JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 1510 - 1522
https://publishoa.com
ISSN: 1309-3452

Prediction of COVID-19 for diabetes patients using pre-trained convoluted Recurrent Neural Network and fast 3D-convolution neural network with U-net++

K. Manohari
Ph.D Research Scholar
Department of Computer Science
Affiliated to Thiruvalluvar University
Theivanai Ammal College For Women, Autonomous, Villupuram, Tamil Nadu, India
hmano730@gmail.com

Dr. S. Manimekalai
HOD, Department of computer science
Affiliated to Thiruvalluvar University
Theivanai Ammal College For Women, Autonomous,Villupuram, Tamil Nadu, India
mamekaroshni@gmail.com

ABSTRACT

This paper aims to predict COVID- 19 to analyse the death rate due to the presence of diabetes. At present, the entire world witnessed the rapid and severe spread of theNovel Coronavirus (COVID-19) since 2019. This spread throughout mankind and animals. Its severe complication and without the earlier disease detection and treatment leads to death. So CAD-based system is obtained for rapid detection of disease and predict the death rate due to diabetes which involves machine learning techniques. First to collect the real-time public dataset of COVID-19 patients with diabetes topredict thedeath rate due to thepresence of diabetes. Here the data has been collected through HDFS. Initially, the diabetes data and chest X-ray for COVID patients has been collected and classified for obtaining the abnormal range of diabetes in corona patients. Here for classification, we use apre-trained convoluted recurrent neural network (PCRNN) and fast 3D-convolution neural network with U-net++ (F-3D-CNN-U net++) which use for classifying the numerical data and image data. The numerical data has been classified using pre-trained convoluted recurrent neural network and image data has been classified using afast 3D-convolution neural network with U-net++. By diagnosing the diabetes range and coronavirus affected part of thelungs, the death rate has been calculated. The experimental results show accuracy, precision, recall, F-1 score, true positive, false positive rate, MSE, MLR.

Keywords: COVID- 19, diabetes, HDFS, chest X-ray, PCRNN, 3D-CNN-U net++

1. INTRODUCTION

The Coronavirus (COVID-19) first appeared in China in December of this year. Over 95 million cases had been documented over the world as of January 2021, with a fatality rate of 2% of all closed cases [1]. This rapid pandemic spread is a worldwide issue and a severe threat to public health and the global economy. Most of the countries restricted social interaction as a preventive measure, such as isolation and quarantine, to prevent the sickness from spreading. However, due to late identification and the virus's unusual and unknown nature, many infected people could not benefit from adequate therapy. Many researchers have recently concentrated on establishing novel approaches to screen infected individuals at various stages to discover significant connections between apatient's clinical traits and thelikelihood of succumbing to disease. Diabetes' global incidence and prevalence are constantly increasing, putting a significant strain on healthcare systems. The prevalence of all forms of diabetes in adults is expected to rise by 69 percent in developing countries and 20 percent in affluent nations between 2010 and 2030 [2]. Diabetes is expected to affect 55 million people in the United States by 2030, 62 million in China, and 87 million in India. Finally, In 2015 it was projected that the global cost of diabetes was $1.31 trillion US dollars. Serious efforts, as well as investments in type 2 diabetes prevention, are critical, and prevention programmes are effective not only in clinical trials but also in practical, real-world situations. These impediments, many of which are also known as socioeconomic determinants of health, contribute to "cascades of care," in which substantial portions of thepopulation fail to reach prevention goals [3]. ML is a subfield of AI that aims to give computers a "learning capacity" through theuse of well-defined algorithms to enhance performance or make accurate predictions. These methods often learn on previously accessible data, which is presented in form of labelled training sets. These labelled training sets are used by supervised learning algorithms to optimise the parameters of a statistical model to minimise a loss function [4]. The trained model can then generate accurate predictions utilising data that was never used during the training phase as input. Naturally, the quality and scale of the datasets employed are critical in assuring the algorithm's proper performance. During the current epidemic, ML are utilised to build various methods aimed at identifying patients who are likely to become infected at an early stage. These methods produce predictions based on basic patient information, clinical symptoms, travel history, and the time spent in the hospital [5].

