Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9107430, 17 pages
https://doi.org/10.1155/2022/9107430

Research Article

PCA-Based Incremental Extreme Learning Machine (PCA-IELM) for COVID-19 Patient Diagnosis Using Chest X-Ray Images

Vinod Kumar,[1] **Sougatamoy Biswas**,[1] **Dharmendra Singh Rajput**,[2] **Harshita Patel**,[2] **and Basant Tiwari**[3]

[1]*Koneru Lakshmaiah Education Foundation, Vaddeswaram, India*
[2]*Vellore Institute of Technology, Vellore, India*
[3]*Hawassa University, Awasa, Ethiopia*

Correspondence should be addressed to Basant Tiwari; basanttiw@hu.edu.et

Received 25 March 2022; Accepted 29 April 2022; Published 4 July 2022

Academic Editor: Muhammad Ahmad

Novel coronavirus 2019 has created a pandemic and was first reported in December 2019. It has had very adverse consequences on people's daily life, healthcare, and the world's economy as well. According to the World Health Organization's most recent statistics, COVID-19 has become a worldwide pandemic, and the number of infected persons and fatalities growing at an alarming rate. It is highly required to have an effective system to early detect the COVID-19 patients to curb the further spreading of the virus from the affected person. Therefore, to early identify positive cases in patients and to support radiologists in the automatic diagnosis of COVID-19 from X-ray images, a novel method PCA-IELM is proposed based on principal component analysis (PCA) and incremental extreme learning machine. The suggested method's key addition is that it considers the benefits of PCA and the incremental extreme learning machine. Further, our strategy PCA-IELM reduces the input dimension by extracting the most important information from an image. Consequently, the technique can effectively increase the COVID-19 patient prediction performance. In addition to these, PCA-IELM has a faster training speed than a multi-layer neural network. The proposed approach was tested on a COVID-19 patient's chest X-ray image dataset. The experimental results indicate that the proposed approach PCA-IELM outperforms PCA-SVM and PCA-ELM in terms of accuracy (98.11%), precision (96.11%), recall (97.50%), F1-score (98.50%), etc., and training speed.

1. Introduction

The World Health Organization (WHO) identified COVID-19 (virus known as SARS-CoV-2) as a worldwide pandemic in February 2020. This triggered never expected countermeasures, such as the closure of cities, districts, and foreign travel. Coronaviruses (CoV) are death-defying viruses that may cause severe acute respiratory syndrome (SARS-CoV). Various researchers and institutions have attempted an effective solution from different possible diminutions in encountering the COVID-19 pandemic. Multimedia dataset (audio, picture, video, etc.) is booming in a massive amount of text information as civilization enters the information era. Image classification has become more essential as the need for real-world vision systems grows [1] and has recently attained a lot of attention from many researchers. It has evolved into one of the most essential operations, serving as a requirement for all other image processing operations. Image classification using learning algorithms is a special open issue in image processing that has sparked a lot of interest due to its promising applications. In general, an image categorization system has two primary processes. The first stage is to create an effective image representation that has enough information about the image to allow for classification further. The second step is to use a good classifier to classify the new image. Thus, there are two major challenges to consider when improving picture classification performance: dimensionality reduction and classifier. Apart from computer vision and image operation, one of the most important stages in image classification is feature extraction which determines the invariant characteristic of images when using computer devices to assess and deal with image data.

In a practical scenario, feature extraction has been applied in many fields like historic structures, medical image processing, remote image sensing, etc. The image's essential lower-level qualities include color, texture, and shape. The color feature has globality, which may be retrieved using tools such as the color histogram, color set, and color moment. It might simply explain the proportions of different colors across the image. The useful characteristic is color for identifying photos that are difficult to distinguish automatically, and the spatial variation should be ignored. However, it is unable to explain the image's local distribution as well as the description of the distinct colors' spatial positions. Image classification with feature extraction using incremental extreme learning machines is proposed in this paper. Firstly, on the COVID-19 dataset of chest X-ray images, features were extracted from an image using PCA. Eventually, the SVM, ELM, and IELM are applied to image classification [2] once the dimension is reduced by PCA method. Different metrics were employed to achieve the robust evaluation: classification accuracy, recall, precision, F-score, true-negative rate (TNR), true-positive rate (TPR), AUC, G-mean, precision-recall curve, and receiver operating characteristics (ROC) curve.

The paper is arranged in the following sequence: several related approaches have been discussed in Section 2. The suggested technique is described and critiqued in Section 3. Section 4 contains a description of PCA and feature extraction techniques. Subsections 4.1–4.6 contain different algorithmic approaches that are compared with the proposed method. In Section 5, the proposed method and algorithm have been discussed. Section 6 describes the different evaluation criteria that are used. Section 7 discusses the experimental setup that has been used. Section 8 describes the dataset. Finally, Section 9 discusses the experimental results, and the research is concluded.

2. Related Works

The content of image features comprises color, texture, and other visual elements. The extracted content from visual features is the main component for analyzing the image. In this segment, some of the earlier work based on PCA and other feature extraction techniques along with different classification techniques has been discussed.

Sun et al. [3] suggested an image classification system based on multi-view depth characteristics and principal component analysis. In this method, depth features are extracted from the image, and from RGB depth, characters are independently extracted and PCA is applied to reduce dimension. The Scene15 dataset, Caltech256 dataset, and MIT Indoor datasets are used in the evaluation process. Eventually, the SVM [4] is used to classify images. The method's performance is demonstrated by the experimental results.

Mustaqeem and Saqib [5] suggested a hybrid method that is based on PCA and SVM. PROMISE (KC1: 2109 observations, CM1: 344 observations) data from NASA's directory have been used for the experiment. The dataset was divided into two parts: training (KC1: 1476 observations, CM1: 240 observations) and testing (KC1: 633 observations, CM1: 104 observations). Principal components of the features are extracted by PCA, and it helps in dimensionality reduction and minimizing time complexity.

In addition to this, SVM is used for further classification, and for hyperparameter tuning, GridSearchCV is used. From this, precision, recall, F-measure, and accuracy for KC1 dataset analysis are 86.8%, 99.6%, 92.8%, and 86.6%, respectively, and for CM1 dataset analysis, precision, recall, F-measure, and accuracy are 96.1%, 99.0%, 97.5%, and 95.2%, respectively. Similarly, Castaño et al. [6] provide a deterministic approach for starting ELM training based on hidden node parameters with activation function. The hidden node parameters are calculated with the help of Moore–Penrose generalized inverse, whereas the output node parameters are recovered through principal component analysis. Experimental validation with fifteen well-known datasets was used to validate the algorithm. The Bonferroni–Dunn, Nemenyi, and Friedman tests were used to compare the results obtained. In comparison with later ELM advancements, this technique significantly reduces computing costs and outperforms them.

Mateen et al. [7] suggested VGG-19 DNN-based DR model with better performance than AlexNet and the spatial invariant feature transform (SIFT) in terms of classification accuracy and processing time. For FC7-SVD, FC7-PCA, FC8-SVD, and FC8-PCA, respectively, classification accuracies are 98.34%, 92.2%, 98.13%, and 97.96% by using SVD and PCA feature selection with fully connected layers.

Zhao et al. [8] suggested extreme learning machines with no iteration along with supervised samples are used for model building as a class incremental extreme learning machine. The algorithm is shown to be stable and has almost equivalent accuracy of batch learning. Similarly, Huang and Chen [9] proposed an algorithm that analytically calculates hidden nodes' output after randomly producing and adding computational nodes to the hidden layer as a convex incremental extreme learning machine. Using a convex optimization, the existing hidden node output is calculated again. This can converge faster while maintaining efficiency and simplicity.

Zhu et al. [10] proposed a principal component analysis (PCA)-based categorization system with kernel-based extreme learning machine (KELM). Based on the resultant output, this model achieves better accuracy than SVM and other traditional classification methods. For the classification of HSIs, Kang et al. [11] developed the PCA-EPF extraction approach. In this research work, they have proposed the combination of PCA and standard edge preserving filtering (EPF)-based feature extraction. The proposed method achieves better classification accuracy with limited training samples. Similarly, Perales-González et al. [12] introduced a new ELM architecture based on the negative correlation learning framework dubbed negative correlation hidden layer ELM (NCHL-ELM). This model shows better accuracy when compared with other classifications by integrating a parameter into each node in the original ELM hidden layer.

Based on fractal dimension technology, Li et al. [13] suggested an enhanced ELM algorithm (F-ELM). By reducing the dimension of the hidden layer, the model improves in training speed. From the experimental results, it can be concluded that as compared to the standard ELM technique, the suggested algorithm significantly reduces computing time while also improving inversion accuracy and algorithm stability.

Because of the complexity of the data models, deep learning is incredibly pricey to train. Furthermore, deep learning necessitates the use of high-priced GPUs and hundreds of computer machines. There is no simple rule that can help you choose the best deep learning tools since it necessitates the understanding of topology, training technique, and other characteristics, whereas the simple ELM is a one-shot computation with a rapid learning pace. But the biggest advantage in IELM is the ability to randomly increase hidden nodes incrementally and analytically fix the output weights. The output error of the IELM rapidly diminishes as the number of hidden neurons increases.

