Received: April 8, 2023. Revised: May 20, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Pelican Optimization with Majority Voting Ensemble Model for Tuberculosis Detection and Classification on Chest X-Ray Images

K. Manivannan[1] **S. Sathiamoorthy**[2]

[1]*Department of Computer and Information Science, Annamalai University, Annamalai Nagar, India*
[2]*Annamalai University PG Extension Centre, Villupuram. Tamilnadu, India*
* Corresponding author's Email: manik81au@gmail.com

Abstract: Tuberculosis (TB) detection and classification on chest X-ray (CXR) images remains the most significant task in medical diagnosis. TB is a contagious disorder that affects the pulmonary region, and its diagnosing process often depends on CXR. CXR images are utilized for classifying and detecting TB lesions, including infiltrates, cavities, pleural effusions, and nodules. Manual analysis by radiologists includes a visual assessment of the X-ray images by a skilled physician or radiologist. There were many techniques to automatically classify and detect TB on CXR, including deep learning-based approaches, manual interpretation by radiologists, and computer-aided diagnosis (CAD) systems. This manuscript offers the design of pelican optimization with majority voting ensemble model for tuberculosis detection and classification (POMVE-TDC) technique on the CXR images. The core objective of the POMVE-TDC approach is to classify the incidence of TB on the CXR images. At the primary stage, the POMVE-TDC technique undergoes a contrast enhancement process. Besides, the densely connected network (DenseNet-161) model is applied for the extraction of feature vectors. Meanwhile, a pelican optimization algorithm (POA) based hyperparameter optimizer is designed for the DenseNet-161 model. Finally, a majority voting ensemble classifier comprising graph convolution network (GCN), autoencoder (AE), and extreme learning machine (ELM) models are used. The performance evaluation of the POMVE-TDC technique on the medical dataset highlights the significant outcomes with maximum accuracy of 98.83%.

Keywords: Tuberculosis, Chest X-ray images, Computer-aided diagnosis, Deep learning, Pelican optimization algorithm.

1. Introduction

Lung diseases are conditions where the patient's breathing pattern is severely affected because of lacking oxygen and relevant symptoms [1]. Tuberculosis (TB) is one of the harmful types of lung disease that has led to a detrimental effect on humans. The mycobacterium tuberculosis bacteria causes TB [2]. Generally, the lung region is the targeted part of this disorder, but it could probably affect other body parts. It is an infectious syndrome, that is, when individuals affected with TB sneeze or cough, they transfer the illness-producing virus into the air [3]. Manual assessment approaches need experts in that field and were labour intensive to deliver precise inferences [4]. Hence, there comes a necessity to merge the existing technological progressions with medical processes and theories. Artificial intelligence (AI) is thriving in current years and has spread over several industries [5]. The introduction of AI into the healthcare sector has increased the growth rate largely in many kinds of study and has rendered the freedom for scientists to sightsee unknown territories. In TB, the most commonly utilized AI tool is computer-aided detection (CAD) [6]. The tool examines the chest X-rays (CXR) of patients and finds if TB infected the patient. Through this, the load on the radiologists is minimalized to precisely scan with the help of all radiographic films and eventually quickens the screening process.

In the past decades, AI-based solutions were used for the automatic detection of brain tumours and breast cancer, physiological monitoring, etc [7]. ML

as a subdivision of AI allows computers to learn without interference from humans, mechanically without being programmed [8]. Simply, ML can identify patterns in data (images). As a subfield of ML, deep learning (DL) leverages manifold layers for extracting higher-level features from raw inputs. In recent times, DL techniques remain the existing method for image classification [9]. Deep convolutional neural networks (DCNN) stand as a potential approach for image classification among DL methods. Lately, quite a few researchers have exploited CNNs for the automatic identification of pulmonary illnesses like pneumonia from CXR [10]. In the DL structure, transfer learning (TL) technique has been applied to detect TB with the pre-trained models and their ensembles.

This manuscript offers the design of pelican optimization with majority voting ensemble model for tuberculosis detection and classification (POMVE-TDC) technique on the CXR images. The objective of the POMVE-TDC approach is to classify the occurrence of TB on the CXR images. At the primary stage, the POMVE-TDC technique undergoes a contrast enhancement process. Besides, the densely connected network (DenseNet-161) model is applied for the extraction of feature vectors. Meanwhile, a pelican optimization algorithm (POA) based hyperparameter optimizer is designed for the DenseNet-161 model. Finally, a majority voting ensemble classifier comprising graph convolution network (GCN), autoencoder (AE), and extreme learning machine (ELM) models are used. The performance evaluation of the POMVE-TDC technique on the medical dataset highlights the significant outcomes over other DL approaches in terms of different measures.

The paper organization is as follows. Section 2 presents the related works and section 3 presents the proposed model. Then, section 4 presents the result analysis and section 5 concludes the paper.

2. Related works

In [11], inspected the performance variations with an InceptionV3 UNet approach utilizing several image resolutions without or with aspect ratio adjustments and lung ROI cropping and through a wide range of empirical assessments, identified the optimum resolution of the image for enhancing the performance of TB-consistent lesion segmentation. Nguyen et al. [12] explored the TL efficiency of medical imagery for detecting TB. The study has shown an improved technique for TL over the regular method of employing ImageNet weights. From ImageNet weights, the author has found that the low-level features are of no use for image tasks for methods like X-rays and devised an innovative method to acquire low-level factors by trained methods in a multiclass condition. Sahlol et al. [13] presented a new hybrid algorithm for the potential classifying of CXR images. Firstly, using MobileNet, the attributes will be extracted from CXR images, a CNN method that is formerly trained over the ImageNet dataset. After, ascertaining the most appropriate feature, the artificial ecosystem-based optimizer (AEO) is applied as a feature selecting technique. In [14], introduced a structure to potentially categorize and discriminate the viral pneumonia, PTB, and bacterial pneumonia from CXR imaging. The analysis was executed through the NN technique. For the data pre-processing, a lot of data augmentation algorithms have been exploited that boost the classification accuracy and validation of the presented method.