The contribution of this paper is as follows:

To collect a dataset of COVID- 19 patients and process them for both numerical and image data

Pre-processing and segmentation have been carried out for both numerical and image data of diabetes patients with COVID-19. Data has been classified based on numerical and image data, the numerical data has been classified using PCRNN and image data classification is done using F-3D-CNN-U net++. The experimental results show in terms of accuracy, precision, recall, F-1 score, true positive, false positive rate, MSE, MLR for theproposed technique with various COVID-19 datasets.

2. RELATED WORKS

Early detection and diagnosis using AI algorithms using a variety of data such as CT scans, clinical data, X-rays, and blood sample data help to avoid spread and combat theCOVID-19 pandemic. The authors of [6] reported a study that used supervised machine learning techniques such LR, LDA, Naive Bayes, k-NN, DT, XGBOOST and SVM to identify patients who could develop severe COVID19 symptoms early. Individual fundamental data such as gender and age range, comorbidities, symptoms and recent travel history were used to train the ML methods, which were taught using a publically available database relevant to Brazil. The authors claim that a ROC of 0.92, a sensitivity of 0.88, and a specificity of 0.82 can predict illness outcomes. Using a database from a tertiary medical centre, [7] focuses on identifying patients at risk of deterioration throughout their hospital stay. The results reveal a sensitivity of 88.0 percent, a specificity of 92.7 percent, and an accuracy of 92.0 percent. In [8], the author proposes a decision strategy based on asupervised XGBoost classifier to forecast high-risk patients. Lymphocytes, LDHand high-sensitivity C-reactive protein were initially used to train the predictive model. Results reveal that model can correctly predict patient outcomes with greater than 90% accuracy. Other attempts are aimed at identifying patients who require specialist treatment, such as hospitalisation and/or specialised care units [9-11], as well as patients who are at a higher risk of death [12]. The use of supervised ML approaches to anticipate the COVID-19 epidemic was examined in [13]. SVM was used to forecast over a dataset received from the WHO containing 303 patients in [14]. During the testing phase, the proposed system has a precision of 0.967. Similarly, utilising numerous machine learning methods including LASSO, SVM, RF and KNN, constructed a model to predict the death of COVID-19 patients in [15]. Models were taught to distinguish between three scenarios: death and survival, death and survival and death and survival. within 14 to 30 days of receiving an initial diagnosis. The researchers discovered that age [16], diabetes mellitus and cancer were all important factors in COVID-19 patients' mortality predictions. Many researchers have been using DL for COVID-19 patients in recent years, however, there is still a scarcity of data on the disease. The application of DL for COVID-19 diagnosis from medical imaging data appears to be the most popular among them, however, a few algorithms still lack transparency and interpretability. This signifies that the image characteristic responsible for creating output is still unknown. As a result, it is required to describe elements of a medical image with the performance of established structures that are responsible for distinguishing COVID-19 instances from others, and this would be beneficial to doctors in learning more about the virus [17]. COVID-19 case reporting data varies from nation to country, potentially misrepresenting the true transmission rate and complicating illness tracking. Some systems require longer length peptides against coronavirus protease for virtual screening, as well as the development of a scoring algorithm to redesign COVID-19 antibodies. When more data becomes available, however, the number of deep learning applications will skyrocket [18].

3. SYSTEM MODEL

This section discusses the proposed design in theprediction of COVID-19 for diabetes patients using novel deep learning architectures. The overall proposed architecture is given in figure-1.