In our method, SVM, ELM, and IELM based on the PCA technique are employed for image classification [14] for COVID-19 patient detection using the COVID-19 chest X-ray dataset. A summary of the most recent and related research works is described in Table 1 [3, 5–13].

3. Proposed Methodology

The back propagation (BP) approach is commonly used to train multi-layer perceptron (MLP). Various algorithms can be used to train this typical architecture. Gradients and heuristics are two types of algorithms that are commonly used. These algorithms have a few things in common: they have a hard time dealing with enormous amounts of data, and they have a slow convergence rate in these situations. Huang et al. (Huang et al.) [15] introduced the extreme learning machine as a solution to this problem.

The typical computing time required to train an SLFN using gradient-based techniques is reduced by this algorithm. The ELM, on the other hand, has several flaws. The randomly generated input weights and bias for ELM [16] result in some network instability. In case if there are outliners in the training data, then the hidden layer's output matrix will have ill-conditioned problems and it results in low generalization performance and lower forecasting accuracy. There are two types of ELM called fixed ELM and IELM [17]. In comparison with the ELM, the output error of the IELM rapidly diminishes and it tends toward zero with the growth in number of hidden neurons (Huang et al.) [15]. In online continuous learning regression and classification problem, this approach is very prominent (Xu and Wang; Zhang et al.) [18, 19].

A trained classifier can be obtained after training the classifiers with a sufficient amount of image data and then fed into the trained classifier for observation and analysis.

4. Feature Extraction

A single feature cannot describe the image feature and quality properly. The image classification will not yield acceptable results unless distinguishing features are described. Three images corresponding to three viewpoints are placed on each RGB color image. Our method uses PCA to extract the image's important information and minimize the input dimension [20–23].

4.1. Classification of Images and PCA Feature Extraction.

Extracting useful features from an image is a prominent task in image classification, and principal component analysis (PCA) is used for this purpose. PCA uses orthogonal transformation and converts variables to fewer independent components than the original variables. The output data with this approach will not lose important data features, and PCA loadings can be used for the identification of important data. A multivariate statistical analysis approach is used by PCA, which can perform linear transformation of numerous variables to pick a few key variables. PCA transforms data using eigenvectors from N-dimension to M-dimension where $M < N$. The new features are a linear mixture of the old ones, allowing them to capture the data's intrinsic unpredictability with little information loss. Figure 1 reveals the steps of the proposed model.

Suppose that the research object has p indexes, these indexes are regarded as p random variables and represented as X_1, X_2, , X_p. With this, new indexes are created by combining p random variable $F_1, F_2, ..., F_p$, which can mirror the data from the original indexes [24]. The independent replacement indexes reflect the original indexes' essential information.

$$\begin{aligned} F_1 &= a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p, \\ F_2 &= a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p, \\ &\quad \cdots\cdots\cdots, \\ F_p &= a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p. \end{aligned} \tag{1}$$

The following are the PCA stages in detail:

(1) *Data standardization*: The following calculation formula is used to standardize the matrix X:

Table 1: Similar work summarization.

SN	References	Applied method	Problem approached	Resulted outcome	Impediments
1	Sun et al. [3]	PCA of multi-view deep representation	Image classification	Comparison result from different databases	Limited classifiers are compared
2	Mustaqeem and Saqib [5]	Principal component-based support vector machine	Software defect detection	Better accuracy than other methods	No probabilistic explanation for SVM classification
3	Castaño et al. [6]	Pruned ELM approach based on principal component analysis	Classification	ELM model based on PCA	Limited classifiers are compared
4	Mateen et al. [7]	VGG-19 architecture with SVD and PCA	Fundus image classification	Better accuracy than other methods	Limited to nonimbalance data
5	Zhao et al. [8]	IELM	Activity recognition	Stable and similar accuracy to the batch learning method	Limited to batch learning
6	Huang et al. [9]	Convex incremental extreme learning machine	Convergence rate of IELM	Faster convergence rate	Limited classifiers are compared
7	Zhu et al. [10]	PCA and kernel-based ELM	Side-scan sonar image classification	Better classification accuracy with stable model	Classify underwater targets only
8	Kang et al. [11]	PCA-based edge-preserving features (EPF)	Hyperspectral image classification	Better accuracy than SVM	Parameters of EPFs are given manually
9	Perales-González et al. [12]	Negative correlation hidden layer for the ELM	Regression and classification	Better accuracy	Variety in the transformed feature space
10	Li et al. [13]	Improved ELM	Transient electromagnetic nonlinear inversion	Improves the inversion accuracy and stability	Less implementation in other industrial domains

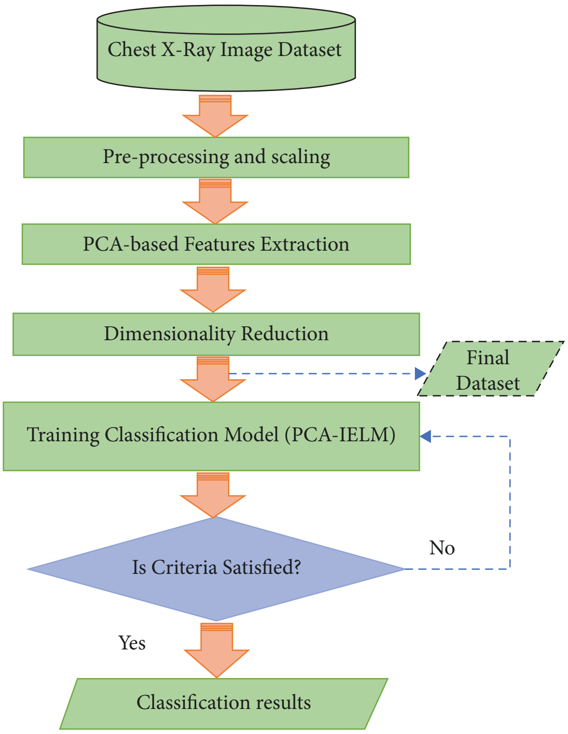

Chest X-Ray Image Dataset
Pre-processing and scaling
PCA-based Features Extraction
Dimensionality Reduction
Final Dataset
Training Classification Model (PCA-IELM)
No
Is Criteria Satisfied?
Yes
Classification results

Figure 1: Flowchart of the proposed model (PCA-IELM).

$$y_{\mathrm{ij}} = \frac{x_{\mathrm{ij}} - \overline{x}_j}{\sqrt{\mathrm{Var}\left(x_j\right)}}, \tag{2}$$

where $X = \{x_{ij}\}$, $Y = \{y_{ij}\}$, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$,

$$\begin{aligned} \overline{x} &= \frac{1}{n}\sum_{i=1}^{n} x_{\mathrm{ij}}, \\ \mathrm{var}\,(\mathrm{xj}) &= \frac{1}{n-1}\sum_{i=1}^{n}\left(x_{\mathrm{ij}} - \overline{x}_j\right)^2. \end{aligned} \tag{3}$$

(2) The following formula is used to solve the correlation coefficient matrix R:

$$R = \frac{Y^T Y}{n-1}. \tag{4}$$

(3) The following formula is used to calculate the eigenvalue and eigenvector of the coefficient matrix:

$$\left|R - \lambda I_p\right| = 0. \tag{5}$$

The calculated eigenvector is $a_i = (a_{i1}, a_{i2}, \ldots, a_{ip})$, where $i = 1, 2, 3, 4, \ldots\ldots\ldots\ldots, p$, and the eigenvalue is $\lambda\ i$ $(i = 1, 2, \ldots, p)$. To get a collection of main components Fi, the eigenvalues are sorted in descending order:

$$F_i = a_{i1}Y_1 + a_{i2}Y_2 + \cdots + a_{ip}Y_p. \tag{6}$$

(4) The following are the main factors to consider kth primary component contribution rate and expressed as

$$\lambda_k\left(\sum_{j=1}^{p}\lambda_j\right)^{-1}. \tag{7}$$

The rate of the first k primary components' cumulative contribution is expressed as

$$\sum_{j=1}^{k} \lambda_j \left(\sum_{j=1}^{P} \lambda_j \right)^{-1}. \tag{8}$$

The first principal component, F_1, is the one with the highest variance out of all the combinations of Y_1, Y_2, ..., Y_p; the second principal component F_2 is one with the highest variance among all the combinations of Y_1, Y_2, ..., Y_p, and they have no relation with F_1.

4.2. SVM. Several algorithms have been implemented and suggested in machine learning to solve the classification problem. Among the different classification problems, support vector machine (SVM) is one of the supervised algorithms in machine learning with [5, 25] the advantages as follows:

(i) It employs L2 regularization to overcome overfitting problems.

(ii) Even with minimal data, provide suitable findings.

(iii) Different kernel functions to match the features' complicated functions and interactions.

(iv) Manages the data nonlinearity.

(v) The model is stable thanks to the hyper-plane splitting rule.

(vi) Analyzes the data with a high degree of dimensionality.