Fati et al. [15] emphases on applying two AI methods, ANN and CNN. As well, this study introduces two different techniques with two organisms for diagnosing TB from the following 2 datasets. The initial one hybridizes two CNN methods i.e., GoogLeNet and Res-Net-50 methods. Before the classification, the PCA approach was enforced, for minimalizing the dimensionality of features, targeting to derive the deep features. Depending on X-ray imageries, from both data, this hybrid method attained better outcomes in spotting TB. Conversely, the second method implements ANN depending on the merged features derived by GoogleNet and ResNet-50 methods and integrates them with attributes derived by LBP, GLCM, and DWT algorithms. Aulia and Hadiyoso [16] devised a system that depends on X-ray images utilizing DL for TB detection. This system makes use of a CNN with the VGG-16 structure.

Though several models are available in the literature, it is still needed to improve TB detection and classification performance. Most of the existing models have not concentrated on hyperparameter tuning techniques. Since trial-and-error tuning process is a tedious procedure, metaheuristic methods can be used. Hence, the POA is used as a hyperparameter optimizer of the DenseNet-161 model.

3. The proposed model

In this manuscript, an automated and accurate TB detection model, named the POMVE-TDC technique was proposed on the CXR images. The core objective of the POMVE-TDC algorithm is to classify the presence of TB on the CXR images. It encompasses

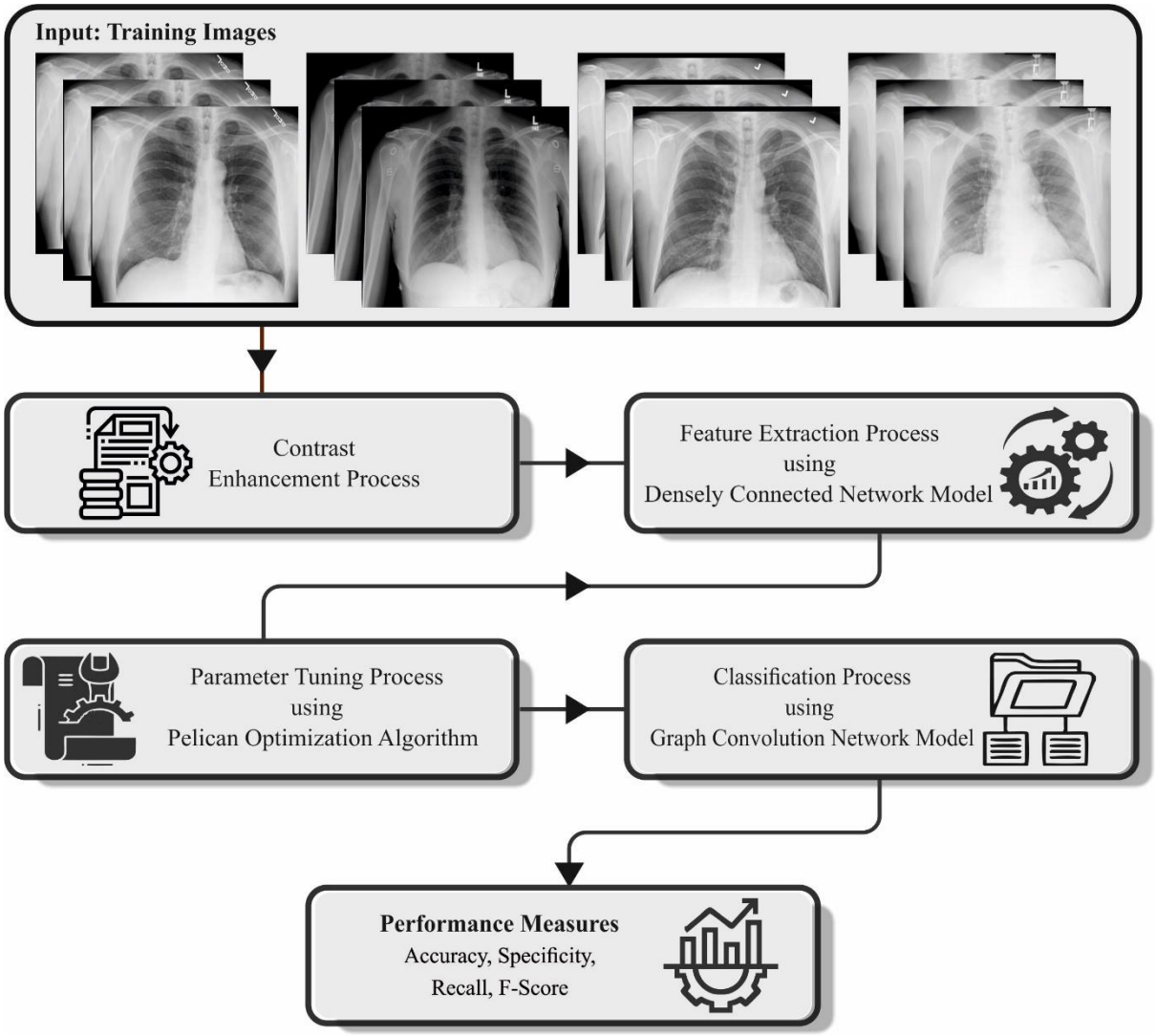

Figure. 1 Workflow of the POMVE-TDC approach

contrast enhancement, DenseNet-161 feature extraction, POA-based hyperparameter tuning, and ensemble voting classifier. Fig. 1 represents the workflow of the POMVE-TDC algorithm.

