

Statistical Analysis of Pneumonia Affected Chest X-Ray for Covid-19 Detection and RNA Cell Analysis Using Machine Learning Architecture

Joly Varghese¹, K. Sivasankar²

¹*Research scholar Department of CSE, Noorul Islam Centre for Higher Education, Thuckaly, Kanyakumari District, Tamilnadu, India.*

²*Associate professor Department of IT, Noorul Islam centre for Higher Education, Thuckaly, Kanyakumari District, Tamilnadu, India.*

The disease COVID-19 has symptoms in the lungs and kills people all over the world. Diagnosis as well as treatment of this pandemic disease are subject of ongoing research. It is essential for human life that this disease be detected early. With deep learning-based diagnostic studies, this process is moving quickly. Subsequently, to add to this field, a profound learning-based method that can be utilized for early conclusion of sickness is proposed in our review. This study proposes a novel machine learning-based statistical analysis of a chest X-ray image from a patient with pneumonia for Covid-19 detection. For the purpose of RNA cell analysis, the input chest X-ray data was processed and segmented using dense alexnet-19 and structural gradient extreme value probability. After that, binary firefly moth optimization analysis was used to optimize the segmented image. Simulation results were evaluated in terms of receiver operating curve (ROC), sensitivity, and specificity for a variety of chest X-ray image datasets. Due to small COVID-19 dataset, effective proposed deep-learning blocks utilized various regularization methods to reduce over-fitting. proposed technique attained training accuracy of 95%, validation accuracy 97%, receiver operating curve (ROC) 63%, sensitivity of 62%, specificity of 68%.

Keywords: chest X-ray image, Covid-19 detection, machine learning techniques, dense alexnet-19, RNA cell analysis.

1. Introduction

The Coronavirus has developed into a pandemic, which has a significant global impact. It is believed that Wuhan, China, was the first location where the coronavirus was found [1]. The Coronavirus spread quickly all over the world. Health officials have attempted to contain virus as number of cases of disease it causes has increased rapidly. On December 1, 2019, in Wuhan, China, one of the patients displayed symptoms of viral pneumonia. A medical report published

in Lancet Journal also cites this patient, making it likely that this is the first COVID-19 case ever documented. Since then, there has been a chain reaction of pneumonia cases reported in Wuhan and around world [2]. Fever, labored breathing, sneezing, and coughing were most common symptoms observed in patients. The Chinese government reported a cluster of pneumonia cases in Wuhan to the World Health Organization (WHO) at the end of December. Wuhan city is associated with other Chinese urban communities through railroad and streets and with different nations through global air terminal. COVID-19 virus is carried by infected individuals who travel from Wuhan to other Chinese and international cities. The infection was sent when others were interacting with the tainted individuals. As of the beginning of June, approximately 6.2 million people had contracted the pandemic disease COVID-19. Consolidation and ground-glass opacities were found in between 50 and 60 percent of COVID-19 cases in children [3]. Automation of chest radiography interpretation has potential to undergo a paradigm shift using deep learning. Establishment of a referent data set [4], organ segmentation, removal of artifacts, multilabel classification, data augmentation, and grading of disease severity [5] are just a few of applications of deep learning that have been subject of more than 40,000 research articles. Reproducibility as well as comparability of research are made possible by the training and testing data set's accessibility, which is crucial to deep learning research. Transfer learning, which enables the reuse of previously trained methods in a specific application, is one common method in deep learning. According to the ImageNet database [6], established pretrained DNN have been trained to recognize thousands of objects using at least a million images.

2. Related works:

Using chest X-ray image data, the disease has been identified in numerous deep learning studies [7]. Three distinct deep learning models were used in a previous study to classify pneumonia X-ray images: fine-tuned method, method without fine-tuning, and method trained from scratch [8]. They also classified using the Multi-Layer Perceptron (MLP), and their accuracy was on average 82.2 percent [9]. Dataset is divided into 3 classes for their research: normal, viral, and bacterial pneumonia to adjust brightness as well as contrast zoom settings using augmentation technique for every dataset image. Highest level of classification success was 96.6% [10]. Pneumonia data were classified utilizing BPNN and Competitive Neural Network models in work [11]. They used normal and pneumonia chest X-ray images for 30% of dataset as test data as well as compared proposed method to existing CNNs. Their success in classification was 89.57 percent. In order to train data, author [12] suggested using a DL method to classify pneumonia data from scratch. Methods input size was 200 x 200 pixels in order to test sigmoid function's classification capabilities. From X-ray images, success rate for pneumonia was 93.73 percent. Work [13] used deep learning models to identify pneumonia images from three dataset classes: normal, pneumonia caused by a virus, and pneumonia caused by bacteria After that, they trained the models by applying augmentation method to every image as well as using transfer learning. Overall accuracy of the classification was 96.39%. The author [14] has classified patients according to whether or not they are infected with pneumonia using chest X-rays. Proposed method has a precision score of 0.843 and takes into account the 36 convolutional layers. work [15] have introduced the execution of move learning in Coronavirus recognition as there is an exceptionally restricted measure of

information accessible. Chest X-ray images for three distinct class values—COVID-19, pneumonia, and normal—were taken into consideration in a related work [16] to reduce the noise using the fuzzy technique and stacking. A new PDCOVIDNet framework with parallel-dilation in chest X-ray images was created utilizing CNN in work [17,18]. In parallel stage of proposed technique, authors used a spread convolution that could capture as well as extend features needed to achieve a precision of 96%. Decomposing, transferring, and composing (DeTraC) is a profound neurotic network that, according to the authors of [19,20], is capable of detecting chest x-ray images from COVID-19 patients.

3. System model:

In this section, a novel COVID detection method based on RNA cell analysis of chest X-ray images as well as ML is discussed for patients with pneumonia. Normalization and noise removal have been applied to the various input chest X-ray images. Based on RNA cell analysis, the processed image has been segmented for covid-19 detection, and binary firefly moth optimization analysis has been used to optimize the segmented image. Proposed method is depicted in figure 1.

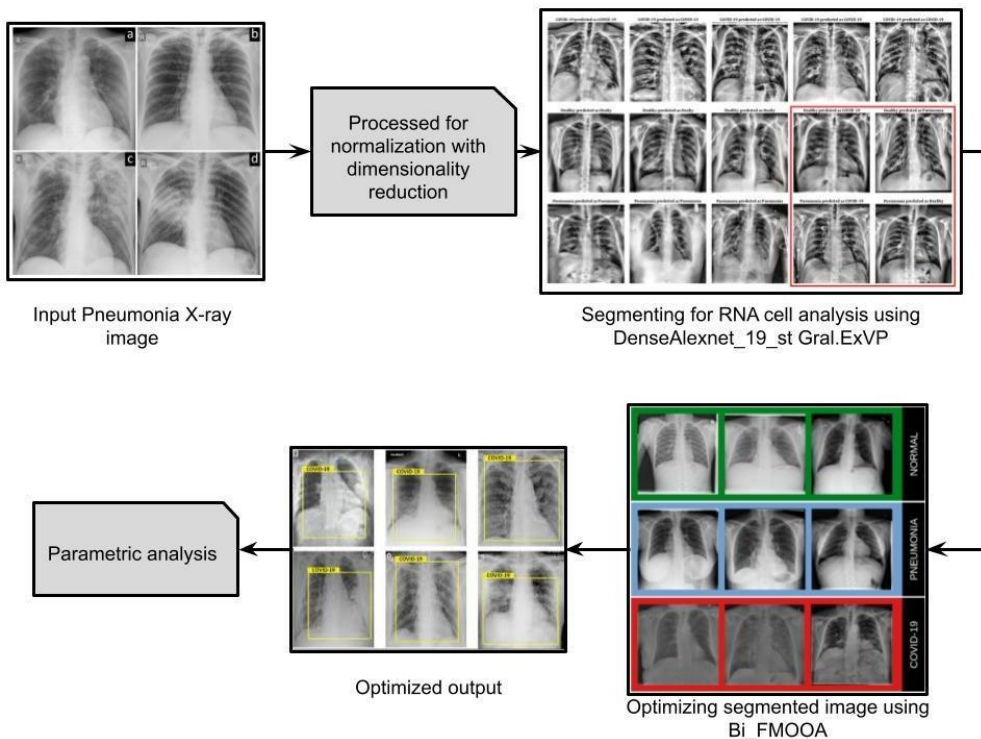


Figure 1: Proposed architecture

Using image contrast enhancement, contrast enhancement was performed on every individual image in original dataset during process of creating enhancement data setOne of the image processing techniques developed to address contrast enhancement issue is contrast

enhancement. It was utilized for first time on X-ray images in this study. Findings from both data sets will be examined as part of the study's proposed strategy. 70% of data set in experimental studies was utilized as training data, while 30% was utilized as test data.