Instead of focusing on decreasing prediction error, SVM focuses more on optimizing classification decision boundaries, which is why the hyper-plane is used to separate classes. If the data dimension is n and the hyper-plane is a $(n-1)$ vector function, then it can be represented mathematically as follows:

$$y = w_0 x_0 + w_1 x_1 + \cdots + w_{n-1} x_{n-1} + b. \tag{9}$$

It also signifies, in a broader sense,

$$y = w^T x + b, \tag{10}$$

where x denotes the input feature vector, w is the weight vector, and b is the bias. By adjusting w and b, several hyper-planes can be created, but the hyper-plane with the best margin will be chosen. The largest feasible perpendicular distance between each class and the hyper-plane is defined as ideal margin. The cost function or objective function is minimized to get the best margin. The cost function may be written as follows:

$$J(w) = \frac{1}{2}\|w\|^2 + \frac{1}{n}\sum_{i=0}^{n} \max\left(0, \left(1 - y_i * \left(w^T x + b\right)\right), \tag{11}$$

Even if the predictions are right and the data are correctly categorized by hypothesis, SMV utilized to penalize any y_i that are close to the borders $(0 < y_i < 1)$. The main goal is to figure out optimal w value to minimize $J(w)$, so differentiating Eq. 11 concerning w, we get the gradient of a cost function as follows:

$$\begin{aligned} \nabla_w J(w) &= \frac{\partial J(w)}{\partial w}, \\ &= \frac{1}{n}\sum_{i=0}^{n} \begin{cases} w, & \text{if } \max\left(0, \left(1 - y_i * \left(w^T x + b\right)\right), \\ w - y_i x_i, & \text{otherwise.} \end{cases} \end{aligned} \tag{12}$$

As far as we have calculated $\nabla_w J(w)$, weights of w can be updated as

$$W_{\text{new}} = W_{\text{old}} - \alpha[J(w)]. \tag{13}$$

We go through the procedure again and again until smallest $J(w)$ discovered. Because data are rarely linearly separable, we must sketch a decision boundary between the classes rather than using a hyper-plane to separate them. We will need to convert (13) into a decision boundary to deal with the dataset's nonlinearity:

$$y = w \cdot \phi(x) + b. \tag{14}$$

$\phi(x)$ is the kernel function in (14). There are various types of kernel functions that may be used to create SVM, such as linear, polynomial, and exponential, but we will use the radial basis function in this model (RBF). Distance parameter that is used is Euclidean distance, and the smoothness of the borders is defined by the parameter σ.

$$\phi(x) = \exp\left(-\frac{x - \overline{x}^2}{2\sigma^2}\right), \tag{15}$$

where $x - \overline{x}^2$ is the square of Euclidean distance between any single observation x and mean of the training sample x.

4.3. PCA-SVM. The motive of the support vector machine (SVM) [3] is to find the best possible hyper-plane that will separate two planes on the training set. The coefficient of the hyper-plane is w that we have to project. It uses structural risk minimization theory to build the best hyper-plane segmentation in the feature space and a learning editor to achieve global optimization.

Assume the training data, (x_1, y_1), $(x_2, y_2), \ldots, (x_n, y_n) \in R^n$, $y \in \{-1, 1\}$.

This could be projected into a hyper-plane:

$$(\omega \cdot x) + b = 0, \omega \in R^n b \in R. \tag{16}$$

For the normalization,

$$y_i(\omega \cdot x_i) + b \geq 1, \quad i = 1, 2, \ldots, l. \tag{17}$$

The classification of the interval is equal to $2/\omega$, when the maximum interval is equal to the minimum ω^2.

Before classifying the data through SVM, the necessary features from the image data need to be extracted. The high-dimensional data can be converted to the low-dimensional data with this approach. For this, the PCA method as a feature extraction through convergence matrix and eigenvalue proportion calculation is used. PCA-based SVM is the method that is used for classification and regression. After that, SVM is used to classify low-dimensional data. Figure 2

```
     Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\mathbf{t_i} \in R^C$.
     Output: SVM classification model.
(1)  Procedure SVM.
(2)  Initialize weights $W$ and bias $b$ with any arbitrary number.
(3)  Feature optimization.
(4)  TrainDataSet, TestingDataSets = Data.Split(Ratio).
(5)  Define $y$.
(6)  Define $f(y)$.
(7)  Define $J(W)$.
(8)  Calculate $\nabla_w J(w)$.
(9)  Repeat for minimum $J(W)$:
         Update $W$.
         Call steps 6–8.
         End.
(10) Calculate accuracy.
(11) Return accuracy.
(12) End procedure.
     Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\mathbf{t_i} \in R^C$.
     Output: SVM classification model.
```

Algorithm 1: SVM algorithm.

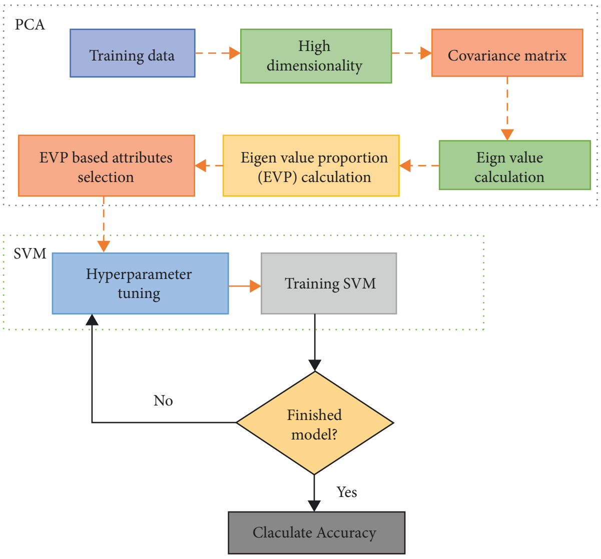


Figure 2: Flowchart of PCA-SVM.

depicts the working flow of PCA-SVM. Once the parameter optimization is done, the model is ready to predict categorization.

4.4. Extreme Learning Machine (ELM).

An extreme learning machine is a single hidden layer feedforward network that can be used for both classification and regression. In ELM [26], weights between the input layer, hidden layer, and biases are randomly generated. The output weights are calculated using the generalized Moore–Penrose pseudo-inverse. ELM performs faster than other feedforward networks [27] and outperforms other iterative methods. Figure 3 shows the basic network architecture of ELM.

Suppose $[\mathbf{x_i}, \mathbf{t_i}]$ denotes N training samples, wherein training instances i ϵ 1, 2, 3, ..., N and $\mathbf{x_i} = [x_{i1}, x_{i2}, \ldots, x_{im}]T$ ϵ R^m denotes i^{th} training instance and its desired output $\mathbf{t_i} = [t_{i1}, t_{i2}, \ldots, t_{iC}]^T$ ϵ R^C.

Let the number of input features and number of neurons be equal and represented by m; similarly, let L be the number of hidden neurons. The number of output neurons and number of classes are equal and denoted by c. Figure 4 [24] shows the flowchart of the principal component analysis [28]. The input weight matrix is represented by $\mathbf{U} = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_j}, \ldots \mathbf{u_L}]^T \in R_{L\times m}$, and the hidden neuron bias is represented by $\mathbf{b} = [b_1, b_2, \ldots, b_j, \ldots b_L]^T \in RL$. $\mathbf{u_j} = [u_{j1}, u_{j2}, \ldots u_{jm}]$ are the connecting weights between the j^{th} hidden neuron with the input neurons. Bias of the j^{th} hidden neuron is bj, and jth hidden layer output for i^{th} instance is represented by

$$h_{ij} = g\left(\mathbf{u_j}\mathbf{x_i} + b_j\right). \tag{18}$$

Here, activation function is represented by g. For all the training instances hidden layer output is represented by $\mathbf{H}$ and can be represented by

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{1L} \\ \cdots & \cdots & \cdots \\ h_{j1} & h_{j2} & h_{jL} \\ \cdots & \cdots & \cdots \\ h_{N1} & h_{N2} & h_{NL} \end{pmatrix}. \tag{19}$$

Between the hidden layer and the output layer, the output weight $\boldsymbol{\beta}$ can be computed using Eq. (20). Linear activation function is used by the output layer in this computation.

$$\beta = \mathbf{H}^{\dagger}\mathbf{T}. \tag{20}$$

Here,

```
     Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\in R^C$.
     Output: PCA-SVM model for classification.
 (1) Procedure PCA-SVM
 (2) Identify the relationship among features through a covariance matrix.
 (3) Through the linear transformation or eigendecomposition of the covariance matrix, we get eigenvectors and eigenvalues.
 (4) Transform our data using eigenvectors into principal components.
 (5) Quantify the importance of these relationships using eigenvalues and keep the important principal components.
 (6) Data extracted from PCA and will be given as input.
 (7) Initialize weights $W$ and bias $b$ with any arbitrary number.
 (8) Feature optimization.
 (9) TrainDataSet, TestingDataSets = Data.Split(Ratio).
(10) Define $y$.
(11) Define $f(y)$.
(12) Define J($W$).
(13) Calculate $\nabla_w J(w)$.
(14) Repeat for minimum J($W$):
     Update $W$.
     Call steps 10–13.
     End.
(15) Calculate accuracy.
(16) Return accuracy.
(17) End procedure.
```

Algorithm 2: PCA-SVM algorithm.