3.1 Contrast enhancement process

At the primary stage, the POMVE-TDC technique undergoes a contrast enhancement process. The CLAHE approach is more suitable for enhancing the brightness of an image and this can be explained in the following [17].

1) With similar sizes of 8x8 blocks, each image is divided into non-overlapping background areas. The area of 64 pixels is represented by each block.
2) The histogram of intensity is calculated for every contextual region.
3) The threshold constraint can be utilized effectively for altering the brightness of an image. Now, the clip limit is set to clip the histogram. By applying the high clip limit, the local image brightness can be enhanced thus it is fixed to a minimal optimal value.
4) Every histogram can be altered by selecting a transformation purpose.
5) Every histogram was changed not outside the chosen limiting of the clip. With this technique, the gray level changed with even dispersion can be represented as follows:

$$gr = [gr_{\max} - gr_{\min}] \times PD(f) + gr_{\min} \quad (1)$$

In Eq. (1), the maximal pixel is indicated $asgr_{\max}$, the minimum pixel value can be determined $asgr_{\min}$, the computed pixel values are determined $asgr$ and the values of CPD can be represented by $PD(f)$. By applying Eq. (2) the gray level can be evaluated for enhancing the exponential distribution.

$$gr = gr_{\min} - \left(\frac{1}{\alpha}\right) \times \ln\left[1 - PD(f)\right] \quad (2)$$

6) The adjacent tiles were combined for bilinear interpolation. The image grayscale value can be modified based on the altered histograms.

3.2 DenseNet-161 feature extractor

In this work, the DenseNet-161 model is applied for the extraction of feature vectors. DenseNet169 is selected as the better model concerning the accuracy and F-score [18]. TL is a powerful tool to accomplish suitable outcomes in classification problems with the help of a small dataset. The TL algorithm is effectively employed for recognizing the severity of DR. Thus, it can be stimulated by the previous outcomes grounded on the DenseNet169 architecture as a pre-train TL algorithm to identify 5 phases of HR. In addition, hyper-tuning the deep TL (DTL) algorithm could improve the accuracy. The DTL algorithm based on DenseNet169 is developed. The presented method leveraged the learned weight over the ImageNet datasets, along with the convolution network, for feature extraction. Direct connections from every prior layer to every succeeding layer were added to improve transmission in the DenseNet169 architecture. The mathematical model of feature concatenation is shown as:

$$X_l = H_l([x_0, x_1, x_2, \dots, x_{l-1}]) \quad (3)$$

In Eq. (3), $x_0, x_1, x_2, \dots, x_{l-1}$ denotes the concatenation of features map generated by the nonlinear conversion H_l that is defined as a composite function involving BN, added by the convolution unit of (3×3) and ReLU. The dense block is generated in the network framework for the downsampling purpose and was split by a layer named transition layer that comprises $a(1\text{x}1)$ convolution layer, $a(2\text{x}2)$ average pooling layer, and BN. Due to its structure that considered feature maps as a network's global state, DenseNet169 implements even with a slow progression rate. Consequently, every succeeding layer has access to every feature map from the preceding layer.

$$f_{maps}(k_l) = kN(l-1) + k_0 \quad (4)$$

In Eq. (4), k_0 represents the channel in the inputted layer. A bottleneck layer viz., a (1 x 1) complex layer was added beforehand to every (3x3) complex layer to improve the computation performance by decreasing the amount of input featuring maps that are generally higher than output featuring maps k. If the dimension of the function map differs, DenseNet was separated into DenseBlock with a set of filters but a similar dimension. The transition layer exploits downsampling to implement BN; this turns out to be a crucial stage in CNN. Although every DenseNet layer only generates k output feature maps, it generally includes multiple inputs. To decrease the input count and thereby increase the computation accuracy, a bottleneck (1x1) convolution is included as a transition layer beforehand every (3 x 3) convolution layer. Accordingly, it is difficult to group them and no distinction was made if the grouping is an addition or concatenation. From a theoretical point of view, networking is a sequence of parallel and serial computations that maps input toward the output.

3.3 Hyperparameter tuning using POA

To adjust the tuning associated with the DenseNet-161 approach, the POA is exploited. POA is a populace-based method where the pelican is a population member [19]. Some of the recent metaheuristics are artificial bee colony (ACO) algorithm, mixed leader based optimizer (MLBO) [20], fixed step average and subtraction based optimizer (FSASO) [21], ring toss game-based optimization (RTGO) algorithm [22], puzzle optimization algorithm (POA) [23], random selected leader based optimizer (RSLBO) [24], three influential members based optimizer (TIMBO) [25], guided pelican algorithm (GPA) [26], stochastic komodo algorithm (SKA) [27], and so on. In the population-based technique, every population member signifies the candidate solution. Every group member presents the values for the optimization issue parameter depending on the position in the search space. At first, an equation was applied for arbitrarily initializing the member of the populace based on the upper and lower bounds of the problem.

$$x_{i,j} = l_j + rand \cdot \left(u_j - l_j\right), \quad i = 1,2,\dots,N, j = 1,2,\dots,m, \quad (5)$$

In Eq. (5), N denotes the member of the populace numbers; $x_{i,j}$ represent the value of the $j\text{-}th$ parameter mentioned by the $i\text{-}th$ candidate solution; $rand$ refers to the random number within $[0,1], m$ shows the number of problem parameters; l_j and u_j denote the $j\text{-}th$ problem parameter's lower and upper bound.