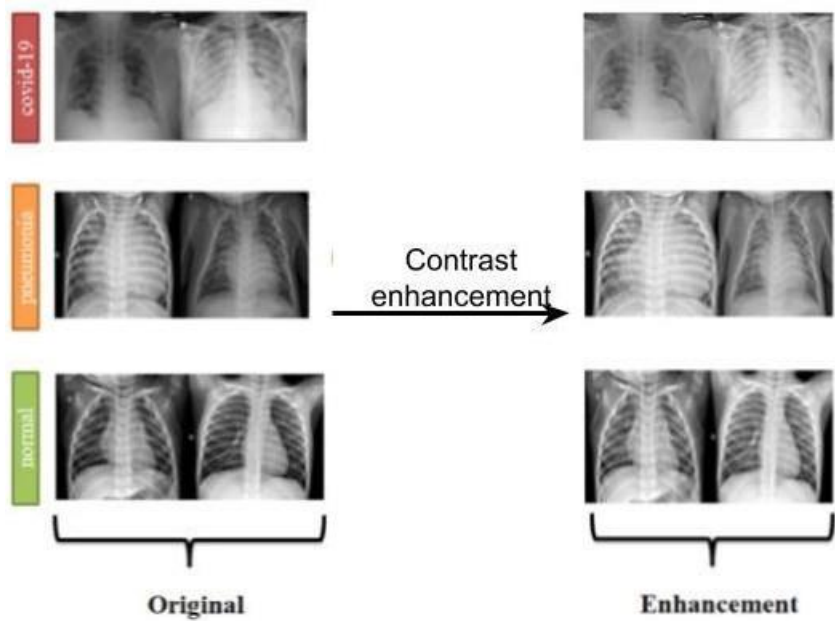


Figure-2 Dataset samples from original and enhancement data set

Performance of the supervised ML methods is largely determined by size of the available training data, which is highly data-dependent. It is often hard to create training datasets that are large enough. This is also true for the COVID-19 disease's previously collected data and images. Because of the speedy spread of this infection in a brief period, sadly, there hasn't been sufficient time such a long ways to gather adequate information on this illness for an enormous scope. Naturally, there are various approaches to this issue, including the utilization of pre-trained networks and data augmentation methods. On the CXR images, we used five data augmentation techniques: rotation, horizontal and vertical flips, scale, translation, and crop. In Figure. 3, several examples of these modifications are shown.

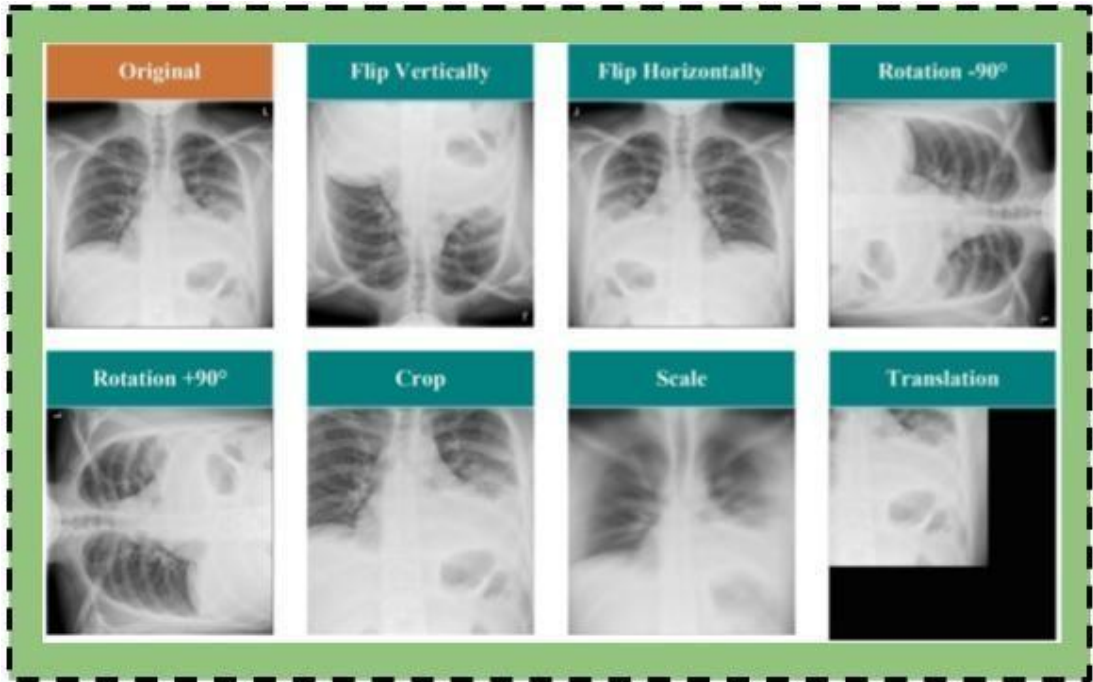


Figure 3 Sample modifications made to the traditional data augmentation process

Segmentation using structural gradient extreme value probability with dense alexnet-19 based RNA cell analysis:

The formula for gradient descent is as follows. Let $f: \mathbb{R}^N \rightarrow \mathbb{R}$ represent the objective function that one wants to reduce. Using the steepest descent of the objective function at the current point eq (1), one iteratively updates an initial point $\mathbf{x}^{(0)} \in \mathbb{R}^N$

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla f(\mathbf{x}^{(t)}) \quad (1)$$

Second derivatives of objective function, or information about curvature, are taken into account at every step in Newton's technique, which extends this method. As a result, the objective function's Hessian matrix $\mathbf{H}$ is included in the iterative update by eq. (2),

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \mathbf{H}^{-1} \nabla f(\mathbf{x}^{(t)}) \quad (2)$$

With $\mathbf{H} = \frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}^T}$ calculated at current point, $\mathbf{x}^{(t)}$

Since gradient's classical side data can spread anywhere in network's structure, it is utilized in error derivation and gradient computation calculations. A unitary operator is created by

applying the L units in order $U(\vec{\theta})$ as eq. (3)

$$U(\vec{\theta}) = U_L(\theta_L) U_{L-1}(\theta_{L-1}) \dots U_1(\theta_1) \quad (3)$$

where $U_i(\theta_i)$ identifies an i -th unitary gate, and $\theta \rightarrow$ is the gate specification vector $\theta^* = (\theta_1, \dots, \theta_{L-1}, \theta_L)^T$. evolution of system of Gradient for a particular input method $\psi \phi$, given by eq. (4)

$$|Y\rangle = U(\vec{\theta})|\psi\rangle \mid \phi\rangle = U(\vec{\theta})|z\rangle|1\rangle = U(\vec{\theta}) \mid z, 1\rangle \quad (4)$$