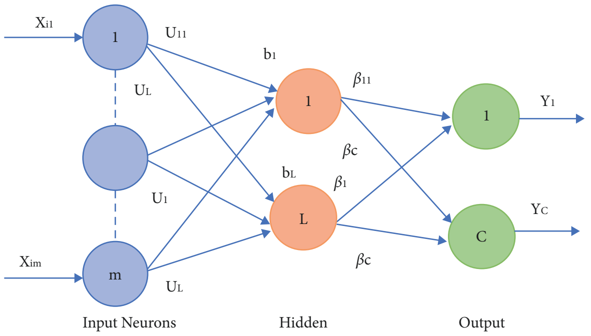


Figure 3: Network architecture of ELM.

$$\beta = \left[\beta_1^T, \ldots, \beta_j^T, \ldots, \beta_L^T\right]_{L\times C}^T. \quad (21)$$

The vector $\boldsymbol{\beta_j} = [\beta_{j1}, \ldots, \beta_{jk}, \ldots, \beta_{jC}]^T$, where $j = (1, 2, 3, \ldots \ldots \ldots, L)$ represents the connecting weights between the j^{th} hidden neuron and the k^{th} output neuron. The predicted outcome of all the output neurons for all training instances is represented as

$$\mathbf{f}(\mathbf{x}) = \mathbf{H}\beta. \quad (22)$$

Here, the output function is $\mathbf{f}(\mathbf{x}) = [f_k(\mathbf{x}), \ldots, f_C(\mathbf{x})]$. From Eq. 23, label for class $\mathbf{x}$ can be predicted.

$$\text{label}(\mathbf{x}) = \arg\max f_k(\mathbf{x}), \quad k = 1, \ldots, C. \quad (23)$$

4.5. PCA-ELM : Classification Method Based on PCA-ELM. In the PCA technique [6], variables are first scaled. The different steps of PCA that has been applied in PCA-ELM are

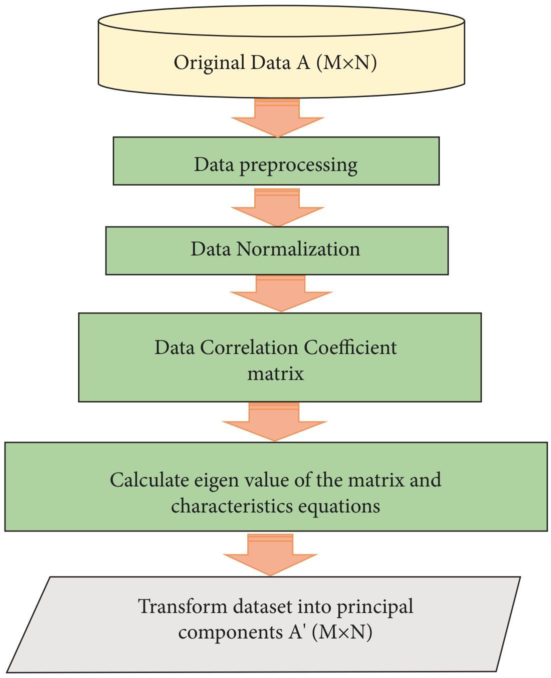

Original Data A (M×N)
Data preprocessing
Data Normalization
Data Correlation Coefficient matrix
Calculate eigen value of the matrix and characteristics equations
Transform dataset into principal components A' (M×N)

Figure 4: Flowchart of the principal component analysis [24].

(1) Scaling of trained data.

(2) Covariance matrix evaluation.

(3) Eigenvalues for the covariance matrix along with eigenvectors are defined.

(4) Evaluating the principal components.

The output from PCA is given as an input to ELM [29]. The process of PCA-ELM [30] is shown in Figure 5.

```
Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m, \in R^C$.
Output: ELM classification model.
ELM procedure.
Hidden biases $\mathbf{b}$ and input weights $\mathbf{U}$ randomly selected.
From (19), $\mathbf{H}$ as hidden layer output is defined.
From (20), $\boldsymbol{\beta}$ is defined as the weights between the hidden layer and the output layer.
Return $\boldsymbol{\beta}$.
```

Algorithm 3: ELM algorithm.

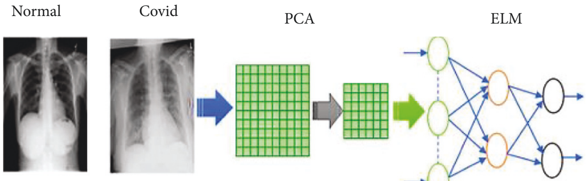


Figure 5: Process of PCA-ELM.

```
Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m, \in R^C$.
Output: PCA-ELM model for classification.
Procedure PCA-ELM.
Identify the relationship among features through a covariance matrix.
Through the linear transformation or eigendecomposition of the covariance matrix, we get eigenvectors and eigenvalues.
Transform our data using eigenvectors into principal components.
Quantify the importance of these relationships using eigenvalues and keep the important principal components.
Extracted principal components given as the input data.
Hidden biases $\mathbf{b}$ and input weights $\mathbf{U}$ randomly selected.
From (19), hidden layer output $\mathbf{H}$ can be computed.
From (20), weights between the hidden and the output layer computed as $\boldsymbol{\beta}$.
Return $\boldsymbol{\beta}$.
End procedure.
```

Algorithm 4: PCA-ELM algorithm.

4.6. ELM. Compared to the other neural networks, the ELM learns faster as there is no need to adjust hidden nodes and provides better generalization capability. But there are various flaws with the ELM. Randomly generated bias and input weights in ELM network [31] are results in some network instability. Training data outliers from the hidden layer's output matrix result in poor network generalization performance. In comparison to the ELM, the output error of the IELM rapidly diminishes and resolves the issue of very small weights of output and validity of hidden layer neurons. In online continuous learning, it is appropriate for regression and classification tasks.

The IELM [32] network model structure is shown in Figure 6. Suppose the size of input, hidden nodes, and outputs are m, l, and n, respectively, and ω_i is the input weight matrix with $l \times m$ dimension of the current hidden layer neuron and uniformly distributed between random numbers $[-1, 1]$. The bias of the ith hidden node b_i is a random number between $[-1, 1]$ uniformly distributed, the activation function for the hidden layer neuron is sigmoid function given by (24), and output weight matrix β is with $l \times n$ dimension.

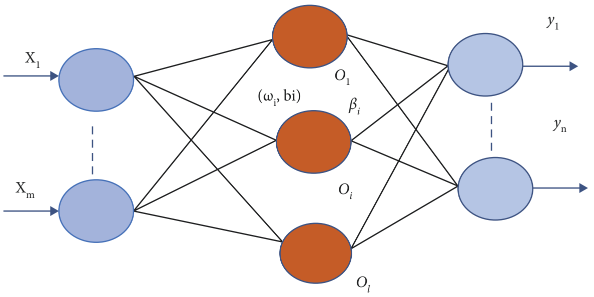


Figure 6: The structure of IELM.

The hidden node activation function (sigmoid) is given by

$$g(x) = \frac{1}{1 + e^{-x}}, \tag{24}$$

where x is the input matrix.

A matrix X is of $m \times N$ dimension, and it represents N dataset input. Y is a $n \times N$ matrix that represents the output where N datasets for a training set $\{(X, Y)\}$. Training steps of IELM algorithm are described as follows:

Step 1. In the initialization phase, suppose $l = 0$ and L is the maximum number of the hidden nodes. Output Y is defined in terms of the initial value of the residuals E (difference between target and actual error) is set to be the and ε is the expected training accuracy.

Step 2. Training phase, while $l < L$ and $E > \varepsilon$

(1) Hidden nodes l will be increased by 1, i.e.,

$$l = l + 1. \tag{25}$$

(2) Hidden layer neuron O_l is evaluated randomly from input weights ω_l and bias b_l.

(3) Output of the activation function $g\ (x')$ is calculated for the node O_l (b_l needs to be extended into a $l \times N$ vector b_l).

$$x' = \omega_l x + b_l. \tag{26}$$

(4) Hidden layer neuron output vector $\overline{H}$ can be calculated from

$$\overline{H} = g(x'). \tag{27}$$

(5) Output weight for O_l can be evaluated from

$$\overline{\beta} = \frac{E \cdot (\overline{H})^T}{\overline{H} \cdot (\overline{H})^T}. \tag{28}$$

(6) After increasing the new hidden node, residual error is calculated:

$$E' = E - \overline{\beta}\overline{H}. \tag{29}$$

The network error rate can be reduced by the output weight O_l. All these steps will iteratively work till the residual error becomes smaller than ε. The training process restarts through the determination of the random input weight ω_l and the bias b_l. Whether the trained network has fulfilled the desirable result or not can be determined from $\{(X', Y')\}$ set.

5. Proposed PCA-Based Incremental ELM (PCA-IELM)

An orthogonal transformation is used to extract meaningful characteristics from data in PCA [33]. PCA may also be used to minimize the dimensions of a large data collection. Principal components from COVID-19 X-ray images are extracted using PCA and given as input to IELM which gradually adds concealed nodes produced at random. A conventional SLFNs function with n hidden nodes can be expressed as

$$f_n(x) = \sum_{i=1}^{n} \beta_i g_i(x), \quad x \in R^{\mathrm{d}}, \beta_i \in R. \tag{30}$$

where $g_i(x) = g(a_i, b_i, x)$ denotes the output of the ith hidden node:$g_i(x) = g(a_i.x + b_i)$ (for additive nodes) or $g_i(x) = g(b_i x - a_i)$.