A population matrix was utilized in Eq. (6) for identifying the populace member of pelicans in

developed POA. All rows in the matrix characterize the candidate resolution, and the column of matrices characterizes the presented values of the problem parameter.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} \chi_{1,1} & \cdots & \chi_{1,j} & \cdots & \chi_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \quad (6)$$

In Eq. (6), X denotes the matrix of the pelican populace, X_i represents the $i\text{-}th$ pelican.

In this work, all the group members denote a pelican, which represents the candidate's solution to a presented issue. Hence, the objective function of the specified issue is estimated according to every candidate solution.

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \quad (7)$$

In Eq. (7), F_j shows the value of the objective function of $the\ i\text{-}th$ candidate solution, and F represents the objective function vector.

The presented POA simulates the pelican behaviour, and strategy while hunting, and attacking prey for upgrading the candidate solution. This hunting method can be simulated in two phases: (1) Wingspan above water (development phase). (2) Move toward prey (exploration stage).

Initially, the pelican recognizes the position of prey and later travels toward that recognized region. Modelling this pelican approach allows search space scan and enhances the exploring capability of this presented work in finding distinct regions of search space. The essential one regarding the POA is that the prey's position was arbitrarily generated in the searching space. This raises the ability of POA for exploring space for accurate search problem resolution.

$$x_{i,j}^{P_1} = \begin{cases} x_{i,j} + rand \cdot (p_j - I \cdot x_{i,j}), & F_p < F_i \\ \chi_{i,j} + rand \cdot (\chi_{i,j} - p_j), & else \end{cases} \quad (8)$$

Where $x_{i,j}^{P_1}$ represents the newest state of $i\text{-}th$ pelicans at $the\ j\text{-}th$ dimension, I denote the random value equivalent to 1 or 2, p_j denotes the prey location of at $j\text{-}th$ parameter, and F_p denotes the objective function values. The variable I was a number that is arbitrarily equivalent to 1 or 2. This variable was arbitrarily chosen for all the iterations and every member. While the value of this variable was equivalent to 2, it generates much displacement for members that might cause them to enter the new segment of searching space. Thus, parameter I will be affecting the POA probe's capability for precisely scanning the search space.

In the presented method, the pelican novel position was suitable if the objective function's value was enhanced at that position. In these types of updates, named valid updates, the process is not permitted from moving to the non-optimum area. The procedure is modelled by Eq. (9):

$$X_i = \begin{cases} X_i^{P_1}, & F_i^{P_1} < F_j \\ X_i, & else \end{cases} \quad (9)$$

Where $X_i^{P_1}$ denotes the novel i-th pelican state, and $F_i^{P_1}$ represent its value of the objective function depending on phase 1. Beforehand gathering their prey in throat bags, they will be spreading their wings over water to transfer the fish upwardly after the pelican reaches the surface. This strategy paves the way for the pelicans in catching more fish in zones under attack. Modelling these behaviours in pelicans enables the presented technique to converge toward the optimum point in the hunting region. This procedure enhances the local searching and advancement abilities of the POA. From the scientific viewpoint, the model should inspect points nearer the pelican location for converging on the best solution. This pelican behaviour can be mathematically simulated as follows.

$$x_{i,j}^{P_2} = \chi_{i,j} + R \cdot \left(1 - \frac{t}{T}\right) \cdot (2 \cdot rand - 1) \cdot \chi_{i,j} \quad (10)$$

In Eq. (10), $R \cdot (1 - t/T)$ represent the $x_{i,j}$ neighbourhood radius, $x_{i,j}^{P_2}$ denotes the $i\text{-}th$ pelican grounded on stage 2 of the novel state of $j\text{-}th$ parameter, R represents constant and equivalent to 0.2, t denotes the iteration count, T shows the maximum amount of iterating. Co-efficient $R \cdot (1 - t/T)''$ signifies the group member's neighbourhood radius and all members in neighbouring local search convergence to the best solution. The co-efficient efficiently presents the POA developing ability nearer to the optimum global solution. In the first iteration, the coefficient value was greater; thus, the area near every component is greater. With that regard, the $''R \cdot (1 - t/T)''$ coefficient reduces, on top of the neighbourhood radius of the total members. This enables to scan of the region around all members of the populace at a small and accurate step sizing

such that POA converges nearer to the global optimum solution.

In this phase, a valid update is used for rejecting or accepting the newest pelican position modelled in Eq. (11).

$$X_i = \begin{cases} X_i^{P_2}, & F_i^{P_2} < F_j \\ X_i, & else \end{cases} \tag{11}$$

In Eq. (11), $F_i^{P_2}$ represent the objective function value and $X_i^{P_2}$ shows the newest state of $the\ i\text{-}th$ pelican.

The fitness selection was a decisive factor in the POA methodology. Solution encoding is leveraged to judge the aptitude of the candidate solution. Here, the accuracy value will be the key condition exploited to devise a fitness function.

$$Fitness\ =\ \max\,(P) \tag{12}$$

$$P = \frac{TP}{TP+FP} \tag{13}$$

Here FP signifies the false positive value and TP represents the true positive value.

3.4 Ensemble majority voting classifier

At the final stage, the ensemble majority voting classification model is used. Assume we have training sets and classifier set as $h_1, h_2, \ldots, h_n$, and all classifiers were trained on training sets [28]. Therefore, the classifier may generate predictions after training. The classifier h_1 can generate the prediction y_1 ; classifier h_2 , prediction y_2 ; and classifierh_n, predictiony_n. For all new data points, n anticipations are available. Voting minimalizes n class labelling anticipations for one data point into one class. Hence, majority voting is used for determining the final voting. Operation mode can be utilized to gain the final vote, which is given in Eq. (14).

$$y_f\ =\ mode\{h_1(x), h_2(x), \ldots, h_n(x)\} \tag{14}$$

Here $h_i(x)\ =\ y_i(x)$.