Y is the $(n+1)$ -length output system. We think about calculating gradient of loss function f as eq. (5) in order to assess the diffusion matrix:

$$f_i = \left\{ \sum_{\mu, \nu}^K (\theta^* \theta - \bar{\theta}^* \bar{\theta}) \text{Tr} (B \sigma^\mu \rho_{\text{in}}^i \sigma^\nu) + \eta \right\}^2 \quad (5)$$

using which we compute by eq. (6,7):

$$\left(\frac{\partial f_i}{\partial \theta_\zeta} \right) = \left\{ \sum_{\mu, \nu, \delta, \gamma}^K (\theta \theta - \bar{\theta}^* \bar{\theta}) (\bar{G}^\delta \theta + G^\gamma \theta^*) \text{Tr} (B \sigma^\mu \rho_{\text{in}}^i \sigma^\nu \otimes B \sigma^\delta \rho_{\text{in}}^i \sigma^\gamma) + 2\eta \sum_{\delta, \gamma}^K (G^\delta \theta + G^\gamma \theta^*) \text{Tr} (B \sigma^\delta \rho_{\text{in}}^i \sigma^\gamma) \right\}, \quad (6)$$

$$\left(\frac{\partial f}{\partial \theta_\zeta} \right) = \left\{ \sum_{\mu, \nu, \delta, \gamma, j, k, p, q}^K 2A^\infty (\theta \theta - \bar{\theta}^* \bar{\theta}) (G^\delta \theta + G^\gamma \theta^*) \text{Tr} (B \sigma^\mu \sigma^j |0\rangle \langle 0| \sigma^k \sigma^\nu \otimes B \sigma^\delta \sigma^p |0\rangle \langle 0| \sigma^q \sigma^\gamma) + 2\eta \sum_{\delta, \gamma, j, k}^K A^\infty (\bar{G}^\delta \theta + G^\gamma \theta^*) \text{Tr} (B \sigma^\delta \sigma^j |0\rangle \langle 0| \sigma^k \sigma^\gamma) \right\} \quad (7)$$

$$G_{wv}^\mu = \left(\frac{\partial g^\mu}{\partial w_v} \right) = \begin{cases} \sum_{l \in \{L(v)\}} \left(\frac{\partial g^\mu(w)}{\partial w_l} \right) \left(\frac{\partial w_l}{\partial w_v} \right) & \mu \neq v \\ 1 & \mu = v \end{cases} \quad (8)$$

$$G_v^\mu = \left(\frac{\partial g^\mu}{\partial w_v} \right) = \begin{cases} \sum_{l \in \{L(v)\}} \left(\frac{\partial g^\mu(w)}{\partial w_l} \right) \left(\frac{\partial w_l}{\partial g^v(w)} \right) & \mu \neq v \\ 1 & \mu = v \end{cases} \quad (9)$$

It is crucial to remember that, like weights across time, G varies with epochs. Matrix G measures dependence between several specifications, which are represented as coordinates. Given that G is a Dirac-delta function, specifications are independent, and parameter space is Euclidean. Matrix G regulates a parameter's dependency on other specifications, which in turn modifies the specification itself, seen from the perspective of a dynamical system. The magnitude of the matrix G_{jI} is proportional to hopping energy from lattice site I to j in manybody interactions with long-range hopping, which are related to this situation. When each matrix G element's hopping energy is high enough to weaken the disorder, ergodicity is shown. When the magnitude of each matrix G element is small enough to result in a minimal hopping energy and an increase in disorder strength, localization takes place and ergodicity is lost. X is formally defined as eq(9) as a result.

$$\checkmark = \{ c_i : \forall l \in \mathbb{I} : \chi(v, f(t)) = 0, \\ c_* : l^* \in fI : x(v^*, f^*(l^*)) = 0, \} \quad (10)$$

v^* and f^* (*) refer to the vertex and function at *, respectively, and * is a specific individual, vertex, or function. () is a compact constraint function.

$$\lim_{n \rightarrow \infty} \Pr \left(\frac{M_n - b_n}{a_n} \leq x \right) = G(x) \quad (11)$$

where $M_n = \max(X_1, \dots, X_n)$ and $X_1, \dots, X_n$ are n -variate i.i.d random variables with the cumulative density function (CDF) F and $\Pr$ is abbreviation of the probability function. Based on this, eq. (12) states that limiting distribution G belongs to GEVD:

$$G(x) = \exp \left(- \left(1 + \xi \left(\frac{x - \mu}{\sigma} \right) \right)^{-\frac{1}{\xi}} \right) \quad (12)$$

The main disadvantage of using these triple extreme value distributions is that modelling approach requires both an additional mechanism for confirming selection of the best model class and an analytical method for picking best method class. One way to solve this issue is by integrating these distributions into the GEVD model. However, the next hypothesis provides an irregular variable's asymptotic tail circulation when its true conveyance capability, or CDF F , is unknown. If Eq. (13) is valid, distribution function of $X-u$ under the assumption that $X > u$ is described by the GPD method as follows for a threshold level u_R :

$$H(y) = 1 - \left(1 + \frac{y}{\bar{\sigma}} \right)^{-\frac{1}{\bar{\sigma}}} \quad (13)$$

where $1 + y > 0$ and $y = X - u > 0$. Moreover, scale parameter of GPD is represented by $\bar{\sigma} > 0$. The fact that parameters of both extreme value distributions are highly meaningful is an important point to make. The fact that these parameters must be approximated using a computational method is another item to keep in mind. Maximum likelihood estimation is one of the useful techniques here (MLE). It is based on defining one and optimising a log-likelihood function with respect to the unidentified parameters.

It is possible to rewrite the log-likelihood function as eq. (15) if $\xi = 0$:

$$L(\mu, \sigma) = -k \log \sigma - \sum_{i=1}^k \left(\frac{x_i - \mu}{\sigma} \right) - \sum_{i=1}^k \exp \left(- \left(\frac{x_i - \mu}{\sigma} \right) \right) \quad (15)$$

The aforementioned equations can be optimised for the unknown parameters to find the GEVD's shape, position, and scale values. Characterizing the GPD model's log-probability capabilities enables a comparable interaction. Let X be a random variable with actual values. The th quantile is the smallest value x_R under the significance level, such that $P(X > x) = 1 - \alpha$, where $0 < \alpha < 1$, as shown in equation (16):

$$G_{(1-\alpha)}^{-1} := \inf \{ x \in \mathbb{R} : G(x) \geq \alpha \} \quad (16)$$

$$q_E = X_{([n\alpha])} \quad (17)$$

where "x" stands for the closest number that is bigger than or equal to x. Given that parametric quantile estimator is based on GEVD method, its CDF can simply be inversed to express it, as shown in Eq (2). In this case, MLE is used to evaluate unknown parameters of GEVD method once a few (sufficient) maximum samples (e.g., $X_1, \dots, X_k$) from all data points ($X_1, \dots, X_n$) have been chosen. Quantile function of GEVD under significance level is expressed by eq(18) in one of the following ways, depending on the shape parameter

$$q_G = \begin{cases} \mu - \frac{\sigma}{\xi} (1 - (-\log(1 - \alpha))^{-\xi}), & \xi \neq 0 \\ \mu - \sigma \log(-\log(1 - \alpha)), & \xi = 0 \end{cases} \quad (18)$$

where $q_G = G(1 - \alpha)^{-1}$. A combination of the parametric and non-parametric quantile functions is described as a new estimator using eq. (19):

$$q^- = \beta q_E + (1 - \beta) q_G \quad (19)$$

The input image in AlexNet needs to be enlarged to a predetermined size of $227 \times 227 \times 3$. With windows of 3×3 pixels, the final three layers are 384, 384, and 256 convolution filters. The five convolutional layers are supplemented by these additional layers. The network is then finished off with a single completely linked output layer that has 1000 output classes. Last but not least, completing computer vision tasks successfully requires the important procedures of Dropout, ReLU, and preprocessing.