The ith hidden layer and the output node are linked with output weights β_i. Hidden nodes are randomly added to the existing networks in IELM. The randomly generated hidden node parameters a_i and b_i and fixed output weight are β_i.

Suppose the residual error function for the current network f_n is defined as $e_n \cong f - f_n$. where n is the number of hidden nodes and $f \in L^2(x)$ is the target function. IELM is mathematically represented as

$$f_n(x) = f_{n-1}(x) + \beta_n g_n(x). \tag{31}$$

6. Evaluation Criteria for Effective Measure of Model

For evaluation of the different models, generally, the confusion matrix is prepared. Table 2 defines a simple representation of the confusion matrix [34, 35], and it can classify between predicted and actual values. From the confusion matrix, we can derive different performance metrics, e.g., accuracy, precision, recall, sensitivity, and F-score. To assess the model, nine different metrics are calculated by formula as given in Table 3 [36].

7. Experimental Setup

The whole experiment was performed on a system having a configuration of 10th Generation Intel (R) Core (TM) i7-10750H CPU @ 2.60 GHz processor, 8 GB RAM, and NVIDIA GTX graphics 1650TI. The code is written in *Python* 3.10.0 and uses Jupyter Notebook as a debugger, which can be installed from the link: https://jupyter.org/install.

8. Dataset Description

The COVID-19 chest X-ray images [37] dataset encompasses a total of 13808 images in which 3616 COVID-19 positive cases (26.2%) along with 10,192 (73.8%) normal cases are downloaded from Kaggle. COVID-19 and normal patient chest X-ray images are kept in separate files. Dataset was divided into training and testing images which had been done randomly with a condition that testing images will not be repeated in training images. During the experiment, 80% of the total images were used for training and 20% for testing. All images have the same dimension (299×299) pixels in the PNG file format. Figure 7 demonstrates the X-ray images of normal and COVID-19 cases.

The histogram of an image gives a global description of the image's appearance. It represents the relative frequency of occurrences of various intensity values in an image. In the histogram of the COVID-19 image, the intensity value is highest between bins 14–15, whereas in the normal image the histogram has the highest intensity value at bins 16–17.

```
    Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\mathbf{t_i} \in R^C$.
    Output: IELM model for classification.
(1) Procedure IELM.
(2) Initialize: hidden nodes $l\ = 0$ and maximum L.
(3) For the newly increased hidden layer neuron $O_l$, input weights $\omega_l$ and bias $b_l$ randomly evaluated.
(4) Output of the activation function $g\ (x\prime)$ calculated for the node $O_l$.
(5) Hidden layer output $\overline{H}$ calculated from Eq. (27).
(6) Weights between the hidden and the output layer $\overline{\beta}$ can be evaluated from Eq. (28).
(7) Return $\overline{\beta}$.
(8) End procedure.
```

Algorithm 5: IELM algorithm.

```
 (i) Input: Given $N$ observations along with the class labels $(\mathbf{x_i}, \mathbf{t_i})$, $\mathbf{x_i} \in R^m$, $\in R^C$.
(ii) Output: PCA_IELM model for classification.
 (1) Procedure PCA_IELM.
 (2) Identify the relationship among features through a covariance matrix.
 (3) Through the linear transformation or eigendecomposition of the covariance matrix, we get eigenvectors and eigenvalues.
 (4) Transform our data using eigenvectors into principal components.
 (5) Quantify the importance of these relationships using eigenvalues and keep the important principal components.
 (6) Input for the model is given in terms of extracted principal components.
 (7) Initialize: hidden nodes $l\ = 0$ and maximum L.
 (8) For the newly increased hidden layer neuron $O_l$, input weights $\omega_l$ and bias $b_l$ randomly evaluated.
 (9) g $(x')$ calculated as the output function for the node $O_l$.
(10) Hidden layer output $\overline{H}$ calculated from Eq. (27).
(11) Weights between the hidden and the output layer $\overline{\beta}$ can be evaluated from Eq. (28).
(12) Return $\overline{\beta}$.
(13) End procedure.
```

Algorithm 6: PCA-IELM algorithm.

Table 2: Confusion matrix.

	Predicted		Total
Actual	TP (true positive)	FP (false positive)	TP + FP
	FN (false negative)	TN (true negative)	FN + TN
Total	TP + FN	TN + FP	ALL

Table 3: Performance evaluation measures [36].

SL	Measures	Formula
1.	Accuracy	TP + TN/TP + FP + TN + FN
2.	Specificity ($\mathrm{TN_{rate}}$)	TN/TN + FP
3.	$\mathrm{FN_{rate}}$	FN/TP + FN
4.	Sensitivity ($\mathrm{TP_{rate}}$)/recall	TP/TP + FN
5.	$\mathrm{FP_{rate}}$	FP/TN + FP
6.	Precision	TP/TP + FP
7.	G-mean	$\sqrt{\mathrm{TP_{rate}} \times \mathrm{TN_{rate}}}$
8.	AUC	$1 + \mathrm{TP_{rate}} - \mathrm{FP_{rate}}/2$
9.	F_1-score	$2 * \mathrm{TP}/2 * \mathrm{TP} + \mathrm{FP} + \mathrm{FN}$

This difference in the color intensity value assists in making the distinction between COVID-19 and normal images. Figure 8 demonstrates the histogram plot of normal and COVID-19 images. Figure 9 shows the training images for X-ray images of COVID-19 and normal.

Because PCA uses orthogonal transformation to convert all features into a few independent features, all features are considered during the feature selection process. The data to be processed are reduced to a set of features called a "reduced representation set."

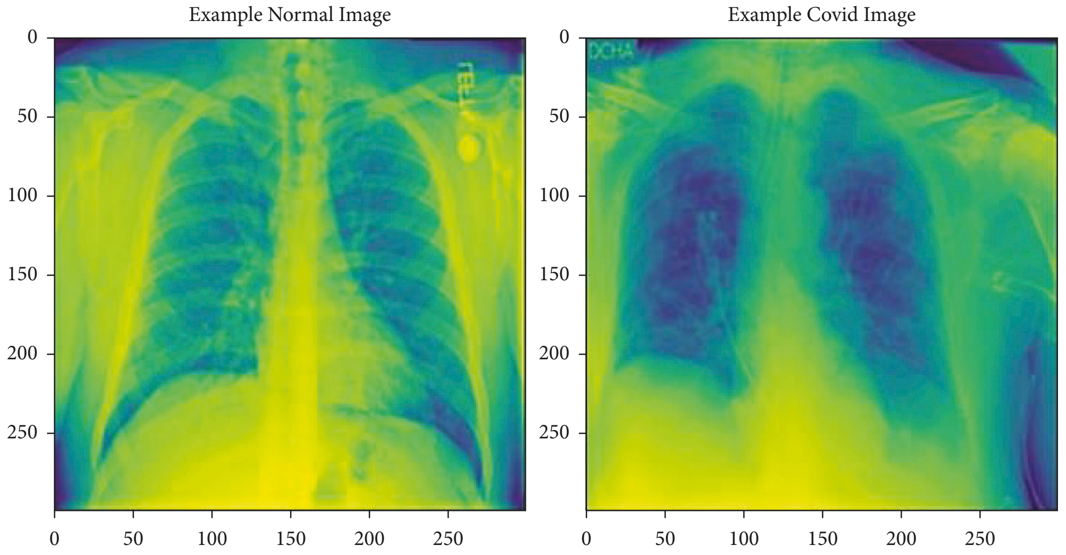


Figure 7: Chest X-ray images of COVID-19 and normal.

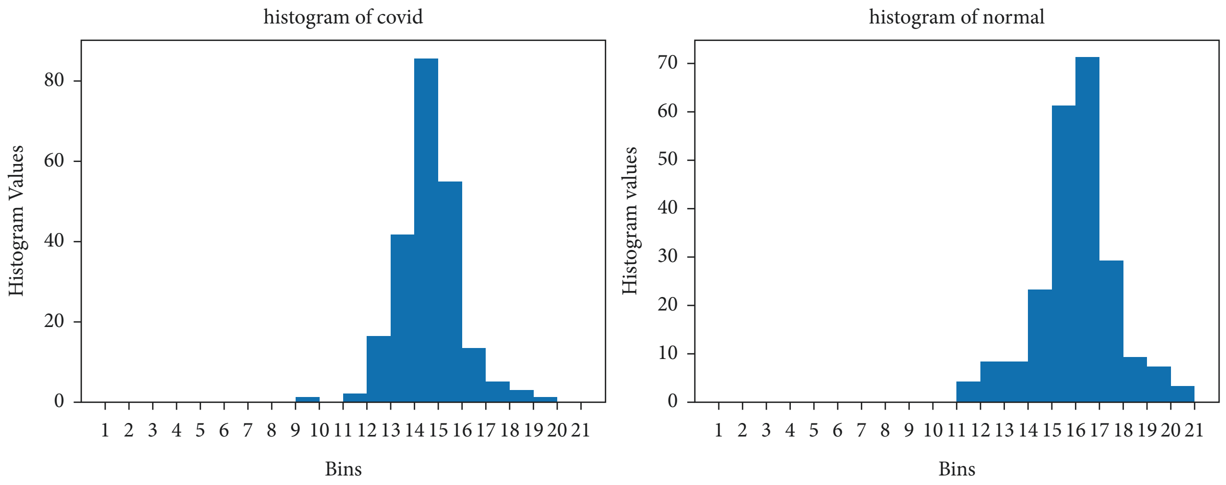


Figure 8: Histogram for X-ray images of COVID-19 and normal.