Utilizing maximum voting is requesting a group of specialists for voting on a particular resolution. If most of the members cast their votes, the resolution was recognized. Like this, the probabilities of making anticipation mistakes are trivial.

3.4.1. ELM model

ELM is a special SLFN without iterative adjustment of the structural parameter. ELM is split into prediction and learning. For ELM with $an\ M$ input layer, an O output layer, activation function $\sigma(W^1, X, b)$ (the activation function could be any non-zero function) and L hidden neuron. Assume input vector $X_{M\times 1}$,

$$X_{M\times 1} = [x_1, x_M]^T. \tag{15}$$

The parameter of ELM is generated at random as:

$$W_{L\times M}^1 = \begin{bmatrix} w_{1,1}^1 & \ldots & w_{1,M}^1 \\ \vdots & \ddots & \vdots \\ w_{L,1}^1 & \ldots & w_{L,M}^1 \end{bmatrix} \tag{16}$$

$$b_{L\times 1} = [b_1, \ldots, b_L]^T. \tag{17}$$

Next, the output of the hidden layer is given below:

$$H_{L\times 1} = \sigma(W_{L\times M}^1 \cdot X_{M\times 1} + b_{L\times 1}). \tag{18}$$

The output weight matrix $W_{O\times L}^2$ is defined by Eq. (19):

$$W_{O\times L}^2 = \begin{bmatrix} w_{1,1}^2 & \ldots & w_{1,L}^2 \\ \vdots & \ddots & \vdots \\ w_{O,1}^2 & \ldots & w_{O\times L}^2 \end{bmatrix}. \tag{19}$$

The output $\hat{Y}_{O\times 1}$ is evaluated by

$$\hat{Y}_{O\times 1} = W_{O\times L}^2 \cdot H_{L\times 1}. \tag{20}$$

Assume a N training instance $(X_{M\times N}, Y_{O\times N})$, the output of ELM is shown below:

$$\hat{Y}_{O\times N} = W_{O\times L}^2 \cdot H_{L\times N}. \tag{21}$$

The training goal can be defined as follows:

$$\|Y_{O\times N} - \hat{Y}_{O\times N}\| = 0. \tag{22}$$

The output weight matrix is mathematically expressed below:

$$W_{O\times L}^2 = Y_{O\times N} \cdot H_{L\times N}^{+}, \tag{23}$$

In Eq. (23), $H_{L\times N}^{+}$ denotes the moore-penrose generalized inversing of $H_{L\times N}$. A regularization term was presented to increase the generalization capability of ELM and prevent overfitting, depending on the principle of ridge regression (23):

$$W_{O\times L}^{2} = Y_{O\times N} \cdot (H_{L\times N}^{T} \cdot H_{L\times N} + \frac{I}{C})^{-1} \cdot H_{L\times N}^{T}, \quad (24)$$

In Eq. (24), C represents the regularization factor and I denote the identity matrix.

3.4.2. AE model

AE is an unsupervised ANN used for learning representation. The AE model is developed to enact a bottleneck in networking that forced a compressed knowledge depiction of inputted value. Thus, the relationship among input factors can be reconstructed and learned. AE has an encoder-decoder framework. The encoder maps the unique input χ into the concealed layer that can be a hidden space depiction. The decoding part reconstructed the hidden representation into $\hat{x}$. The decoding and encoding procedures are formulated by the subsequent equation

$$h = \sigma(Wx + b), \quad (25)$$

$$\hat{x} = \sigma(W'h + b') \quad (26)$$

Where $\hat{x} = (\hat{x}_1, \hat{x}_2, \dots \hat{x}_n)$ shows the reconstructed input, $\chi = (x_1, x_2, \dots x_n)$ characterizes the input data vector, and $h = (h_1, h_2, \dots h_n)$ signifies the lower dimensional vector acquired from the concealed layer. σ shows the sigmoid activation function, viz., $\sigma = \frac{1}{1+e^{-x}}$, W and W' denotes the weight matrices, and b and b' indicates the bias vectors.

3.4.3. GCN model

GCN is a NN intended for working straightly on graphing and leveraging its structural data. The GCN input was a graph $= (V, E)$, whereas V and E represent the group of nodes and edges correspondingly. V is defined as $N \times F$ feature matrix X, produced by the feature vector of length F linked with all the N nodes. The structural data of the graph surrounded in E, conversely, is demonstrated as $N \times N$ adjacent matrix A. Fig. 2 depicts the architecture of GCN. GCNs make a node-level output O from the procedure of $the\ N \times K$ matrix, whereas K implies the count of resultant features calculated to all the nodes.

This matrix is expressed as a function of feature matrix X and adjacent matrix A as:

$$O = f(X, A) = \sigma(AXW) \quad (27)$$

whereas W signifies the $F \times K$ trainable weighted matrix.