Layer of Convolution: When a convolutional layer subjects an input to a filter, activation takes place. Filter is applied repeatedly to an input, and result is a feature map that displays intensity of found features at various input places. Multiple filtering can be used to construct a feature map, which can then be activated using techniques like ReLU. A dot product is frequently used as the operation between the two in convolutional layers since they require smaller filters than the input data.

The softmax function collaborates with the output layer to forecast the probability of each output class in a convolutional neural network. The softmax function, by definition, provides the same value to each neuron in output layer. Each of these neurons in the output layer produces an output, which is the likelihood of that node being output.

$$\Theta(z)_x = \frac{e^{\bar{v}_i}}{\sum_{j=1}^m e^{\bar{v}_j}} \quad (20)$$

Binary cross-entropy loss function is employed in this work with softmax serving as activation function. When dealing with binary classification issues, binary cross-entropy is typically used. Binary cross-entropy loss function for a network of n layers is shown in equations (21).

Optimization using binary firefly moth optimization analysis:

FA is defined by three main steps: (i) A firefly piques the interest of all other fireflies, regardless of gender. ii) The brightness of a fly is directly proportional to its magnetism. With low brightness, glowing fly will attract the fly. The fireflies are closer together when there is more light. iii) A fitness function is used to map the brightness of fireflies. Luminosity of a

firefly with origin brightness Y is given as eq. (1):

$$Y(s) = Y_0 e^{-\delta s} \quad (1)$$

where Y_0 denotes the source of brightness, s represents the distance between two fireflies, and is the light-related coefficient that determines the brightness and occupied of a space. As brightness and attractiveness are proportional, the equation (2) can be used to express attraction T .

$$T(s) = T_0 e^{-\delta s} \quad (2)$$

The attractiveness is T_0 when s is 0. The equation (3) represents attraction of Firefly l and m :

$$P_l^{z+1} = T_l e^{-\delta s_{lm}^z (P_m^z - P_l^z) + \phi (\text{Rand} - 0.5)} \quad (3)$$

Rand produces a random number between 0 and 1 where ϕ specifies parameter unpredictability, z is number of iterations, and The expression eq. (4) can be used to express distance between l th and m th Firefly, which is indicated by s_{lm} .

$$s_{lm} = \|P_m - P_l\| = \sqrt{\sum_{k=1}^K (P_{lk} - P_{mk})^2} \quad (4)$$

What's new? In this method, the equation for FBGO is given as eq (5)

$$l_i^{h+1} = \text{crossover}(l_1, l_2, l_3) \quad (5)$$

A suitable crossing between the solutions l , m , n and (l_1, l_2, l_3) , which are binary vectors illustrating effect of wolves migrating towards alpha, beta, and delta grey wolves, respectively, is $\text{crossover}(l, m, n)$. Equation (6) can be used to calculate (l_1, l_2, l_3)

$$l_1^t = \begin{cases} 1 & \text{if } (l_a^t + \text{biste}_a^t) \\ 0 & \text{Otherwise} \end{cases} \quad (6)$$

where binary step is given by $\text{biste } t$ in dimension t_a and the location vector in dimension t is indicated by l_1^t . It could be calculated using eq. (7)

where continuous value of size step is represented by $\text{costep } pt$ and rand is an integer chosen at random from a uniformly distributed range of $[0, 1]$. (8)

where c is continuously lowered in the context of Equation (10), X^* is the updated position of the prey, r^*1^* is the random distribution, and $(2,0)$. The prey's distances from each grey wolf are represented by D^*a in Equation (18), and the coefficient variable is represented by C^*1 . The binary step is represented by $\text{biste } t$ dimension t in Equation (11), while the location vector

in dimension t is indicated by lt .

MFO imitates the natural navigational system of moths [58]. This algorithm begins by assigning a population, N , and assigning possible solutions $I \ I = 1, 2, 3, \dots N$, with features dimension ij , ($j = 1, 2, 3, \dots \text{Dim}$). Search strategy is represented by eq. (12):

$$\Psi_t = (\Psi_{IP}, \Psi_{UP}, \Psi_{SP}) \quad (12)$$

where Ψ_{UP} stands for an updated phase, Ψ_{IP} stands for an initial phase, and SP stands for the stopping condition. The population is produced at random during the first phase, with N equal to 20. The following equation(13) uses the fitness function to evaluate the characteristics

$$\text{Fitness}(\theta) = r \times \text{Error}(\theta) + (1 - r) \times (|\Psi_i|) \quad (13)$$

where the fitness function is represented by $\text{Error}(\theta)$. Cubic is the chosen kernel function, and one-versus-all is the method; r is a random number between $[0, 1]$ utilized to balance classifier accuracy; θ_i stands for input features, and Ψ_i stands for selected features in the i -th iteration. The definition of the activation function is eq (14)

where V is the value that is updated after mean operation on every chosen feature set I is performed. In this equation, the number 1 stands for the feature that is chosen for the

following iteration and taken into account during the updated phase, and the value 0 for the feature that is eliminated from the following iteration. If this approach is used, it's possible that some important features may be dropped, but there's also a good chance that useful features will be carried over for the next version. The following eq(15) is used to update the features.

$$\Psi_i = \text{dielbcos}(2\pi l) + \text{Flame}(u) \times w + (1 - w) \times u, \quad (15) \text{ where } i, u = 1, 2, \dots, N, \text{di} = |\text{Flame}(u) - \Psi_i|, \text{ and } l = [1, -1] \text{ by eq. (16)}$$

Up until the end of all iterations, the fitness function assesses the characteristics given by this function one more. Iterations in this study were limited to 100. This approach was performed to both deep feature vectors and resulted in two optimised vectors. Optimised feature vectors have lengths of $N \times 1262$ and $N \times 826$, which are represented by the symbols Ψ_i respectively. In the end, we blended the two vectors to create a more helpful feature matrix. Although accuracy is feature fusion's main goal, computational time is also extended. We first calculate lengths of both vectors before classifying longest feature vector in order to execute feature fusion. Based on maximum length, we conducted padding and calculated the entropy of feature vectors with lesser lengths.

As a result, the length of the final vector must match that of the other feature vectors. The majority of the time, zero padding is applied to feature vectors of identical length. Following this, maximum correlation coefficient between pair characteristics $I(j)$ was calculated as shown in equations (18, 19):

$\Psi_i, \Psi_j \in \mathbb{R}$ are supremum functions, and the interval is $[1, 1]$, where a value of 1 denotes a high positive correlation, Y_e is a supremum function, and these values are: $\Psi_i, \Psi_j \in \mathbb{R}$. Hence, we only selected feature pairs that had a maximum correlation of one or very near to one. Using this approach, selecting feature pairs is essential for obtaining the optimal values. This feature selection procedure was continued until $Y(i, j)$ was computed for all feature pairings (1262 pairs). Final product was an improved breast cancer, and dimension of Ψ_k was $N \times 1632$.

The performance validation of a specific model was evaluated using the cross-validation method for a sample of future data. The data were divided into groups to accomplish this: the

testing set, which was used to calculate the results error ratio, and training set, which was used to apply method. Proposed method will benefit from this in order to achieve maximum performance.