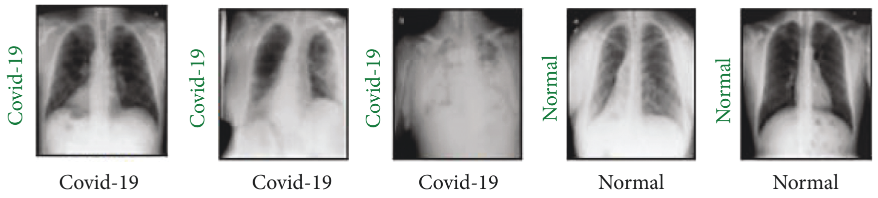


Figure 9: Training images for X-ray images of COVID-19 and normal.

9. Results and Discussion

In this segment, we present the outcomes and analysis of the experiments performed in the COVID-19 patient prediction using the chest X-ray dataset. From the experimental results, the proposed method shows better performance in terms of accuracy, precision, recall, F1-score, AUC, G-mean, and other parameters. For each model, PCA-SVM, PCA-ELM, and PCA-IELM, a separate confusion matrix is formed. All the performance metrics values are derived from the confusion matrix (Tables 4–6). Classification accuracy gained by the proposed method PCA-IELM is 98.11% over the chest X-ray dataset, which suggests better results than the other two models, PCA-based SVM (91.8%) and PCA-based ELM (93.80%) in terms of accuracy. Sometimes, performance metrics' accuracy may be misleading and can misclassify instances. So, other metrics are also taken into consideration to confirm the

Table 4: Confusion matrix for PCA-SVM.

	Predicted		Total
Actual	819	152	971
	187	2985	3172
Total	1006	3137	4143

Table 5: Confusion matrix for PCA-ELM.

	Predicted		Total
Actual	828	110	938
	147	3058	3205
Total	975	3168	4143

Table 6: Confusion matrix for PCA-IELM.

	Predicted		Total
Actual	1192	48	1240
	30	2873	2903
Total	1222	2921	4143

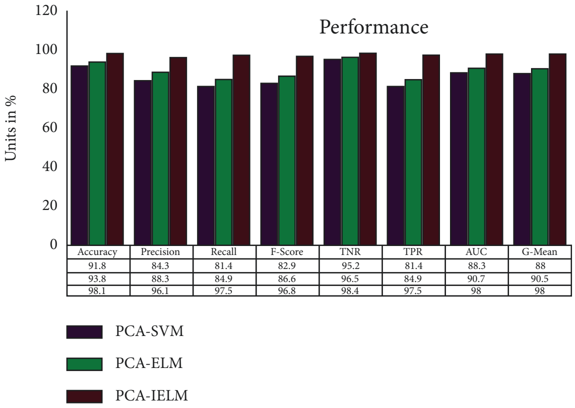


Figure 10: Performance comparison of different classifiers.

claim made by the classifier. PCA-IELM has the highest precision value of 96.11%. That means PCA-IELM is 96.11% reliable in making decisions, whereas models PCA-SVM and PCA-ELM record less precision, 84.3% and 88.3%, respectively. Similarly, for the proposed method PCA-IELM, other metrics (refer to Figure 10) recall, F_1-score, TPR, TNR, and G-mean are considerably higher than the other two methods, PCA-SVM and PCA-ELM.

The geometric mean (G-mean) is a statistic that analyzes categorization performance across majority and minority classes. Even if negative examples are correctly labelled as such, a poor G-mean suggests weak performance in identifying positive occurrences. This statistic is essential for preventing overfitting the negative class while underfitting the positive class, since the COVID-19 dataset understudy is also class imbalanced (IR = 2.81). Even then, the PCA-ELM model indicates good performance by attaining the highest G-mean value of 98%. Similarly, PCA-SVM and PCA-ELM have 88% and 90.5% success rates, respectively.

Table 7 demonstrates the performance variation (sensitivity, specificity, precision, F1-score, accuracy) based on different counts of hidden nodes in the range of 10–150 with an interval of 10 hidden nodes. Training and testing accuracies of PCA-IELM demonstrated almost the same behavior on the COVID-19 dataset (refer to Figure 11). There is moderate variation in the accuracy of PCA-IELM with respect to different numbers of hidden nodes. The accuracy at 10 numbers of hidden nodes was found to be 97.73%, and 98.11% was achieved at 140 numbers of hidden nodes in the PCA-IELM model and beyond (refer to Table 7).

When there is a moderate to large class imbalance, precision-recall curves should be drawn. Here, the COVID-19 dataset is imbalanced with an imbalance ratio (IR) of 2.81. It is worth noticing that precision is also called the positive

Table 7: Performance variation based on different hidden nodes.

Number of hidden nodes	Performance metrics (%)				
	Sensitivity	Specificity	Precision	F_1-score	Accuracy
10	94.13	96.23	93.74	95.19	97.73
20	94.16	96.19	93.36	95.19	97.73
30	94.24	96.35	94.48	95.24	97.74
40	93.98	96.07	93.01	94.98	97.70
50	94.11	96.16	93.09	95.11	97.71
60	94.18	96.28	93.45	95.18	97.73
70	94.73	96.82	94.69	95.64	97.75
80	94.84	96.79	94.86	95.29	97.75
90	94.25	96.77	94.92	95.36	97.75
100	94.03	96.11	93.11	95.02	97.71
110	94.17	96.24	93.22	95.16	97.73
120	94.48	96.46	94.59	95.71	97.75
130	94.10	96.15	93.19	95.11	97.71
140	97.62	98.12	96.33	96.50	98.11
150	97.54	98.35	96.12	96.83	98.11

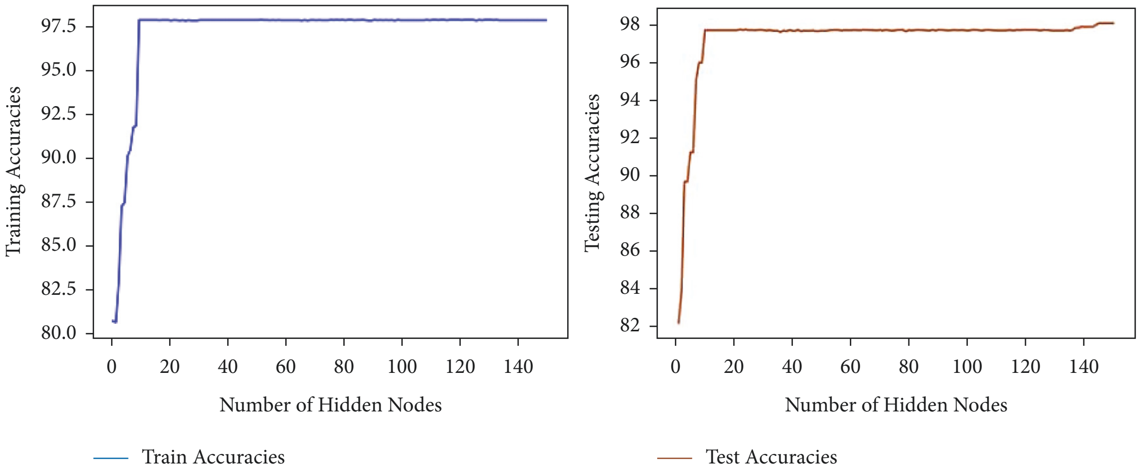
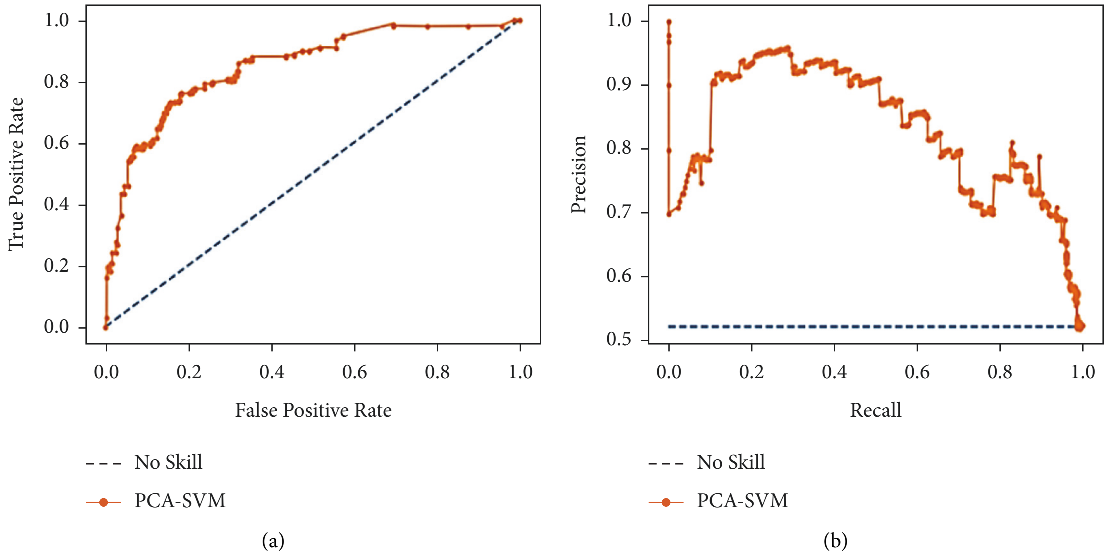


Figure 11: Accuracy variation with number of hidden nodes for PCA-IELM.