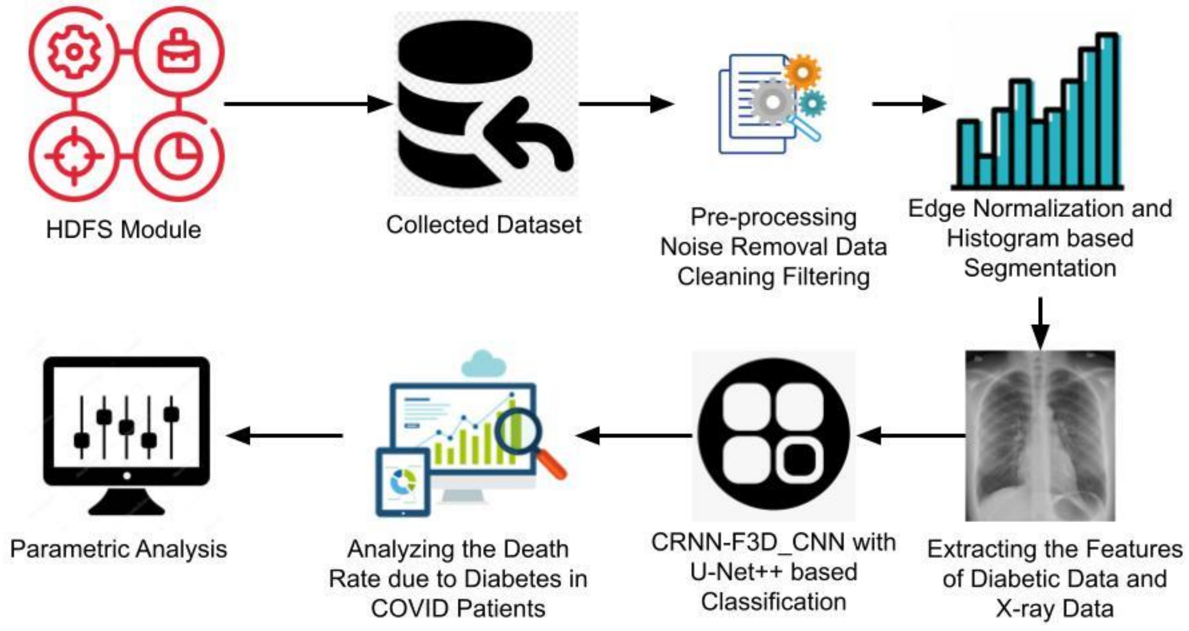

HDFS Module
Collected Dataset
Pre-processing Noise Removal Data Cleaning Filtering
Edge Normalization and Histogram based Segmentation
Extracting the Features of Diabetic Data and X-ray Data
CRNN-F3D_CNN with U-Net++ based Classification
Analyzing the Death Rate due to Diabetes in COVID Patients
Parametric Analysis

Figure-1 Overall Proposed architecture

3.1 Pre-processing and segmentation of input dataset:

The text and image have to be preprocessed initially for cleaning the data to attain better features. This will be achieved by pre-processing steps on the training data. It has to filter and remove the noise of data using amedian filter. Any text or image that isn't related to the context of data or end-output can be designated as noise. This stage involves removing all forms of distracting things from the text. Preparing a dictionary of noisy entities and iterating text objects by tokens, deleting those tokens that are involved in the noise dictionary, is a general strategy for noise reduction.

The primary goal of the first stage is to eliminate noise from the image. The greyscale conversion of the input leaf image will be performed here. The filtering method will be used to eliminate the noise from the image. The image is filtered in this stage of pre-processing to reduce noise. Convoluted Gaussian filtering is used in the suggested model.

Gaussian filters are a type of convolutional filter where the weights are selected based on the shape of a Gaussian function. For reducing noise from a normal distribution, the Gaussian smoothing filter is an excellent choice. In one dimension, the zero-mean Gaussian function is given by equation (1)

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \qquad (1)$$

In 3-D an isotropic Gaussian has aform (2):

$$G(x,y,z) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} \qquad (2)$$

Convolution is used in the Gaussian filtering to employ this distribution as a 'point-spread' function.A convolution kernel with integer values approximates a Gaussian with a value of 1.0. It's not clear how to choose the mask's values to resemble a Gaussian.Gaussian value has been integrated over the entire pixel.Integrals are not integers, so the array was rescaled to give corners a value of 1. Finally, the 273 represents the sum of all of the mask's values.

It was added noise with ImageJ's Add noise built-in function and then filtered with the Gaussian Blur built-in function. The "Gaussian Blur" is achieved via convolution of a kernel with the picture pixels, which is characterised by a Gaussian function. In the discrete example, convolution is given by Eqn (3)

$$\mathrm{f} * \mathrm{g}[\mathrm{n}] \overset{\mathrm{def}}{=} \sum_{m=-\infty}^{\infty} \mathrm{f}[\mathrm{m}] \cdot \mathrm{g}[\mathrm{n}-\mathrm{m}] \; (3)$$

The two-dimensional Gaussian function was utilised to produce the kernel. The amplitude is A, the centre is (xo,yo), and the standard deviations in the x and y directions are x, y by Eqn (4).

$$f(x,y) = A \cdot e^{-\left(\frac{(x-x_0)^2}{2\sigma x^2}+\frac{(x-y_0)^2}{2\sigma y^2}\right)} \qquad (4)$$

The challenge of splitting a string of written language into its component text is known as numerical data segmentation (also known as sentence segmentation). It can divide phrases apart when it come across a punctuation mark in English or several other languages. Tables including abbreviations with periods can help to prevent incorrect text boundary assignment while processing plain text. N-grams segmentation was used to segment the text in this scenario.