Proposed optimization algorithm:

Start

Step 1: Describe fitness function: $h(x)$, $x = (x_1, x_2, \dots, x_d)$

Step 2: Produce initial population of fireflies P_l , where $l = 1, 2, 3, \dots, n$

Step 3: Evaluate Brightness $Y(s) = Y_0 e^{-\delta s}$

Step 4: Describe Absorption Coefficient δ

-While ($z < \text{MaxGeneration}$)

- for $l = 1: n$
- for $m = 1: l$
- if ($Y_m > Y_l$)
- Vary attractiveness with distance s via $e^{-\delta s}$
- Move firefly l towards m using

$$P_{z^{l+1}} = T e^{-\delta s} (P_z - P_z) + \varphi(\text{Rand} - 0.5)$$

$$l \quad 0 \quad l m \quad m \quad l$$
- Calculate new solutions as well as update brightness
- end for m
- end for l
- Determine latest best Firefly
- Entropy-based activation is applied $H(\Phi)$
- Best Optimal Features are Selected Φk
- end While
- Processing results as well as visualization End

4. **Results and discussion:**

Original data set is structured utilizing Fuzzy and Stacking techniques with Python 3.6. Plus, the proposed calculation was ordered in Python, and point by point data about source codes as well as examination utilized in this study are given in web connect determined in Open Source Code segment. The interfaces program used to compile Python is Jupyter Notebook. For classification, MATLAB (2019b) software was used with deep learning models. Windows 10 operating system (64 bit), an Intel i5-Core 2.5 GHz processor, and a 4 GB memory card are hardware features needed to compile software.

Description of dataset: Covid-chestxray-dataset, COVID-CT, COVID-chestxray-dataset, Actualmed-COVID-chestxray-dataset, SARS-COV-2 Ct-Scan Dataset, COVID-19 Xray Dataset (Train & Test Sets) with COVID-19 CNN Pneumonia Detector, COVID-19 Radiography Database, and COVID-19 Patients Lungs X Ray Images 10,000 were all COVID-19 Patients Lungs X Ray Images 10,000.

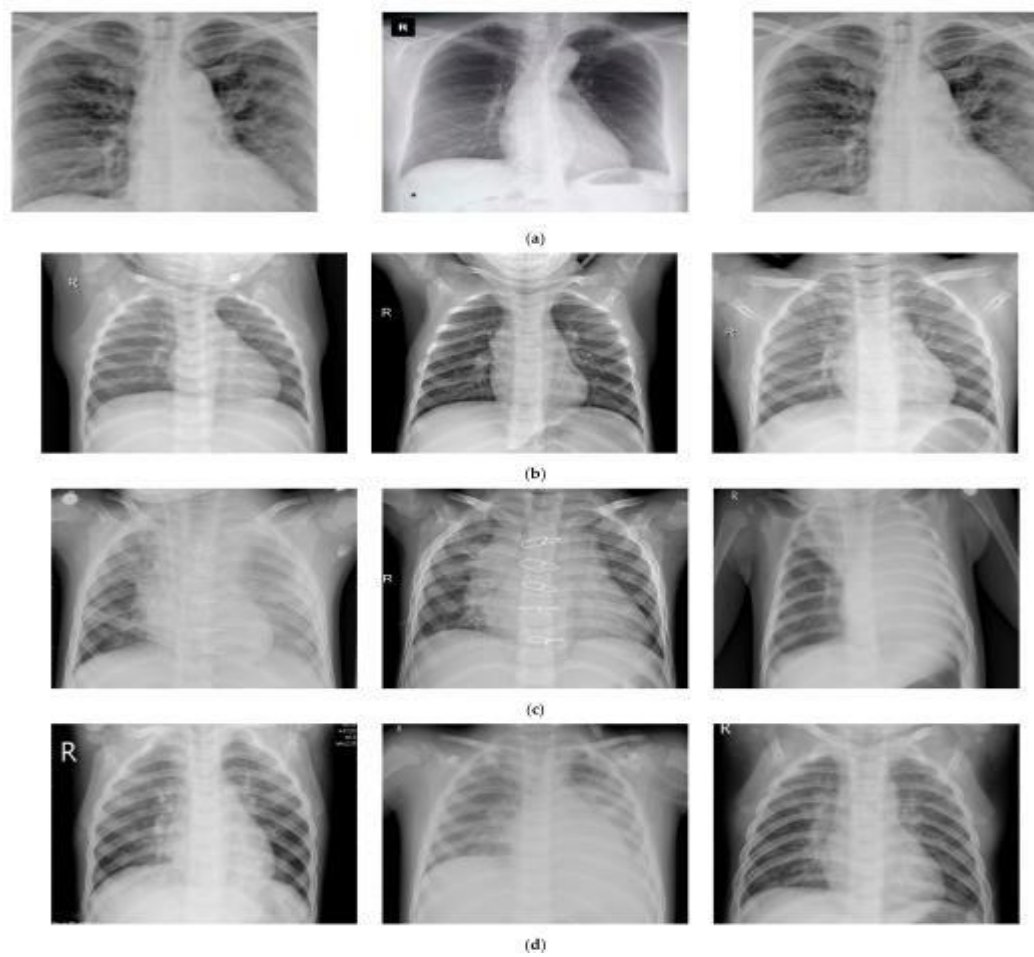


Figure 4. Sample of collected dataset: (a) normal images, (b) bacterial pneumonia, (c) viral pneumonia, and (d) Coronavirus-infection images.

1203 normal images, 660 bacterial pneumonia infections, and 931 viral pneumonia infections are included in this dataset. These archives were accessed for a total of 1251 images. 1251 images were used in benchmark paper, and we utilized 1251 images in this study for the same reason. With a 72 dpi resolution, collected images were resized to 224 x 224 pixels. Dataset details are given in Table 1 and Figure 2.

Table 1. Dataset details.

Sr. No.	Disease	Total Samples	Training Samples	Testing Samples
1	Normal images	310	248	62
2	Pneumonia-bacterial-infection images	330	264	66
3	Viral-pneumonia images	327	261	66
4	Corona-infection images	284	227	57

We selected one image from each class and added it to figures. The experimental results section discusses these images. Fig. 2 depicts P-A view of a COVID patient's chest radiograph, which reveals hyperlucent lung fields that indicate hyperinflation of lungs as a result of bronchiolitis or bronchitis-caused obstruction of the small airways. Many patchy opacities in both lung fields suggest consolidations brought on by bacterial co-infection. No hyperlucency, air-liquid level, or increased horse vascular marks are present in the centre.

Proposed analysis:

Table-2 Proposed technique based parametric analysis

Dataset	Training accuracy	Validation accuracy	ROC	Sensitivity	Specificity
COVID-chestxray	89	91	55	52	61
Actualmed-COVID-chestxray	92	93	59	55	63
SARS-COV-2 Ct-Scan	93	95	61	59	65
COVID-19 Xray	95	97	63	62	68

Table-2 shows parametric analysis for proposed technique based on various COVID- X-ray dataset. here the dataset analysed are COVID-chestxray-dataset, Actualmed-COVID-chestxray-dataset, SARS-COV-2 Ct-Scan Dataset, COVID-19 Xray Dataset in terms of training accuracy, validation accuracy, receiver operating curve (ROC), sensitivity, specificity.

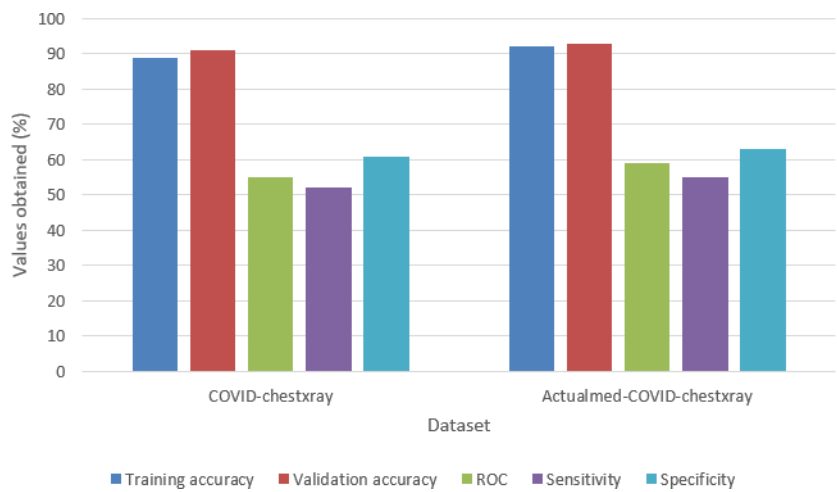


Figure-5 Parametric analysis of COVID-chestxray-dataset, Actualmed-COVID-chestxray-dataset for proposed technique

The above figure-5 shows parametric analysis based on COVID-chestxray-dataset, Actualmed-COVID-chestxray-dataset. proposed technique attained training accuracy of 89%, validation accuracy 91%, receiver operating curve (ROC) 55%, sensitivity of 52%, specificity of 61% for COVID-chestxray-dataset, for Actualmed-COVID-chestxray-dataset proposed technique attained training accuracy of 92%, validation accuracy 93%, receiver operating curve (ROC) 59%, sensitivity of 55%, specificity of 63%.

