expected Accuracy can be computed using the following formula

$$Expected\ Accuracy = \frac{\big((TP + FN) * (TP + FP)\big) + ((FP + TN) * (FN + TN))}{(TP + TN + FP + FN)^2} \tag{27}$$

(vii) Error Rate

Error rate represents the number of observations that are misclassified which is computed using the following Equation.

$$Error\ Rate = 1 - \frac{TP + TN}{TP + FP + TN + FN} \tag{28}$$

8.1.2 Performance Analysis of proposed MMFO

The performance of proposed MMFO is evaluated using the performance metrics and the experimental results are discussed in this section. The performance of feature selection greatly depends on the fitness functions or objective functions used. There are 23 standard benchmark functions (f_1 to f_{23}) [27,28] that can be used as objective function for optimization. The performance of 23 objective functions with the proposed MMFO is evaluated using Modified XceptionNet classifier and the accuracy is shown in Table 3. In Table 3, functions f_1 to f_7 refers to Unimodal benchmark functions which contains only one global optima, f_8 to f_{13} are called multimodal benchmark functions which contains many local optima locations but the true global best solution is one and f_{14} to f_{23} are called as fixed dimensional multimodal benchmark functions with minimum and fixed search space and dimensions. From the experimental results, it is identified that the proposed MMFO works efficient when the objective function f_8 is used. f_8 selects 536 features from 1074 features and provides an accuracy of 97.62%. The objective function f_8 is used in this research work for optimization which is shown in Equation (29). [28].

$$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{|x_i|}\right) \tag{29}$$

Where (x_i) represents the search agent's position and n refers to the population size.

Table 3. Performance of various objective functions using proposed MMFO and Modified XceptionNet

Sl. No.	Function	Range	No. of Features selected	Accuracy (%)

1	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	548	94.27
2	$f_2(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	[-10,10]	542	95.00
3	$f_3(x) = \sum_{i=1}^{n}(\sum_{j-1}^{i} x_j{}^2)$	[-100,100]	612	90.28
4	$f_4(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	[-100,100]	600	90.68
5	$f_5(x) = \sum_{i=1}^{n-1}[100\,(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	540	93.78
6	$f_6(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	[-100,100]	592	88.00
7	$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random\,[0,1]$	[1.28,1.28]	610	89.80
8	$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	[-500,500]	536	97.62
9	$f_9(x) = \sum_{i=1}^{n}[x_i^2 - 10\,\cos(2\pi x_i) + 10]$	[-5.12,5.12]	600	91.10
10	$f_{10}(x) - 20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	[-32,32]	688	88.16
11	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i{}^2 - \prod_{i=1}^{n}\cos\left(\frac{x}{\sqrt{i}}\right) + 1$	[-600,600]	549	96.28
12	$f_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$	[-50,50]	692	88.20
13	$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	[-50,50]	558	95.18
14	$f_{14}(x) = -\sum_{i=1}^{n}\sin(x_i).\left(\sin\left(\frac{ix_i^2}{\pi}\right)\right)^{2m},\ m = 10$	[0, π]	812	80.26
15	$f_{15}(x) = \left[e^{-\sum_{i=1}^{n}(x_{i/\beta})^{2m}} - 2e^{-\sum_{i=1}^{n} x_i^2}\right].\prod_{i=1}^{n} cos^2 x_i,\ m = 5$	[-20,20]	562	94.00
16	$f_{16}(x) = \{[\sum_{i=1}^{n}\sin^2(x_i)] - \exp(-\sum_{i=1}^{n} x_i^2)\}.exp[-\sum_{i=1}^{n} sin^2\sqrt{\lvert x_i\rvert}]$	[-10,10]	550	95.68
17	$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	[-5,5]	622	90.88
18	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27\,x_2^2)]$	[-2,2]	698	86.20
19	$f_{19}(x) = -\sum_{i=1}^{4} c_i\exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	[1,3]	548	91.20
20	$f_{20}(x) = -\sum_{i=1}^{4} c_i\exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	[0,1]	590	89.56
21	$f_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	648	88.40
22	$f_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	538	92.20
23	$f_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	644	85.00

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

In Table 3, Ranges refers to the Upper bound and Lower bound of the function's search space.

A total of 536 features are selected as significant features by the proposed MMFO. The experimental results of the proposed MMFO and Modified XceptionNet classifier for multi-level classification is shown in Table 4.