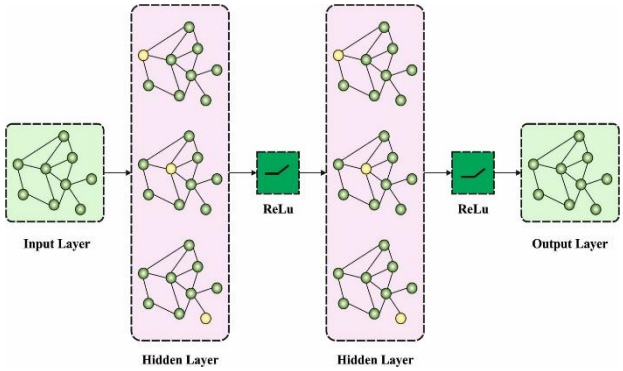

Figure. 2 GCN structure

Table 1. Details of dataset

Class Name	Number of Samples
Normal	3500
Tuberculosis (TB)	700
Total Number of Samples	**4200**

Once multiplied by A, for every node a weighted sum has been calculated among the featuring vectors of every neighbourhood node eliminating itself. Due to this cause, self-looping can be supplemented by determining $\hat{A} = A + I$, whereas I denote the NxN identity matrix.

While A was generally non-normalized, the multiplications of the matrix determined in Eq. (27) cause a charge of scales from the featuring vectors. The adjacent matrix is the degree matrix by calculating $D^{\frac{1}{2}} A D^{\frac{1}{2}}$ for avoiding numerical variabilities and vanishing or exploding gradients, whereas D signifies the diagonal node degree matrix demonstrated as $D_{ii} = \sum_j A_{ij}$.

Assuming the preceding adjustments, the resultant GCN layer is rewritten as:

$$O = f(X, A) = \sigma\left(D^{\frac{1}{2}} \hat{A} D^{\frac{1}{2}} XW\right) \quad (28)$$

It can be feasible for defining a multiple-layer GCN by presenting the resultant featuring matrices of layers along with the adjacent matrix A as an input value for subsequent layers. Commonly, the GCNs were trained by minimising a loss function $\mathcal{L}$ determined among its output O and the anticipated output $\overline{O}$ using the backpropagation (BP) technique.

4. Results and discussion

In this study, the TB classification outcomes of the POMVE-TDC technique are studied on the Kaggle dataset [29] that comprises 4200 trials and two classes as depicted in Table 1. Fig. 3

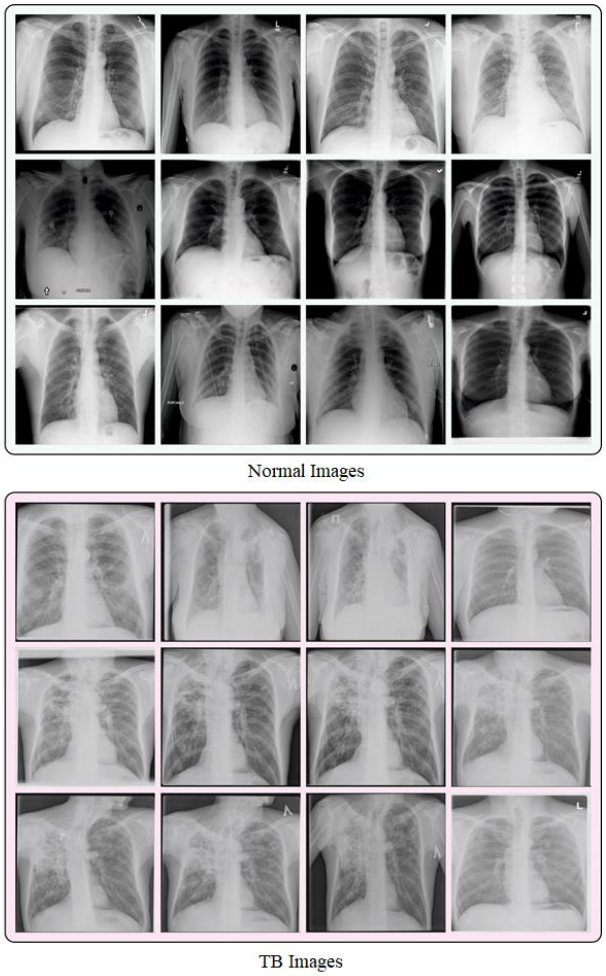

Figure. 3 Sample images

demonstrates the trial normal and TB imageries.

Table 2 illustrates a comprehensive TB classification result of the POMVE-TDC methodology under varying epochs. The figure indicates the improved diagnosis results under all epochs. For instance, under 200 epochs, the POMVE-TDC technique gains an average $accu_y$ of 97.31, $reca_l$ of 97.31%, $spec_y$ of 97.31%, F_{score} of 97.26%, and MCC of 94.52%. At the same time, under 800 epochs, the POMVE-TDC system reaches an average $accu_y$ of 98.77, $reca_l$ of 98.77%, $spec_y$ of 98.77%, F_{score} of 98.38%, and MCC of 96.78%. In the meantime, under 1400 epochs, the POMVE-TDC algorithm attains an average $accu_y$ of 98.46, $reca_l$ of 98.46%, $spec_y$ of 98.46%, F_{score} of 98.46%, and MCC of 96.91%. Furthermore, under 2000 epochs, the POMVE-TDC methodology obtains an average $accu_y$ of 99.56, $reca_l$ of 99.56%, $spec_y$ of 99.56%, F_{score} of 99.70%, and MCC of 99.40%.

A comparative $accu_y$ investigation of the POMVE-TDC methodology with current DL approaches [30] is given in Table 3. The results indicate that the POMVE-TDC methodology reaches a higher $accu_y$ of 98.83%.