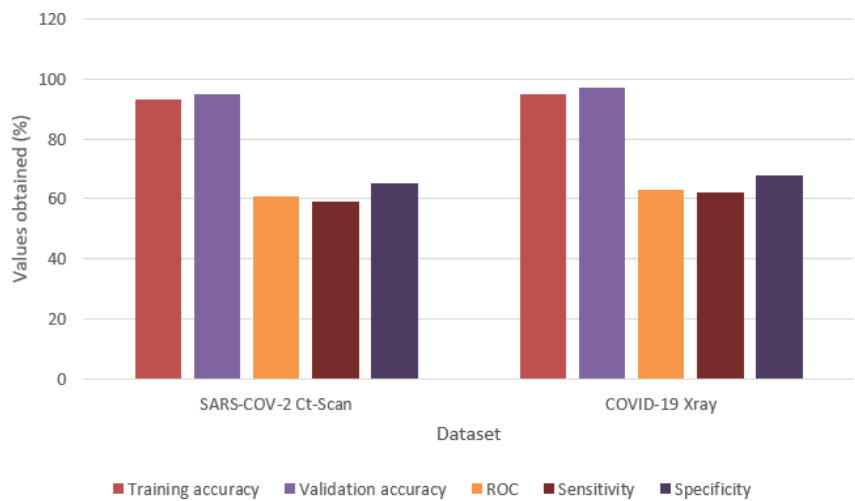


Figure-6 Parametric analysis of SARS-COV-2 Ct-Scan Dataset, COVID-19 Xray Dataset for proposed technique

From above figure 6 shows parametric analysis based on SARS-COV-2 Ct-Scan Dataset, COVID-19 Xray Dataset based on proposed technique. the proposed technique attained training accuracy of 93%, validation accuracy 95%, receiver operating curve (ROC) 61%, sensitivity of 59%, specificity of 65% for SARS-COV-2 Ct-Scan Dataset, for COVID-19 Xray

Dataset proposed technique attained training accuracy of 95%, validation accuracy 97%, receiver operating curve (ROC) 63%, sensitivity of 62%, specificity of 68%.

We have employed batch normalisation to increase model's stability by employing consistent input sizes. Updates to the weights, the cross-entropy loss function, and the selected learning rate were all performed using the suggested optimizer. The hardest part of creating the model for us was combining all of layers, activation functions, and optimizer values to obtain best accuracy and other assessment criteria from our suggested model. Best outcomes from model are what we are looking to identify. Our suggested model outperforms the published and preset models that we described in experimental result section, despite fact that the majority of existing approaches have deeper layers and more parameters than our model.

Comparative analysis:

Table-3 comparative analysis between proposed and existing technique based on various COVID X-ray dataset

Dataset	Techniques	Training accuracy	Validation accuracy	ROC	Sensitivity	Specificity
COVID-chestxray	MLP	81	88	51	45	58
	PDCOVIDNet	83	89	53	48	59
	SAP_RNA_MLA	89	91	55	52	61
Actualmed-COVID-chestxray	MLP	91	89	55	51	59
	PDCOVIDNet	92	92	59	53	62
	SAP_RNA_MLA	92	93	59	55	63
SARS- COV-2 Ct-Scan	MLP	91	91	59	55	59
	PDCOVIDNet	92	93	61	58	64
	SAP_RNA_MLA	93	95	61	59	65
COVID-19 Xray	MLP	91	92	62	55	65
	PDCOVIDNet	93	94	63	59	66
	SAP_RNA_MLA	95	97	63	62	68

Table-3 shows analysis based on various COVID- X-ray dataset. here the dataset analysed are COVID-chestxray-dataset, Actualmed-COVID-chestxray-dataset, SARS-COV-2 Ct-Scan Dataset, COVID-19 Xray Dataset in terms of training accuracy, validation accuracy, receiver operating curve (ROC), sensitivity, specificity.

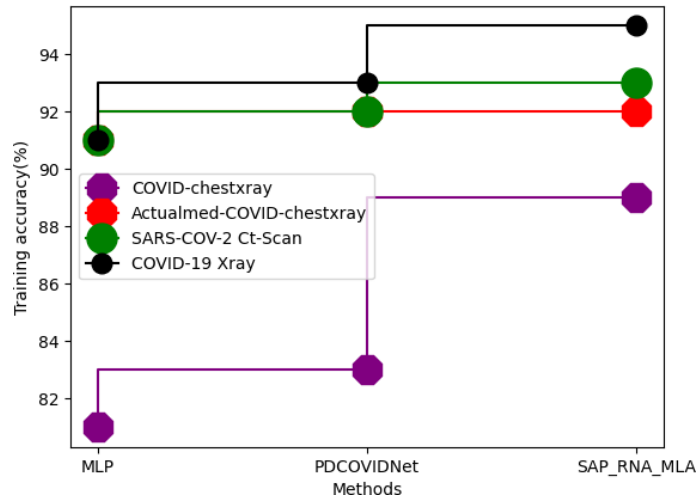


Figure-7 Comparison of Training accuracy

The above figure 7 shows comparison of training accuracy based on proposed and existing technique. proposed technique attained training accuracy of 89%,existing MLP attained training accuracy of 81%, PDCOVIDNet attained training accuracy of 83% for COVID-chestxray dataset, for Actualmed-COVID-chestxray dataset proposed technique attained training accuracy of 92%,existing MLP attained training accuracy of 91%, PDCOVIDNet attained training accuracy of 92%, proposed technique attained training accuracy of 93%,existing MLP attained training accuracy of 91%, PDCOVIDNet attained training accuracy of 92% for SARS-COV-2 Ct-Scan, for COVID-19 Xray dataset the proposed technique attained training accuracy of 95%,existing MLP attained training accuracy of 91%, PDCOVIDNet attained training accuracy of 93%

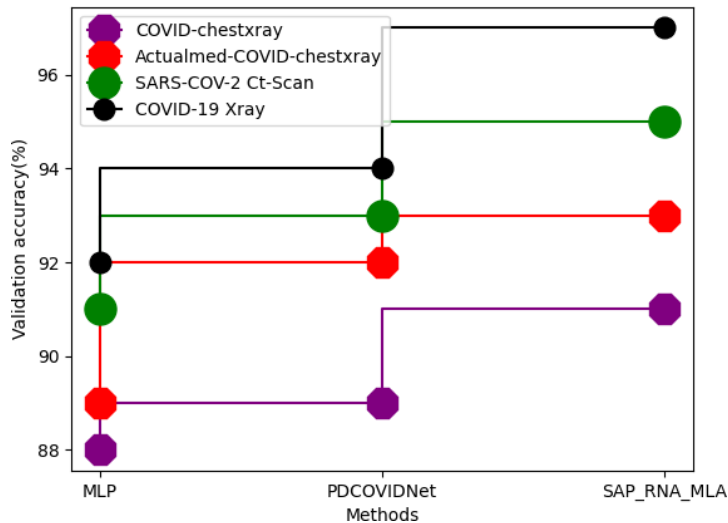


Figure-8 Comparison of Validation accuracy

From above figure-8 analysis in terms of Validation accuracy. proposed technique attained Validation accuracy of 91%,existing MLP attained Validation accuracy of 89%, PDCOVIDNet attained Validation accuracy of 90% for COVID-chestxray dataset, for Actualmed-COVID-chestxray dataset the proposed technique attained Validation accuracy of 93%,existing MLP attained Validation accuracy of 89%, PDCOVIDNet attained Validation accuracy of 92%, the proposed technique attained Validation accuracy of 95%,existing MLP attained Validation accuracy of 91%, PDCOVIDNet attained Validation accuracy of 93% for SARS-COV-2 Ct-Scan, for COVID-19 Xray dataset the proposed technique attained Validation accuracy of 97%,existing MLP attained Validation accuracy of 92%, PDCOVIDNet attained Validation accuracy of 94%.