N-grams are made up of numerous words that are used together. Unigrams are N-grams with the number N equal to one. Bigrams (N=2), trigrams (N=3), and so on can all be utilised in the same way. When compared to bigrams and trigrams, unigrams usually carry less information. More contexts you have to work with, thelonger the n-gram. The ideal length is determined by the application; nevertheless, if your n-grams are too short, you may miss critical differences. On the other hand, if they're too long, you can miss out on capturing "universal knowledge" and instead focus on specific examples.

Curves enclosing objects which have to be identified are inserted to define the actual level contours. Gradient operator against discrete image x is given by first-order finite differences represented as Eqn (5).

$$(\nabla \mathrm{x})_i = \left(x_{i_1,1_2} - x_{i_1-1,l_2}, x_{i_1,l_2} - x_{i_1,l_2-1}\right) \qquad (5)$$

With first-order finite differences. divergence operator is described for continuous vector field v is given as Eqn (6)

$$div = -\nabla(6)$$

whose discrete form is given in Eqn (7),

$$\mathrm{div}\ (v)_i = v^1 i_1 + 1, i_2 - v^I_{i_1,i_2} + v^2 i_{1,i_2+1} - v^2_{i_1,i_2} \qquad (7)$$

When the cost function is minimized, anoptimized image is obtained in Eqn (8)

$$\underset{\mathrm{x \in R^A}}{min} \mathrm{f(x)} = \frac{1}{2} \parallel \mathrm{y} - \mathrm{x} \parallel^2 + \lambda \mathrm{J}_\in(\mathrm{x}) \qquad (8)$$

In which,$\mathrm{J}_\in(\mathrm{x})$ is $\sum_\mathrm{i} \parallel (\mathrm{Gx})_\mathrm{i} \parallel_i$is the complete smoothed variation and$(\mathrm{Gx})_\mathrm{i}$is an approximated image gradient over pixel is $\lambda > 0$ is a Lagrange multiplier parameter where data fit$\parallel \mathrm{y} - \mathrm{x} \parallel^2$ is controlled by weights and $\mathrm{J}_\in(\mathrm{x})$is a term specifying regularization.L^2 norm inR^2 si smoothed and used to compute u with $1 > \in > 0$ by Eqn (9),

$$\parallel \mathrm{u} \parallel_\in = \sqrt{\epsilon^2 + \parallel \mathbf{u} \parallel^2} \qquad (9)$$

$\in$represent the small regularization parameter and $u = \nabla \mathrm{f(x)}$. The gradient of any function f is defined as Eqn (10),

$$\nabla \mathrm{f(x)} = \mathrm{x} - \mathrm{y} + \lambda \nabla \mathrm{J}_\in(\mathrm{x}) \qquad (10)$$

The total variation gradient norm after smoothing is given as Eqn (11)

$$\nabla \mathrm{J}_\in(\mathrm{x})_\mathrm{i} = \mathrm{G}^*(\mathrm{u}), \text{ in which } \parallel \mathrm{u} \parallel_\mathrm{i} = \frac{(\mathrm{Gx})_\mathrm{i}}{\parallel (\mathrm{Gx})_\mathrm{i} \parallel_\in} \qquad (11)$$

While segmenting images using the proposed model, once the boundary is limited, asaliency map was applied to the input fundus.

3.2 Pre-trained convoluted recurrent neural network (PCRNN) based Text classification:

In this section, details of theframework model, which comprises convolutional and RNN, are described. In the model's design feeds word embeddings to a CNN that learns to extract high-level features as inputs. The CNN outputs are then fed into an RNN language model with LSTM, which is followed by a classifier layer.