Table 4. Experimental results of proposed MMFO with Modified XceptionNet classifier

Classification	Number of features	Accuracy	Precision	Recall	Specificity	F-measure	Kappa	Error Rate
First Level classification (Triclass classification) – (Normal, Tuberculosis, Pneumonia)	536	98.26	94.08	94.61	96.99	94.34	89.88	1.74
Second Level classification (Binary classification) – (Bacterial Pneumonia, Viral Pneumonia)	536	96.98	96.88	97.10	97.86	96.90	90.92	3.02

From Table 4, it is evident that the proposed MMFO performs well with Modified XceptionNet classifier by providing an average classification accuracy of 97.62%. The proposed MMFO is compared with few existing optimization algorithms namely Grey Wolf Optimization (GWO) Algorithm, Whale Optimization Algorithm (WOA), Crow Search Optimization (CSA), Dragonfly Optimization algorithm (DA) and Moth Flame Optimization (MFO) algorithm along with Modified XceptionNet. The experimental results are shown in Table 5.

Table 5. Performance analysis of proposed and existing optimization algorithms with Modified XceptionNet

Models	Number of features selected	Performance Metrics						
		Accuracy	Precision	Recall	Specificity	F1 Score	Kappa	Error Rate
Grey Wolf Optimization (GWO) Algorithm	722	88.40	86.82	89.60	88.68	88.80	80.26	11.60
Whale Optimization Algorithm (WOA)	686	92.40	89.92	90.14	92.96	92.46	86.40	7.60
Crow Search Optimization Algorithm (CSA) [20]	628	89.52	88.75	80.00	89.90	88.00	80.88	10.48

Dragonfly Optimization Algorithm (DA)	560	94.26	92.88	93.10	94.98	94.38	86.92	5.74
Aquila Optimizer (AQU) [2]	552	95.40	93.29	92.34	95.82	95.66	87.64	4.60
Moth Flame Optimization (MFO) Algorithm	548	96.04	93.65	94.59	96.78	96.12	88.42	3.96
Proposed Modified Moth Flame Optimization (MMFO) Algorithm	536	97.62	95.80	96.22	97.82	97.68	89.88	2.38

It is inferred from Table 3 that, the proposed MMFO with Modified XceptionNet provides an accuracy of 97.62%. The second highest accuracy is provided by Moth Flame Optimization (MFO) algorithm. The proposed MMFO provides better accuracy as the addition of inertia weight improve the population diversity which in turn helps to identify the best solutions. The accuracy comparison of MFO and the proposed MMFO is shown in Figure 5.

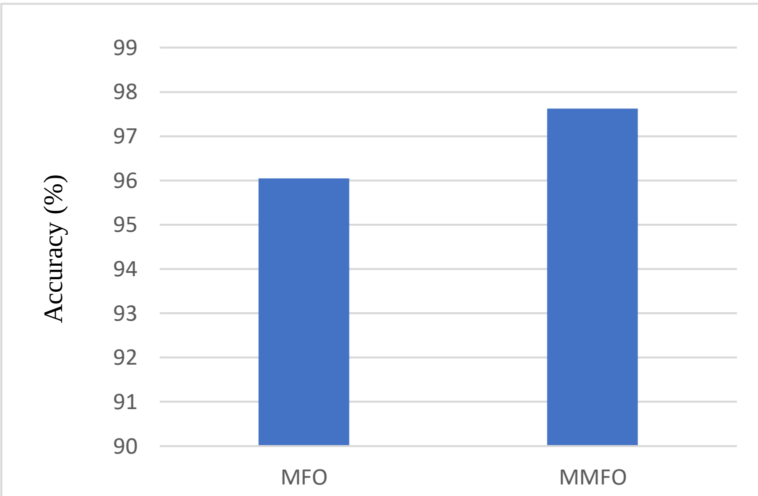


Figure 5. Accuracy comparison of MFO and MMFO

It is observed from Figure 5 that, the proposed MMFO performs better than the actual MFO. The performance of the proposed MMFO with Modified XceptionNet is also evaluated for various number of epochs and the variations in training and testing accuracy is shown in Figure 6.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727