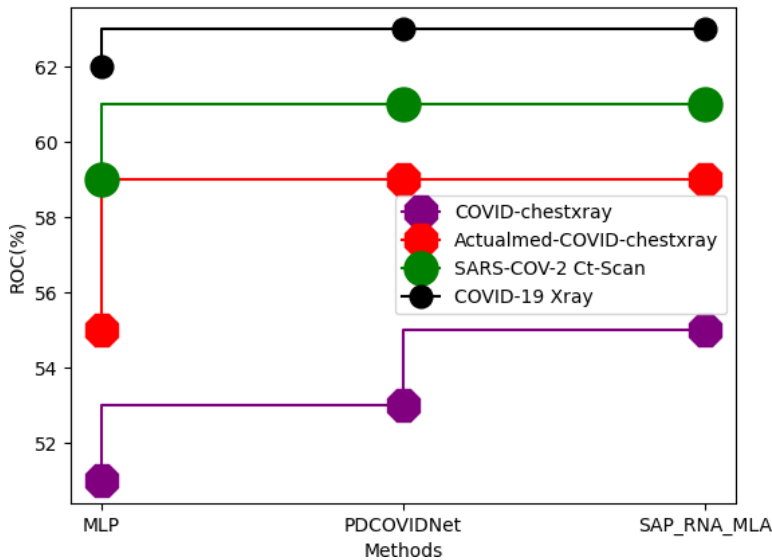


Figure-9 Comparison of ROC

The above figure 9 shows comparison of ROC. proposed technique attained ROC of 55%,existing MLP attained ROC of 51%, PDCOVIDNet attained ROC of 53% for COVID-chestxray dataset, for Actualmed-COVID-chestxray dataset the proposed technique attained ROC of 59%,existing MLP attained ROC of 55%, PDCOVIDNet attained ROC of 59%, the proposed technique attained ROC of 61%,existing MLP attained ROC of 59%, PDCOVIDNet attained ROC of 61% for SARS-COV-2 Ct-Scan, for COVID-19 Xray dataset the proposed technique attained ROC of 63%,existing MLP attained ROC of 62%, PDCOVIDNet attained ROC of 63%.

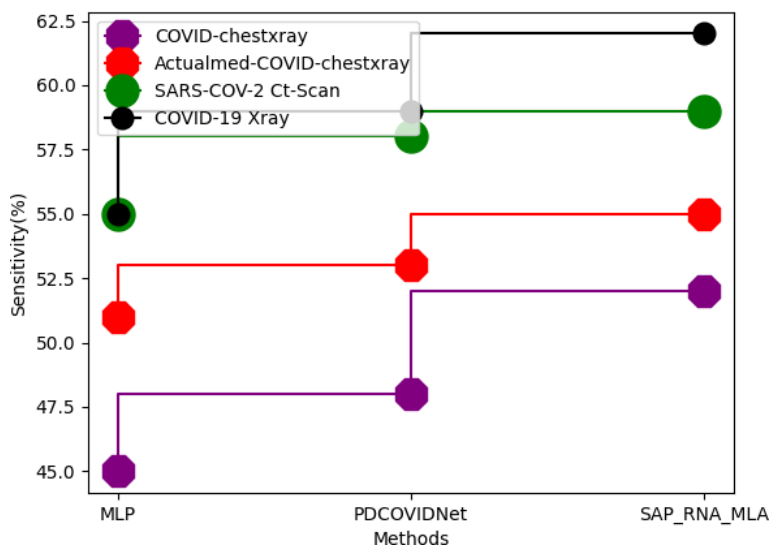


Figure-10 Comparison of Sensitivity

From above figure-10 analysis in terms of Sensitivity. the proposed technique attained Sensitivity of 52%,existing MLP attained Sensitivity of 45%, PDCOVIDNet attained Sensitivity of 48% for COVID-chestxray dataset, for Actualmed-COVID-chestxray dataset the proposed technique attained Sensitivity of 55%,existing MLP attained Sensitivity of 51%, PDCOVIDNet attained Sensitivity of 53%, the proposed technique attained Sensitivity of 59%,existing MLP attained Sensitivity of 55%, PDCOVIDNet attained Sensitivity of 58% for SARS-COV-2 Ct-Scan, for COVID-19 Xray dataset the proposed technique attained Sensitivity of 62%,existing MLP attained Sensitivity of 55%, PDCOVIDNet attained Sensitivity of 59%.

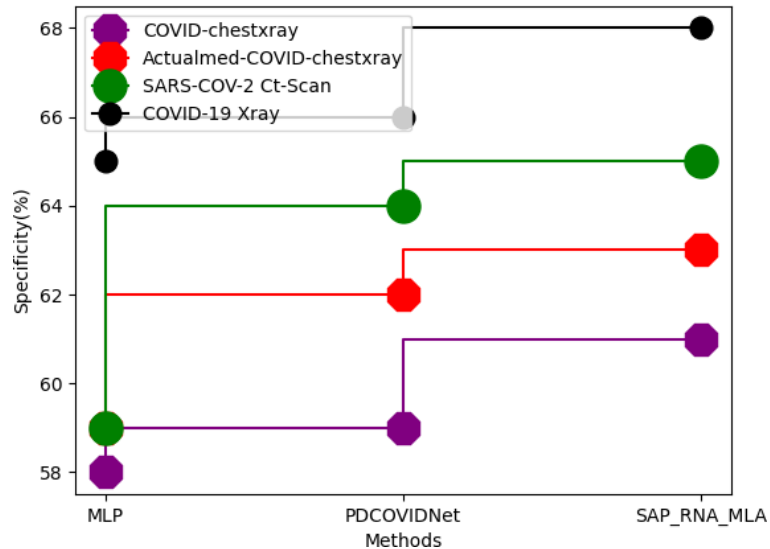


Figure-11 Comparison of Specificity

the above figure 11 shows comparison of Specificity. Proposed technique attained Specificity of 61%,existing MLP attained Specificity of 58%, PDCOVIDNet attained Specificity of 59% for COVID-chestxray dataset, for Actualmed-COVID-chestxray dataset the proposed technique attained Specificity of 63%,existing MLP attained Specificity of 59%, PDCOVIDNet attained Specificity of 62%, the proposed technique attained Specificity of 65%,existing MLP attained Specificity of 59%, PDCOVIDNet attained Specificity of 64% for SARS-COV-2 Ct-Scan, for COVID-19 Xray dataset the proposed technique attained Specificity of 68%,existing MLP attained Specificity of 65%, PDCOVIDNet attained Specificity of 66%.

After that, proposed convolution layer with an image size of 224 x 224 pixels is added after the base layer is frozen. Here, RMSprop streamlining agent and ReLU actuation capability are likewise utilized. The initial residual convolution receives input from the (2, 2) kernel size. Third residual convolution layer is fed with a kernel size of (2, 2), stride of 1, maximum pooling, dropout, and RMSprop optimizer. Fourth and fifth residual convolution layers are fed with kernel sizes of (1, 1), maximal pooling, dropout, and no optimizer in order to combat overfitting. Then, on images with kernel sizes of (3, 3), RMSprop optimizer, and 224 x 224 pixels, the proposed layer, RMSprop optimizer, and ReLU activation function are utilized. Calculation's final step is to determine calculation's precision, accuracy, f1 score, and review of outcome.