(a)

(b)

Figure 12: (a) Analysis of ROC curve and (b) analysis of precision-recall for PCA-SVM.

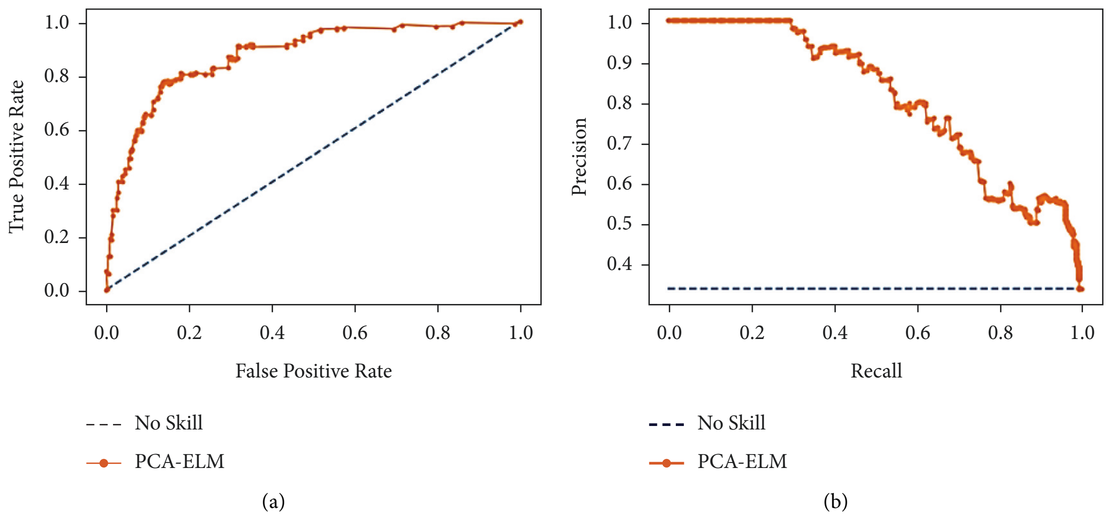


Figure 13: (a) Analysis of ROC curve and (b) analysis of precision-recall for PCA-ELM.

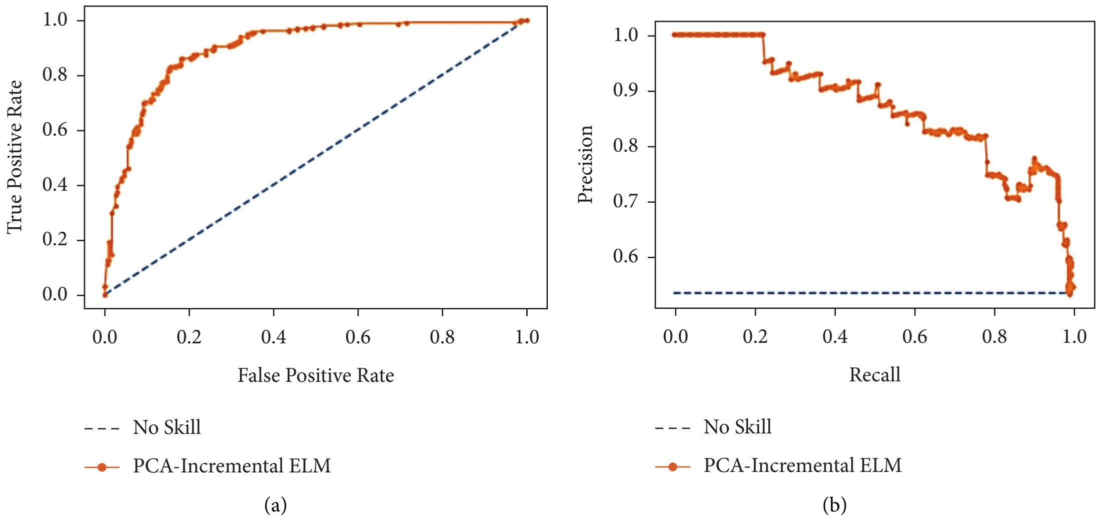


Figure 14: (a) Analysis of ROC curve and (b) analysis of precision-recall for PCA-incremental ELM.

Table 8: Proposed method and other related models' comparative analysis [38].

S.No.	Study	Method used	Number of cases	Type of images	Accuracy (%)
1	Ioannis et al. [39]	VGG-19	700 pneumonia, 504 healthy, 224 COVID-19 (positive)	Chest X-ray	93.48
2	Gunraj, Wang, and Wong [40]	COVID-Net	5526 COVID-19 (negative), 8066 healthy, 53 COVID-19 (positive)	Chest X-ray	92.4
3	Sethy et al. [41]	ResNet50þ SVM	25 COVID-19 (negative), 25 COVID-19 (positive)	Chest X-ray	95.38
4	Hemdan et al. [42]	COVIDX-Net	25 normal, 25 COVID-19 (positive)	Chest X-ray	90.0
5	Narin et al. [43]	Deep CNN ResNet-50	50 COVID-19 (negative), 50 COVID-19 (positive)	Chest X-ray	98
6	Ying et al. [44]	DRE-Net	708 healthy, 777 COVID-19 (positive)	Chest CT	86
7	Wang et al. [45]	M-Inception	258 COVID-19 (negative), 195 COVID-19 (positive)	Chest CT	82.9
8	Zheng et al. [46]	UNetþ3D deep network	229 COVID-19 (negative), 313 COVID-19 (positive)	Chest CT	90.8
9	Xu et al. [47]	ResNetþ location attention	175 healthy, 224 viral pneumonia, 219 COVID-19 (positive)	Chest CT	86.7
10	Tulin et al. [38]	DarkCovidNet	500 Pneumonia, 500 no-findings, 125 COVID-19 (positive	Chest X-ray	98.08
11	Proposed model	PCA-IELM	10,192 normal, 3616 COVID-19 (positive)	Chest X-ray	98.11

Table 9: Time elapsed during training and testing of models.

Dataset	Algorithm	Training time (s)	Testing time (s)
COVID-19 chest X-ray	PCA-SVM	0.027	0.022
	PCA-ELM	0.053	0.049
	PCA-IELM	12.353	9.525

predictive value (PPV). Moreover, recall is also known as sensitivity, hit rate, or true-positive rate (TPR). It means they talk about positive cases and not negative ones. Most machine learning algorithms often involve a trade-off between recall and precision. A good PR curve has a greater AUC (area under curve). Figures 12(b), 13(b), and 14(b) depict PR curves. Figure 13(b) shows the greater AUC, which is an indication of the better performance of PCA-IELM than the other two models. In addition to these, ROC of Figure 14(a) also grabs more AUC than two other Figures 12(a) and 13(a). Therefore, PCA-IELM claims better performance than PCA-SVM and PCA-IELM. The proposed PCA-IELM model outperforms other previously developed models for identification of COVID-19 patients from chest X-ray image (refer Table 8 [38–47]). As far as the training and testing time taken by the proposed model PCA-IELM is concerned, it was higher (refer to Table 9) because the execution of the model happened in an incremental way and not in one go.

10. Conclusions

In this paper, an effective classification model is proposed on the COVID-19 chest X-ray image dataset using principal component analysis (PCA) and incremental extreme learning machine (IELM). This study established the valuable application of the ELM model to classify COVID-19 patients from X-ray images by developing the PCA-IELM model. The proposed PCA-based IELM algorithm is an efficient IELM-based algorithm. The hidden node parameters are measured by the information returned to the PCA in the training dataset, and using the Moore–Penrose generalized inverse output, the node parameters are determined. PCA-IELM utilizes the best feature of IELM, which is to increase hidden nodes incrementally and wisely determine the output weights, whereas ELM requires you to set the appropriate number of hidden nodes manually, and this is similar to the hit and trial method. In comparison with the ELM, the output error of the IELM rapidly reduces and is near to zero as the number of hidden neurons increases. It was observed that as the number of hidden nodes increased, the performance of the PCA-IELM increased and it became stable at 150 hidden nodes. PCA-IELM outperforms PCA-SVM and PCA-ELM in terms of accuracy (98.11%), precision (96.11%), recall (97.50%), F1-score (98.50%), G-mean (98%), etc. The suggested research contributes to the prospect of a low-cost, quick, and automated diagnosis of the COVID-19 patient, and it may be used in clinical scenarios. This effective system can provide early detection of COVID-19 patients. As a result, it is helpful in controlling the further spread of the virus from an affected person. This is an intelligent assistance for radiologists to accurately diagnose COVID-19 in X-ray images.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] F. Cao, B. Liu, and D. Sun Park, “Image classification based on effective extreme learning machine,” *Neurocomputing*, vol. 102, pp. 90–97, 2013.