A. Embedded Layer

The network's first layer converts words into real-valued feature vectors that contain semantic and grammatical data. The input to the model is a string of words $\left[w_i, \dots, w_{\}}\right]$, each of which is drawn from vocabulary V. This is created by simply concatenating all of the words in V's embeddings.

B. Convolutional Layer:

Figure 2 shows a somewhat different model design than the CNN architecture.

k-dimensional word vector comparable to ith word in a sentence of length n will be $x_i = R^k$which is represented in eq (12):

$$x_{l:n} = x_1 \oplus x_2 \oplus \dots \oplus x_n \qquad (12)$$

A concatenation operator $\oplus$ is used here. Let $x_{i:i+j}$ stand for concatenation of the terms xi, $x_i, x_{i+1} \dots x_{i+j}$. A filter $w \in R^{\mathrm{hh}}$ is used in a convolutional process. For example, a feature is created from a word window $|x_{il+h-1}|$by Eqn (13):

$$c_i = f(W \cdot X_{i:i+h-1} + b) \qquad (13)$$

The bias term b ∈ R is used here, and f is a non-linear function like hyperbolic tangentis given by Eqn (14).

$$c = [c_1, c_2 \dots \dots c_{n-h+1}] \qquad (14)$$

$c \in R^{n-h+1}$ feature maps were then put into an LSTM recurrent layer to capture long-term dependencies. The number of parameters in themodel will be reduced using this method.

C. Recurrent Layer

RNN is a form of NN architecture that is utilized to model sequences. A recurrent layer uses the recursive process to take the input vector x iRnand hidden state at each time step t is given by Eqn (15).

$$h_t = f(W_{st} + U h_{t-1} + b) \qquad (15)$$

Where f is an element-wise nonlinearity and $W \in R^{m+n}, b \in R^{m+m}, b \in R^m$ are parameters. Because of the disappearing and inflating gradient, LSTM dependencies with a vanilla RNN is tough. LSTM addresses shortcomings of an RNN by adding a memory cell that receives as input x_t, h_{t-1}, c_{t-1} and outputs $h_t, c_t$by the followingEqn (16):

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i)$$
$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f)$$
$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o)$$
$$g_t = \sigma(W^g x_t + U^g h_{t-1} + b^z) \qquad (16)$$

The element-wise sigmoid, as well as hyperbolic tangent functions, are it, ft, and ot, respectively, and the input, forget, and output gates are it, ft, and ot. At time t = 1, the vectors h0 and c0 are set to zero. The element-by-element multiplication operator is 2. In terms of time, the LSTM parameters are time-preserving. On many tasks, including language modelling, LSTM beats vanilla RNNs.

D. Back Propagation Through Time

Errors can even be spread backwards. BPTT is a basic RNN variant of thebackpropagation technique; with BPTT, the mistake is transmitted back in time for certain time steps via recurrent connections. As a result, when the network is taught by BPTT, it absorbs as well as remembers data for several time steps in the buried layer.

E. Classification Layer

Before computing the prediction probabilities for all of the categories using a softmax activation function. This is done byEqn (17):

$$p(y = k \mid X) = \frac{\exp\left(w_k^T x + b_k\right)}{\sum_{k=1}^{k} \exp\left(w_k^T x + b_k\right)} \tag{17}$$

Where $w_k and\ b_k$are weight and bias vectors.

3.3 Fast 3D-convolution neural network with U-net++ (F-3D-CNN-U net++) based image classification:

As a linear classifier, an F- 3D CNN is used. It uses weighted softmax cross-entropy loss and Adam Optimizer. For therelatively tiny Kaggle dataset, the network has been reduced to avoid parameter overload.

To produce various channels of data from input frames, first, apply a series of hardwired kernels. In the second layer, this results in 33 feature maps in five separate channels labelled grey, gradient-x, gradient-y, optflow-x and y. Seven input frames' grey pixel values are stored in the grey channel. CNN is made up of several convolutional layers, one or more fully connected layers and an output layer.