A pre-trained network has already undergone training on a larger dataset, which is typically sufficient to produce a distinctive hierarchy from which characteristics can be extracted. It performs better with very small datasets. Except for the novel approach, Keras provides pre-packaged versions of the utilized and described models. Keras-tune is used to first tune all models to find the best hyperparameter ranges. A well-liked approach for parameter tuning is the grid search method. At first, the following were selected at random: All deep learning models explicitly choose the following variables: Size of hidden layer: 8–96 neurons Activation = ('ReLU' (hidden), 'Sigmoid' (final)) Dropout = 0.1 Kernel Dimensions: 2 2 Also, all deep learning models were able to attain the parameter values utilising the grid search strategy. Batch size = (50, 100, 150, 200, 250, 300, 350, 400) Learning rate = (0.0000001,

0.000001, 0.00001, 0.000

Number of batches ranged from 50 to 300. ImageNet loads were utilized to prepare every classifier. Number of dropout sections was changed to prevent overfitting. For upgraded execution, every one of the eleven models with the exception of the clever model ages, streamlining agents, bunch size and learning rate were picked utilizing network search. Performance scores were obtained by repeating training and testing with selected epochs as well as adjusting parameters. Next, by doing training and testing with a various epoch as well as adjusted hyperparameters, performance scores were achieved.

The best ten epochs were recorded and analysed, and the loss function was optimised for each model using epochs (100–400) and optimizers at a learning rate of 0.000001. Images were downscaled to the models' preferred size—256 256 for AlexNet—before being fed into the neural network. Remaining 20% of data was designated for testing, with remaining 80%

designated for training.

5. Conclusion:

This study proposes a novel machine learning-based technique for Covid-19 detection. For RNA cell analysis, structural gradient extreme value probability with dense alexnet-19 is used to perform the segmentation, and binary firefly moth optimization analysis is used to optimize the results. The most recent Covid-19 weekly case data were then used to examine the data distribution, and its parameters were derived using statistical distributions. In addition, a machine learning-based time series prediction model was proposed for predicting the epidemic tendency and the disease curve. The literature's findings demonstrate the significance of deep learning in fight against COVID-19 outbreak. proposed technique attained training accuracy of 95%, validation accuracy 97%, receiver operating curve (ROC) 63%, sensitivity of 62%, specificity of 68%. Strengthening the data set can improve the success ratio in future studies. Lung tomography can be utilized notwithstanding chest radiographs. Both performance and success ratio can be improved through the creation of various deep learning models.

References

1. Jalil, Z., Abbasi, A., Javed, A. R., Badruddin Khan, M., Abul Hasanat, M. H., Malik, K. M., & Saudagar, A. K. J. (2022). Covid-19 related sentiment analysis using state- of-the-art machine learning and deep learning techniques. *Frontiers in Public Health*, 9, 2276.
2. Sunitha, D., Patra, R. K., Babu, N. V., Suresh, A., & Gupta, S. C. (2022). Twitter sentiment analysis using ensemble based deep learning model towards COVID-19 in India and European countries. *Pattern Recognition Letters*, 158, 164-170.
3. Stone, D., Michalkova, L., & Machova, V. (2022). Machine and deep learning techniques, body sensor networks, and Internet of Things-based smart healthcare systems in COVID-19 remote patient monitoring. *American Journal of Medical Research*, 9(1), 97-112.
4. Chang, Y. C., Ku, C. H., & Le Nguyen, D. D. (2022). Predicting aspect-based sentiment using deep learning and information visualization: The impact of COVID- 19 on the airline industry. *Information & Management*, 59(2), 103587.
5. Zvarikova, K., Horak, J., & Bradley, P. (2022). Machine and Deep Learning Algorithms, Computer Vision Technologies, and Internet of Thingsbased Healthcare Monitoring Systems in COVID-19 Prevention, Testing, Detection, and Treatment. *American Journal of Medical Research*, 9(1), 145-160.
6. Gulati, K., Kumar, S. S., Boddu, R. S. K., Sarvakar, K., Sharma, D. K., & Nomani, M. Z. M. (2022). Comparative analysis of machine learning-based classification models using sentiment classification of tweets related to COVID-19 pandemic. *Materials Today: Proceedings*, 51, 38-41.
7. Mohan, S., Solanki, A. K., Taluja, H. K., & Singh, A. (2022). Predicting the impact of the third wave of COVID-19 in India using hybrid statistical machine learning models: A time series forecasting and sentiment analysis approach. *Computers in Biology and Medicine*, 144, 105354.
8. Lyu, Z., & Takikawa, H. (2022). Media framing and expression of anti-China sentiment in COVID-19-related news discourse: An analysis using deep learning methods. *Heliyon*, 8(8), e10419.

9. Arya, V., Mishra, A. K. M., & González-Briones, A. (2022). Analysis of sentiments on the onset of COVID-19 using machine learning techniques. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 11(1), 45-63.
10. Shahzad, A., Zafar, B., Ali, N., Jamil, U., Alghadhbhan, A. J., Assam, M., ... & Eldin, E. T. (2022). COVID-19 vaccines related user's response categorization using machine learning techniques. *Computation*, 10(8), 141.
11. Chandrasekaran, G., & Hemanth, J. (2022). Deep learning and TextBlob based sentiment analysis for coronavirus (COVID-19) using twitter data. *International Journal on Artificial Intelligence Tools*, 31(01), 2250011.
12. Khanday, A. M. U. D., Rabani, S. T., Khan, Q. R., & Malik, S. H. (2022). Detecting twitter hate speech in COVID-19 era using machine learning and ensemble learning techniques. *International Journal of Information Management Data Insights*, 2(2), 100120.
13. Ahmad, M., Sadiq, S., Alluhaidan, A. S., Umer, M., Ullah, S., & Nappi, M. (2022). Industry 4.0 technologies and their applications in fighting COVID-19 pandemic using deep learning techniques. *Computers in biology and medicine*, 145, 105418.
14. Kuo, K. M., Talley, P. C., & Chang, C. S. (2022). The accuracy of machine learning approaches using non-image data for the prediction of COVID-19: A meta- analysis. *International journal of medical informatics*, 104791.
15. Shyni, H. M., & Chitra, E. (2022). A comparative study of X-ray and CT images in COVID-19 detection using image processing and deep learning techniques. *Computer Methods and Programs in Biomedicine Update*, 100054.
16. Nistor, A., & Zadobrischi, E. (2022). The influence of fake news on social media: analysis and verification of web content during the COVID-19 pandemic by advanced machine learning methods and natural language processing. *Sustainability*, 14(17), 10466.
17. Constantinou, M., Exarchos, T., Vrahatis, A. G., & Vlamos, P. (2023). COVID-19 classification on chest X-ray images using deep learning methods. *International Journal of Environmental Research and Public Health*, 20(3), 2035.
18. Oğuz, Ç., & Yağanoğlu, M. (2022). Detection of COVID-19 using deep learning techniques and classification methods. *Information Processing & Management*, 59(5), 103025.
19. Ahemad, M. T., Hameed, M. A., & Vankdothu, R. (2022). COVID-19 detection and classification for machine learning methods using human genomic data. *Measurement: Sensors*, 24, 100537.
20. Zhang, Y., Chen, K., Weng, Y., Chen, Z., Zhang, J., & Hubbard, R. (2022). An intelligent early warning system of analyzing Twitter data using machine learning on COVID-19 surveillance in the US. *Expert Systems with Applications*, 198, 116882.