Table 2. TB classifier outcome of POMVE-TDC approach under varying epochs

Class	Accuracy	Recall	Specificity	F-Score	M CC
Epoch - 200					
Entire Dataset					
Normal	99.06	99.06	95.57	99.09	94.52
TB	95.57	95.57	99.06	95.44	94.52
Average	**97.31**	**97.31**	**97.31**	**97.26**	**94.52**
Epoch - 400					
Entire Dataset					
Normal	99.37	99.37	95.29	99.22	95.26
TB	95.29	95.29	99.37	96.04	95.26
Average	**97.33**	**97.33**	**97.33**	**97.63**	**95.26**
Epoch - 600					
Entire Dataset					
Normal	99.37	99.37	97.29	99.41	96.49
TB	97.29	97.29	99.37	97.08	96.49
Average	**98.33**	**98.33**	**98.33**	**98.25**	**96.49**
Epoch - 800					
Entire Dataset					
Normal	99.26	99.26	98.29	99.46	96.78
TB	98.29	98.29	99.26	97.31	96.78
Average	**98.77**	**98.77**	**98.77**	**98.38**	**96.78**
Epoch - 1000					
Entire Dataset					
Normal	99.69	99.69	98.71	99.71	98.29
TB	98.71	98.71	99.69	98.57	98.29
Average	**99.20**	**99.20**	**99.20**	**99.14**	**98.29**
Epoch - 1200					
Entire Dataset					
Normal	98.69	98.69	95.14	98.86	93.19
TB	95.14	95.14	98.69	94.33	93.19
Average	**96.91**	**96.91**	**96.91**	**96.59**	**93.19**
Epoch - 1400					
Entire Dataset					
Normal	99.49	99.49	97.43	99.49	96.91
TB	97.43	97.43	99.49	97.43	96.91
Average	**98.46**	**98.46**	**98.46**	**98.46**	**96.91**
Epoch - 1600					
Entire Dataset					
Normal	99.34	99.34	96.43	99.31	95.88
TB	96.43	96.43	99.34	96.57	95.88
Average	**97.89**	**97.89**	**97.89**	**97.94**	**95.88**
Epoch - 1800					
Entire Dataset					
Normal	99.97	99.97	99.14	99.90	99.40
TB	99.14	99.14	99.97	99.50	99.40
Average	**99.56**	**99.56**	**99.56**	**99.70**	**99.40**
Epoch - 2000					
Entire Dataset					
Normal	99.80	99.80	97.86	99.69	98.11
TB	97.86	97.86	99.80	98.42	98.11
Average	**98.83**	**98.83**	**98.83**	**99.05**	**98.11**

Table 3. Accuracy outcome of POMVE-TDC approach with recent DL algorithms

Methods	Accuracy (%)
ResNet-l8	96.53
ResNet-50	97.63
ResNet-101	97.98
Inception-V3	98.91
Vgg-19 Model	98.60
DenseNet-201	98.32
POMVE-TDC	98.83

5. Conclusion

In this manuscript, an automated and accurate TB detection model, named the POMVE-TDC technique has been developed on the CXR images. The core objective of the POMVE-TDC method is to classify the TB presence on the CXR images. The proposed POMVE-TDC method comprises contrast enhancement, DenseNet161-based feature extraction, POA-based hyperparameter tuning, and ensemble classification. Moreover, a majority voting ensemble classifier comprising ELM, AE, and GCN is used. The performance evaluation of the POMVE-TDC technique on the medical dataset highlights the significant outcomes over other DL approaches with maximum accuracy of 96.53%. In the future, advanced DL feature extractors can boost the performance of the presented approach.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Manivannan and Sathiamoorthy; methodology, Manivannan and Sathiamoorthy; software, Manivannan; validation, Manivannan; formal analysis, Manivannan and Sathiamoorthy; investigation, Manivannan and Sathiamoorthy; resources, Manivannan; data curation, Manivannan; writing-original draft preparation, Manivannan; writing-review and editing, Manivannan and Sathiamoorthy; visualization, Manivannan; supervision, Manivannan; project administration, Manivannan; funding acquisition, Manivannan. All authors have read and approved the final manuscript.

References

[1] A. Iqbal, M. Usman, and Z. Ahmed, "An efficient deep learning-based framework for tuberculosis detection using chest X-ray images", *Tuberculosis,* Vol. 136, p. 102234, 2022.

[2] E. Showkatian, M. Salehi, H. Ghaffari, R. Reiazi, and N. Sadighi, "Deep learning-based automatic detection of tuberculosis disease in chest X-ray images", *Polish Journal of Radiology,* Vol. 87, No. 1, pp. 118-124, 2022.

[3] M. Ayaz, F. Shaukat, and G. Raja, "Ensemble learning based automatic detection of tuberculosis in chest X-ray images using hybrid feature descriptors", *Physical and Engineering Sciences in Medicine,* Vol. 44, No. 1, pp. 183-194, 2021.

[4] S. Urooj, S. Suchitra, L. Krishnasamy, N. Sharma, and N. Pathak, "Stochastic Learning-Based Artificial Neural Network Model for an Automatic Tuberculosis Detection System Using Chest X-Ray Images", *IEEE Access,* Vol. 10, pp. 103632-103643, 2022.

[5] V. Acharya, G. Dhiman, K. Prakasha, P. Bahadur, A. Choraria *et al.*, "AI-assisted tuberculosis detection and classification from chest X-rays using a deep learning normalization-free network model", *Computational Intelligence and Neuroscience,* 2022, doi: 10.1155/2022/2399428.

[6] K. C. Santosh, S. Allu, S. Rajaraman, and S. Antani, "Advances in Deep Learning for Tuberculosis Screening Using Chest X-Rays: The Last 5 Years Review", *Journal of Medical Systems,* Vol. 46, No. 11, p. 82, 2022.