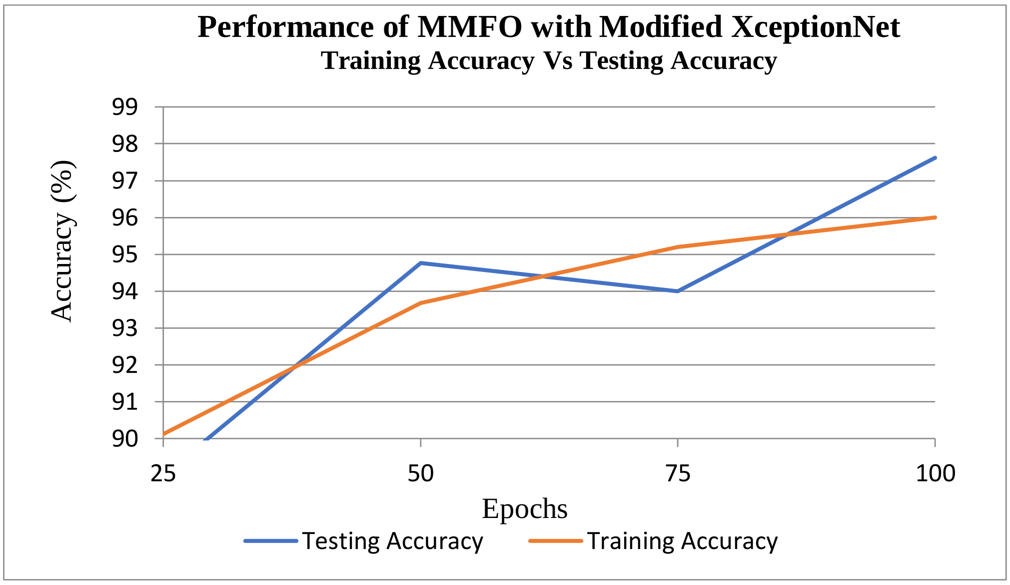


Figure 6. Performance of proposed MMFO with Modified XceptionNet

The classification accuracy during training and testing phase is shown in Figure 6. From the figure, it is observed that the proposed MMFO provides an accuracy of 97.62% with the Modified XceptionNet when the epoch is set to 100.

8.1.3 Grad-CAM Visualization

Gradient-weighted Class Activation Mapping (Grad-CAM) [16] is a concept that helps for visual validation of a deep learning model. It employs gradients from any target concept, flowing into the final convolutional layer to create a coarse localization map that emphasises key areas in the image for prediction or classification. In this research, Global Average Pooling (GAP) convolutional feature map is utilized and are fed directly into the SoftMax layer of Modified XceptionNet. The Grad-CAM $V^{c}{}_{G-CAM} \in R^{i*j}$ in deep convolutional neural networks and after training its feature mappings $\propto$ are utilized to compute the layer's gradient of g^c. Weights δ_k^c are computed using global average pooled interpretations of these gradients which is shown in Equation (30).

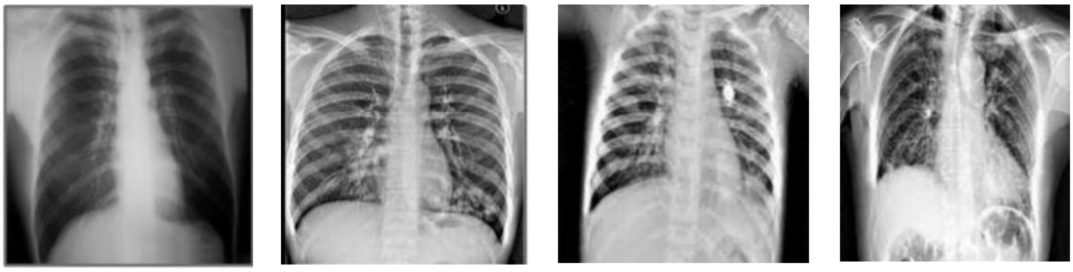
$$\delta_k^c = \frac{1}{z}\sum_m \sum_n \frac{\beta g^c}{\beta \alpha_{mn}^k} \tag{30}$$

Where δ_k^c represents weights which defines feature map k for class c and acts as a partial linearization of deep network downstream of α and Z refers to the number of pixels in the feature map where m and n are the pixels. The Grad-CAM generates heat map which is normalized for visualization. The generated heatmap is a weighted combination of feature maps and is refined using ReLU:

$$V^{c}{}_{G-CAM} = RELU(\sum_k \delta_k^c \propto^k) \tag{31}$$

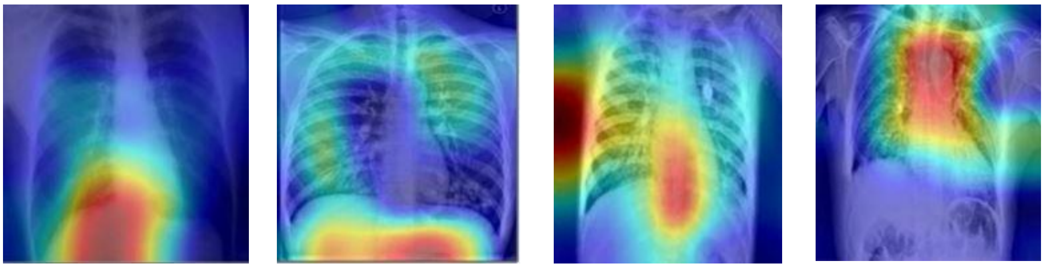
The original CXR images and the corresponding heatmap are shown in Figure 7 and Figure 8 respectively.

Journal of Chemical Health Risks
www.jchr.org
JCHR (2023) 13(3), 1177-1197 | ISSN:2251-6727



Tuberculosis | Pneumonia | Bacterial Pneumonia | COVID-19

Figure 7. Original CXR images



Tuberculosis | Pneumonia | Bacterial Pneumonia | COVID-19

Figure 8. Results of Grad-Cam

The heatmap used to identify the infected area of the lungs in CXR images which helps in further classification of lung diseases.

Conclusion

In this work, a modified version of Moth Flame Optimization (MFO) algorithm is developed and named as "Modified Moth Flame Optimization" (MMFO). The main concept behind MMFO is to update the position of moths in a different way by adding an inertia weight to increase the population diversity. The proposed MMFO selects 536 features from 1074 features and is evaluated using Modified XceptionNet. Modified XceptionNet provides a classification accuracy of 97.62% for selected 536 features whereas the classification accuracy is 96.2% when the number of features is 1074. The proposed MMFO is also evaluated using 23 benchmark objective functions and it is identified that the function f_8 performs well for optimization. Therefore, f_8 is used as fitness function in the proposed MMFO. The proposed MMFO is also compared with few existing optimization algorithms such as Moth Flame Optimization (MFO), Grey Wolf Optimization (GWO), Crow Search Optimization Algorithm (CSA), Dragonfly Optimization Algorithm (DA), Aquila Optimizer and Whale Optimization Algorithm (WOA). From the experimental results, it is identified that the proposed MMFO provides greater accuracy of 97.62% than the existing optimization algorithms. As a result, it is proved that the proposed modified version of MFO performs well for optimization than the original MFO.

References

1. Abdollahzadeh, Benyamin, Farhad Soleimanian Gharehchopogh, and Seyedali Mirjalili. "Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems." International Journal of Intelligent Systems 36, no. 10 (2021): 5887-5958.
2. Abualigah, Laith, Dalia Yousri, Mohamed Abd Elaziz, Ahmed A. Ewees, Mohammed AA Al-Qaness, and Amir H. Gandomi. "Aquila optimizer: a novel meta-heuristic optimization algorithm." Computers & Industrial Engineering 157 (2021).
3. Agrawal, Prachi, Hattan F. Abutarboush, Talari Ganesh, and Ali Wagdy Mohamed. "Metaheuristic algorithms on feature selection: A survey of one