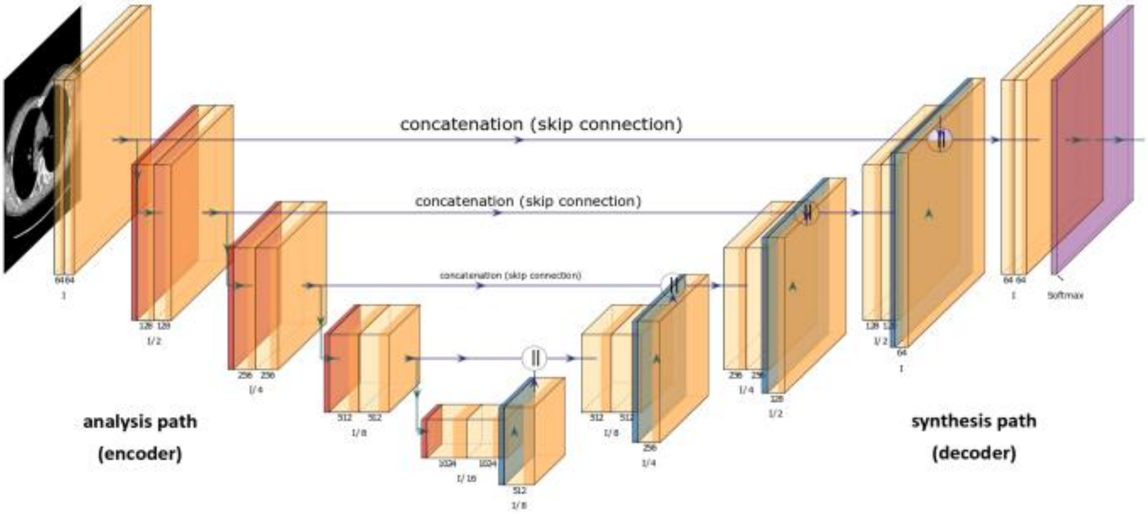


Figure-2 Volumetric scheme of F-3D CNN U-Net++

FCNs can produce promising outcomes in data-driven segmentation and classification tasks. Consider a set of input images and labels $T = I_x, L_n$, where I_x represents N's X-ray images and L n represents N's corresponding ground truth labels. By adjusting the values of several parameters, NN can find a straight mapping from theoriginal image to its segmented version. An encoder and a decoder are part of our network. The encoder has four resolution levels, each with two convolutional layers and three 3 × 3 × 3 kernels. The final layers of the decoder are the transposed 3 × 3 × 3 convolutional layers, each of which uses ReLU activations. The proposed architecture's scheme is depicted in Figure. 2. We use the same input and output volume sizes as F-3D CNN U-Net++ and uses convolution layers with 3 × 3 × 3 kernels that use zero-padding throughout to allow robust experimentation with medical pictures. To train the model, we use randomly divided subvolumes taken from a few training CT scans. Each subvolume has a scope of 64 × 64 × 64, which allows for minibatch training on a single GPU. We chose a batch size of 256 because we believe it will result in a better model approximation during training. Our architecture, like the original F-3D CNN U-Net++, uses shortcut connections in the symmetric analysis path, resulting in about 20 M trainable weights.Each volume contains 450–1167, 512 × 512-pixel slices. The voxel sizes were -chosen to be [0.57–0.91, 0.58–0.91, 0.48–0.98] mm. The arteries, gallbladder, pancreas, liver, stomach, spleen and vein are all depicted in these images. The images and tagged training data were subjected to smooth B-spline distortions. We also reduced the extension of all the original photos by four times to enable efficient calculation on a single GPU, resulting in axial image sizes of 128 × 128 × (number of slices) / 4.

Minimisation of Loss Function:SDSC is a metric that determines how similar two binary regions are. As a result, SDSC is frequently used to evaluatethe performance of various segmentation techniques. SDSC formula is as follows by Eqn (18):

$$\mathrm{SDSC} = 2 \cdot \frac{|A \cap B|}{|A| + |B|} \qquad (18)$$

where A is a set of ground truth and B denotes segmentation that was computed. Both sets are binary, with each voxel having a value of 0 or 1. We used a differentiable version of SDSC to train our F-3D CNN U-Net++ to expand theapplication of SDSC to volumetric pictures. We used the following gradient in Eqn(19)and (20) to minimise the loss function to optimise the SDSC score on training data.

$$\frac{\partial L}{\partial p_j} = 2 \cdot \frac{g_j\left(\sum_{i=1}^{r} p_i^2 + \sum_{i=1}^{r} g_j^2\right) - 2p_j\left(\sum_{i=1}^{r} p_i g_i\right)}{\left(\sum_{i=1}^{r} p_i^2 + \sum_{i=1}^{r} g_i^2\right)} \qquad (19)$$

$$L_{total} = \frac{1}{M} \sum_{j}^{M} w_j L_j \qquad (20)$$

where M denotes the total number of foreground and background classes, and wj denotes a weight indicator that can be applied to any label class j. For all labels in our application, we keep $w_j = 0.5 \times 10^{-3}$.