[7] A. Wong, J. R. H. Lee, H. R. Khah, A. Sabri, A. Alaref *et al.*, "TB-Net: a tailored, self-attention deep convolutional neural network design for detection of tuberculosis cases from chest X-ray images", *Frontiers in Artificial Intelligence,* Vol. 5, 2022, doi: 10.3389/frai.2022.827299.

[8] S. Lee, J. J. Yim, N. Kwak, Y. J. Lee, J. K. Lee *et al.*, "Deep learning to determine the activity of pulmonary tuberculosis on chest radiographs", *Radiology,* Vol. 301, No. 2, pp. 435-442, 2021.

[9] M. Nijiati, Z. Zhang, A. Abulizi, H. Miao, A. Tuluhong *et al.*, "Deep learning assistance for tuberculosis diagnosis with chest radiography in low-resource settings", *Journal of X-ray Science and Technology,* Vol. 29, No. 5, pp. 785-796, 2021.

[10] M. K. Puttagunta and S. Ravi, "Detection of Tuberculosis based on Deep Learning based methods", *Journal of Physics: Conference Series,* Vol. 1767, p. 012004, 2021.

[11] S. Rajaraman, F. Yang, G. Zamzmi, Z. Xue, and S. Antani, "Assessing the Impact of Image Resolution on Deep Learning for TB Lesion Segmentation on Frontal Chest X-rays", *Diagnostics,* Vol. 13, No. 4, p. 747, 2023.

[12] Q. H. Nguyen, B. P. Nguyen, S. D. Dao, B.

Unnikrishnan, R. Dhingra *et al.*, “Deep Learning Models for Tuberculosis Detection from Chest X-ray Images”, In: *Proc. of 2019 26th International Conference on Telecommunications (ICT)*, Hanoi, Vietnam: IEEE, 2019, pp. 381–385, doi: 10.1109/ICT.2019.8798798.

[13] A. T. Sahlol, M. A. Elaziz, A. T. Jamal, R. Damaševičius, and O. F. Hassan, "A novel method for detection of tuberculosis in chest radiographs using artificial ecosystem-based optimisation of deep neural network features", *Symmetry*, Vol. 12, No. 7, p. 1146, 2020.

[14] D. Verma, C. Bose, N. Tufchi, K. Pant, V. Tripathi *et al.*, "An efficient framework for identification of Tuberculosis and Pneumonia in chest X-ray images using Neural Network," *Procedia Computer Science*, Vol. 171, pp. 217-224, 2020.

[15] S. M. Fati, E. M. Senan, and N. E. Hakim, "Deep and Hybrid Learning Technique for Early Detection of Tuberculosis Based on X-ray Images Using Feature Fusion", *Applied Sciences*, Vol. 12, No. 14, p. 7092, 2022.

[16] S. Aulia and S. Hadiyoso, "Tuberculosis Detection in X-Ray Image Using Deep Learning Approach with VGG-16 Architecture", *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, Vol. 8, No. 2, pp. 290-297, 2022.

[17] S. Gupta, S. Thakur, and A. Gupta, "Optimized hybrid machine learning approach for smartphone based diabetic retinopathy detection", *Multimedia Tools and Applications*, Vol. 81, No. 10, pp. 14475-14501, 2022.

[18] Q. Abbas, I. Qureshi, and M. E. Ibrahim, "An automatic detection and classification system of five stages for hypertensive retinopathy using semantic and instance segmentation in DenseNet architecture", *Sensors*, Vol. 21, No. 20, p. 6936, 2021.

[19] J. Li, R. Chen, C. Liu, X. Xu, and Y. Wang, "Capacity Optimization of Independent Microgrid with Electric Vehicles Based on Improved Pelican Optimization Algorithm", *Energies*, Vol. 16, No. 6, p. 2539, 2023.

[20] F. A. Zeidabadi, S. A. Doumari, M. Dehghani, and O. P. Malik, “MLBO: Mixed Leader Based Optimizer for Solving Optimization Problems”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, 2021, doi: 10.22266/ijies2021.0831.41.

[21] P. D. Kusuma and A. A. Dinimaharawati, “Fixed Step Average and Subtraction Based Optimizer”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 339-351, 2022.

[22] S. Doumari, H. Givi, M. Dehghani, and O. Malik, “Ring toss game-based optimization algorithm for solving various optimization problems”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 545–554, 2021, doi:10.22266/ijies2021.0630.46.

[23] F. A. Zeidabadi and M. Dehghani, “Poa: Puzzle optimization algorithm”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, 2022, doi:10.22266/ijies2022.0228.25.

[24] F. A. Zeidabadi, M. Dehghani, and O. P. Malik, “RSLBO: Random Selected Leader Based optimizer”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 529–538, 2021, doi:10.22266/ijies2021.1031.46.

[25] F. Zeidabadi, M. Dehghani, and O. Malik, “Timbo: Three influential members based optimizer”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 121–128, 2021, doi:10.22266/ijies2021.1031.12.

[26] P. D. Kusuma and A. L. Prasasti, “Guided pelican algorithm”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 179–190, 2022, doi:10.22266/ijies2022.1231.18.

[27] P. D. Kusuma and M. Kallista, “Stochastic komodo algorithm”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, 2022, doi:10.22266/ijies2022.0831.15.

[28] A. A. Alqarni, "Majority vote-based ensemble approach for distributed denial of service attack detection in cloud computing", *Journal of Cyber Security and Mobility*, pp. 265-278, 2022, doi: 10.13052/jcsm2245-1439.1126.

[29] https://www.kaggle.com/datasets/tawsifurrahman/tuberculosis-tb-chest-xray-dataset

[30] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam *et al.*, "Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization", *IEEE Access*, Vol. 8, pp. 191586-191601, 2020.