decade of research (2009-2019)." Ieee Access 9 (2021): 26766-26791.
4. Barroso, Elias Saraiva, Evandro Parente, and Antonio Macario Cartaxo de Melo. "A hybrid PSO-GA algorithm for optimization of laminated composites." Structural and Multidisciplinary Optimization 55 (2017): 2111-2130.
5. Bashir, Saba, Irfan Ullah Khattak, Aihab Khan, Farhan Hassan Khan, Abdullah Gani, and Muhammad Shiraz. "A novel feature selection method for classification of medical data using filters, wrappers, and embedded approaches." Complexity (2022).
6. Benitez, Ian P., Ariel M. Sison, and Ruji P. Medina. "An improved genetic algorithm for feature selection in the classification of disaster-related Twitter messages." In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 238-243. IEEE, 2018.
7. Bhattacharjee, Kaushik Kumar, and Sarada Prasad Sarmah. "Modified swarm intelligence based techniques for the knapsack problem." Applied Intelligence 46 (2017): 158-179.
8. Brezocnik, Lucija, Iztok Fister Jr, and Vili Podgorelec. "Swarm intelligence algorithms for feature selection: a review." Applied Sciences 8, no. 9 (2018).
9. Chakraborty, Amrita, and Arpan Kumar Kar. "Swarm intelligence: A review of algorithms." Nature-inspired computing and optimization: Theory and applications (2017): 475-494.
10. El Gannour, Oussama, Soufiane Hamida, Bouchaib Cherradi, Mohammed Al-Sarem, Abdelhadi Raihani, Faisal Saeed, and Mohammed Hadwan. "Concatenation of pre-trained convolutional neural networks for enhanced covid-19 screening using transfer learning technique." Electronics 11, no. 1 (2021).
11. Ezugwu, Absalom E., Jeffrey O. Agushaka, Laith Abualigah, Seyedali Mirjalili, and Amir H. Gandomi. "Prairie dog optimization algorithm." Neural Computing and Applications 34, no. 22 (2022): 20017-20065.
12. Fang, W., J. Sun, and W. B. Xu. "Diversity-controlled particle swarm optimization algorithm." Control and Decision 23, no. 8 (2008): 863-868.
13. Guedria, Najeh Ben. "Improved accelerated PSO algorithm for mechanical engineering optimization problems." Applied Soft Computing 40 (2016): 455-467.
14. Guo, Pengfei, Xuezhi Wang, and Yingshi Han. "The enhanced genetic algorithms for the optimization design." In 2010 3rd international conference on biomedical engineering and informatics, vol. 7, pp. 2990-2994. IEEE, 2010.
15. Hammouri, Abdelaziz I., Majdi Mafarja, Mohammed Azmi Al-Betar, Mohammed A. Awadallah, and Iyad Abu-Doush. "An improved dragonfly algorithm for feature selection." Knowledge-Based Systems 203 (2020).
16. Hamza, Ameer, Muhammad Attique Khan, Shui-Hua Wang, Majed Alhaisoni, Meshal Alharbi, Hany S. Hussein, Hammam Alshazly, Ye Jin Kim, and Jaehyuk Cha. "COVID-19 classification using chest X-ray images based on fusion-assisted deep Bayesian optimization and Grad-CAM visualization." Frontiers in Public Health 10 (2022).
17. Holden, Nicholas Paul, and Alex A. Freitas. "A hybrid PSO/ACO algorithm for classification." In Proceedings of the 9th annual conference companion on Genetic and evolutionary computation, pp. 2745-2750. 2007.
18. Hou, Yuxiang, Huanbing Gao, Zijian Wang, and Chuansheng Du. "Improved grey wolf optimization algorithm and application." Sensors 22, no. 10 (2022).
19. https://www.kaggle.com/datasets/nih-chest-xrays/data
20. Hussien, Abdelazim G., Mohamed Amin, Mingjing Wang, Guoxi Liang, Ahmed Alsanad, Abdu Gumaei, and Huiling Chen. "Crow search algorithm: theory, recent advances, and applications." IEEE Access 8 (2020): 173548-173565.
21. Kennedy, James, and Russell Eberhart. "Particle swarm optimization." In Proceedings of ICNN'95-international conference on neural networks, vol. 4, pp. 1942-1948. IEEE, 1995.
22. Khadanga, Rajendra Kumar, Amit Kumar, and Sidhartha Panda. "A modified Grey Wolf Optimization with Cuckoo Search Algorithm for load frequency controller design of hybrid power system." Applied soft computing 124 (2022).