[2] L. Ying, “Orthogonal incremental extreme learning machine for regression and multiclass classification,” *Neural Computing & Applications*, vol. 27, no. 1, pp. 111–120, 2016.

[3] Y. Sun, L Li, L. Zheng et al., “Image classification base on PCA of multi-view deep representation,” *Journal of Visual Communication and Image Representation*, vol. 62, pp. 253–258, 2019.

[4] V. Kumar, “Evaluation of computationally intelligent techniques for breast cancer diagnosis,” *Neural Computing & Applications*, vol. 33, no. 8, pp. 3195–3208, 2021.

[5] M. Mustaqeem and M. Saqib, “Principal component-based support vector machine (PC-SVM): a hybrid technique for software defect detection,” *Cluster Computing*, vol. 24, no. 3, pp. 2581–2595, 2021.

[6] A. Castaño, F. Fernández-Navarro, and C. Hervás-Martínez, “Pca-Elm PCA-ELM: a robust and pruned extreme learning machine approach based on principal component analysis,” *Neural Processing Letters*, vol. 37, no. 3, pp. 377–392, 2013.

[7] M. Mateen, J. Wen, N. Nasrullah, S. Song, and Z. Huang, “Fundus image classification using VGG-19 architecture with PCA and SVD,” *Symmetry*, vol. 11, no. 1, 2018.

[8] Z. Zhao, Z. Chen, Y. Chen, S. Wang, and H. Wang, “A class incremental extreme learning machine for activity recognition,” *Cognitive Computation*, vol. 6, no. 3, pp. 423–431, 2014.

[9] G. B. Huang and L. Chen, “Convex incremental extreme learning machine,” *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.

[10] M. Zhu, Y. Song, J. Guo et al., “PCA and Kernel-based extreme learning machine for side-scan sonar image classification,” *2017 IEEE Underwater Technology (UT)*, vol. 238, p. 2017, 2017.

[11] X. Kang, S. Member, X. Xiang, J. A. Benediktsson, S. Li, and S. Member, “PCA-based edge-preserving features for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7140–7151, 2017.

[12] C. Perales-González, F. Fernández-Navarro, J. Pérez-Rodríguez, and M. Carbonero-Ruz, “Negative correlation hidden layer for the extreme learning machine,” *Applied Soft Computing*, vol. 109, Article ID 107482, 2021.

[13] R. Li, H. Zhang, S. Gao, Z. Wu, and C. Guo, “An improved extreme learning machine algorithm for transient

electromagnetic nonlinear inversion," *Computers & Geosciences*, vol. 156, Article ID 104877, 2021.

[14] J. Zhang, S. Ding, N. Zhang, and Z. Shi, "Incremental extreme learning machine based on deep feature embedded," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 1, pp. 111–120, 2016.

[15] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[16] B. S. Raghuwanshi and S. Shukla, "Classifying imbalanced data using SMOTE based class-specific kernelized ELM," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 5, pp. 1255–1280, 2021.

[17] G. B. Huang, M. B. Li, L. Chen, and C. K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, no. 4–6, pp. 576–583, 2008.

[18] S. Xu and J. Wang, "A fast incremental extreme learning machine algorithm for data streams classification," *Expert Systems with Applications*, vol. 65, pp. 332–344, 2016.

[19] J. Gao, S. Chai, B. Zhang, and Y. Xia, "Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis," *Energies*, vol. 12, no. 7, p. 1223, 2019.

[20] S. M. Basha and D. S. Rajput, "A supervised aspect level sentiment model to predict overall sentiment on tweeter documents," *International Journal of Metadata Semantics and Ontologies*, vol. 13, no. 1, pp. 33–41, 2018.

[21] S. M. Basha and D. S. Rajput, "A roadmap towards implementing parallel aspect level sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 20, pp. 29463–29492, 2019.

[22] A. K. Dubey and K. K. Mohbey, "Enabling CT-Scans for covid detection using transfer learning-based neural networks," *Journal of Biomolecular Structure and Dynamics*, pp. 1–12, 2022.

[23] S. M. Basha and D. S. Rajput, "Survey on evaluating the performance of machine learning algorithms past contributions and future roadmap," *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, Academic Press, Cambridge, Ma, pp. 153–164, 2019.

[24] Y. Z. Miao, X. P. Ma, and S. P. Bu, "Research on the learning method based on PCA-ELM," *Intelligent Automation & Soft Computing*, vol. 23, no. 4, pp. 637–642, 2017.

[25] R. Sakthivel, I. S. Thaseen, M. Vanitha et al., "An efficient hardware architecture based on an ensemble of deep learning models for COVID-19 prediction," *Sustainable Cities and Society*, vol. 80, Article ID 103713, 2022.

[26] S. Zhang and W. Tan, "An extreme learning machine based on the mixed kernel function of triangular kernel and generalized hermite dirichlet kernel," *Discrete Dynamics in Nature and Society*, vol. 2016, pp. 1–11, 2016.

[27] B. S. Raghuwanshi and S. Shukla, "UnderBagging based reduced Kernelized weighted extreme learning machine for class imbalance learning," *Engineering Applications of Artificial Intelligence*, vol. 74, pp. 252–270, 2018.

[28] B. S. Raghuwanshi and S. Shukla, "Classifying imbalanced data using ensemble of reduced kernelized weighted extreme learning machine," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 11, pp. 3071–3097, 2019.

[29] K. Vishnu Vandana and D. S. Rajput, "A Review on the significance of machine leaning for data analysis in big data," *Jordanian Journal of Computers and Information Technology*, vol. 06, no. 01, pp. 41–57, 2020.

[30] J. Qin, T. F. Burks, M. S. Kim, K. Chao, and M. A. Ritenour, "Citrus canker detection using hyperspectral reflectance imaging and PCA-based image classification method," *Sensing and Instrumentation for Food Quality and Safety*, vol. 2, no. 3, pp. 168–177, 2008.

[31] K. Cheng, S. Gao, W. Dong, X. Yang, Q. Wang, and H. Yu, "Boosting label weighted extreme learning machine for classifying multi-label imbalanced data," *Neurocomputing*, vol. 403, pp. 360–370, 2020.

[32] S. Song, M. Wang, and Y. Lin, "An improved algorithm for incremental extreme learning machine," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 308–317, 2020.

[33] G. T. Reddy, M. P. K Reddy, K. Lakshmanna et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, Article ID 54776, 2020.

[34] P. Kumar and R. Singh Thakur, "An approach using fuzzy sets and boosting techniques to predict liver disease," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 3513–3529, 2021.

[35] V. Kumar and G. S. Lalotra, "predictive model based on supervised machine learning for heart disease diagnosis," in *Proceedings of the 2021 IEEE International Conference on Technology, Research, and Innovation for Betterment of Society (TRIBES)*, pp. 1–6, IEEE, Raipur, India, 17 December 2021.

[36] P. Kumar and R. S. Thakur, "Liver disorder detection using variable- neighbor weighted fuzzy K nearest neighbor approach," *Multimedia Tools and Applications*, vol. 80, no. 11, pp. 16515–16535, 2021.

[37] D. P Tuan, "COVID-19 Radiography Database," *COVID-19 Chest X-ray Database*, vol. 9, no. 1, 2021, https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

[38] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. Rajendra Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Computers in Biology and Medicine*, vol. 121, Article ID 103792, 2020.

[39] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Phys. Eng. Sci. Med.*vol. 43, no. 2, pp. 635–640, 2020.

[40] H. Gunraj, L. Wang, and A. Wong, "COVIDNet-CT: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest ct images," *Frontiers of Medicine*, vol. 7, pp. 1–12, 2020.

[41] P. K. Sethy, S. K. Behera, P. K. Ratha, and P. Biswas, "Detection of coronavirus disease (COVID-19) based on deep features and support vector machine," *Int. J. Math. Eng. Manag. Sci*, vol. 5, no. 4, pp. 643–651, 2020.

[42] E. E.-D. Hemdan, M. A. Shouman, and M. E. Karar, "COVIDX-net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images," *Biomedical Image Analysis*, 2020, http://arxiv.org/abs/2003.11055.

[43] T. Gadekallu Reddy, P Quoc-Viet, T Thien Huynh, B Sweta, M Praveen Kumar Reddy, and L Madhusanka, "Federated learning for big data: a survey on opportunities, applications, and future directions," *Mobile Edge Computing and Wireless AI in 5G and BeyondarXiv preprint*, 2021.

[44] S. Ying, S. Zheng, L. Li et al., "Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, no. –1, p. 1, 2020.

[45] S. Wang, B. Kang, J. Ma et al., “A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19),” *European Radiology*, vol. 31, no. 8, pp. 6096–6104, 2021.

[46] S. Bhattacharya, P. K. Reddy Maddikunta, Q. V. Pham et al., “Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey,” *Sustainable Cities and Society*, vol. 65, Article ID 102589, 2021.

[47] X. Xu, X. Jiang, C. Ma et al., “A deep learning system to screen novel coronavirus disease 2019 pneumonia,” *Engineering*, vol. 6, no. 10, pp. 1122–1129, 2020.