23. Lin, Lin, and Mitsuo Gen. "Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation." Soft Computing 13 (2009): 157-168.
24. Mafarja, Majdi, Ibrahim Aljarah, Ali Asghar Heidari, Hossam Faris, Philippe Fournier-Viger, Xiaodong Li, and Seyedali Mirjalili. "Binary dragonfly optimization for feature selection using time-varying transfer functions." Knowledge-Based Systems 161 (2018): 185-204.
25. Mavrovouniotis, Michalis, Changhe Li, and Shengxiang Yang. "A survey of swarm intelligence for dynamic optimization: Algorithms and applications." Swarm and Evolutionary Computation 33 (2017): 1-17.
26. Mirjalili, Seyedali, Amir H. Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems." Advances in engineering software 114 (2017): 163-191.
27. Mirjalili, Seyedali, and Andrew Lewis. "The whale optimization algorithm." Advances in engineering software 95 (2016): 51-67.
28. Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." Advances in engineering software 69 (2014): 46-61.
29. Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Xin-She Yang. "Binary bat algorithm." Neural Computing and Applications 25 (2014): 663-681.
30. Mirjalili, Seyedali. "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems." Neural computing and applications 27 (2016): 1053-1073.
31. Mirjalili, Seyedali. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm." Knowledge-based systems 89 (2015): 228-249.
32. Mirjalili, Seyedali. "The ant lion optimizer." Advances in engineering software 83 (2015): 80-98.
33. Mirjalili, Seyedeh Zahra, Seyedali Mirjalili, Shahrzad Saremi, Hossam Faris, and Ibrahim Aljarah. "Grasshopper optimization algorithm for multi-objective optimization problems." Applied Intelligence 48 (2018): 805-820.
34. Mohanaiah, P., P. Sathyanarayana, and L. GuruKumar. "Image texture feature extraction using GLCM approach." International journal of scientific and research publications 3, no. 5 (2013): 1-5.
35. Mostafa Bozorgi, Seyed, and Samaneh Yazdani. "IWOA: An improved whale optimization algorithm for optimization problems." Journal of Computational Design and Engineering 6, no. 3 (2019): 243-259.
36. Nadimi-Shahraki, Mohammad H., Shokooh Taghian, and Seyedali Mirjalili. "An improved grey wolf optimizer for solving engineering problems." Expert Systems with Applications 166 (2021).
37. Nickabadi, Ahmad, Mohammad Mehdi Ebadzadeh, and Reza Safabakhsh. "A novel particle swarm optimization algorithm with adaptive inertia weight." Applied soft computing 11, no. 4 (2011): 3658-3670.
38. Olorunda, Olusegun, and Andries P. Engelbrecht. "Measuring exploration/exploitation in particle swarms using swarm diversity." In 2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence), pp. 1128-1134. IEEE, 2008.
39. Perez, Luis, and Jason Wang. "The effectiveness of data augmentation in image classification using deep learning." arXiv preprint arXiv:1712.04621 (2017).
40. Shim, Seong-O., Monagi H. Alkinani, Lal Hussain, and Wajid Aziz. "Feature ranking importance from multimodal radiomic texture features using machine learning paradigm: A biomarker to predict the lung cancer." Big Data Research (2022).
41. Shreem, Salam Salameh, Hamza Turabieh, Sana Al Azwari, and Faiz Baothman. "Enhanced binary genetic algorithm as a feature selection to predict student performance." Soft Computing 26, no. 4 (2022): 1811-1823.
42. Singh, Indu, Raghav Bansal, Anchal Gupta, and Ayush Singh. "A hybrid grey wolf-whale optimization algorithm for optimizing SVM in breast cancer diagnosis." In 2020 Sixth International Conference on Parallel, Distributed

and Grid Computing (PDGC), pp. 286-290. IEEE, 2020.

43. Singh, Narinder, and Hanaa Hachimi. "A new hybrid whale optimizer algorithm with mean strategy of grey wolf optimizer for global optimization." Mathematical and computational applications 23, no. 1 (2018).
44. Singh, Narinder, and S. B. Singh. "Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance." Journal of Applied Mathematics (2017).
45. Storn, Rainer. "On the usage of differential evolution for function optimization." In Proceedings of north american fuzzy information processing, pp. 519-523. Ieee, 1996.
46. Teng, Zhi-jun, Jin-ling Lv, and Li-wen Guo. "An improved hybrid grey wolf optimization algorithm." Soft computing 23 (2019): 6617-6631.
47. Tuba, Milan, Milos Subotic, and Nadezda Stanarevic. "Modified cuckoo search algorithm for unconstrained optimization problems." In Proceedings of the 5th European conference on European computing conference, pp. 263-268. 2011.
48. Wahba, Maram A., Amira S. Ashour, Sameh A. Napoleon, Mustafa M. Abd Elnaby, and Yanhui Guo. "Combined empirical mode decomposition and texture features for skin lesion classification using quadratic support vector machine." Health information science and systems 5 (2017): 1-13.
49. Wu, Yu. "A survey on population-based meta-heuristic algorithms for motion planning of aircraft." Swarm and Evolutionary Computation 62 (2021).
50. Yang, Xin-She, and Amir Hossein Gandomi. "Bat algorithm: a novel approach for global engineering optimization." Engineering computations 29, no. 5 (2012): 464-483.
51. Yang, Xin-She. "Bat algorithm for multi-objective optimisation." International Journal of Bio-Inspired Computation 3, no. 5 (2011): 267-274.
52. Zukhri, Zainudin, and Irving Vitra Paputungan. "A hybrid optimization algorithm based on genetic algorithm and ant colony optimization." International Journal of Artificial Intelligence & Applications 4, no. 5 (2013).