4. PERFORMANCE ANALYSIS:

The suggested PCRNN_ F-3D CNN U-Net++ implementation in its entirety is written in Python.

4.1 Database description

Covid-19 Open Research Dataset (CORD-19): The COVID-19 Open Studies Dataset is a growing collection of scholarly papers on COVID-19 and associated coronavirus research from the past. CORD-19's massive collection of metadata and structured full-text documents is intended to aid in the development of text mining and data retrieval systems. In this post, it go over the mechanics of dataset generation, highlighting roadblocks and important design decisions, as well as an overview of how CORD-19 are utilized and numerous shared activities that have sprung up around it.

COVID-19 X-ray dataset: Suggested dataset was created using Pneumonia dataset Chest X-ray Images. Dataset is divided into two folders (train and test), with subfolders for each image category. There are 306 X-ray images in four categories .

CheXNet: CheXNet is a neural network that has undergone two pieces of training (on ImageNet and ChestX-ray14). With 112,120 pictures from 30,805 patients, ChestX-ray14 is one of the largest chest X-ray datasets. The images were obtained by theNational Institutes of Health of the US. It includes photos of 14 different diseases as well as images with no findings.

4.2 Performance Metrics

The results of the experiment are analysed using Python software, with the parameters accuracy, precision, recall, and F1 score being used. TP (True Positive), FN (False Negative), TN (True Negative) and FP (False Positive) are utilized to evaluate parameters.

Accuracy: It is defined as total precisely predicted values to total number of predictions as in equation (21)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (21)$$

Recall: It is the precisely predicted values to the total prediction values as in equation (22)

$$Recall = \frac{TP}{TP+FN} \qquad (22)$$

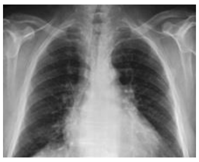
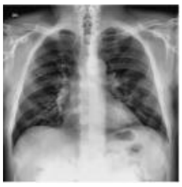
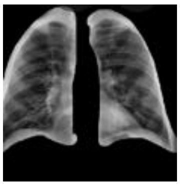
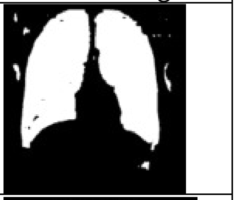
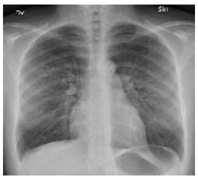
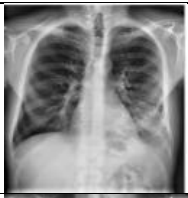
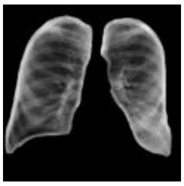
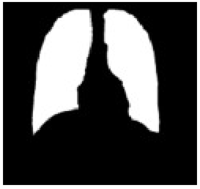
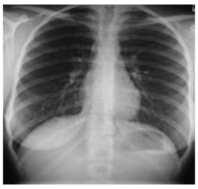
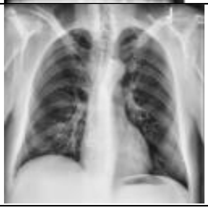
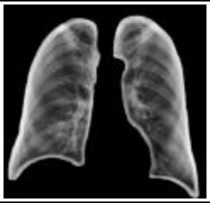
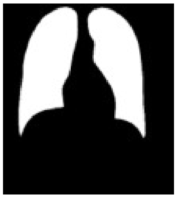
Precision: It is the true positive values to the total predicted values as in equation (23)

$$Precision = \frac{TP}{TP+FP} \qquad (23)$$

F1 - Score: It is theratio of average precision and recall as stated in equation (24)

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} (24)$$

Table-1 Comparative analysis in chest X-ray image classification for various datasets

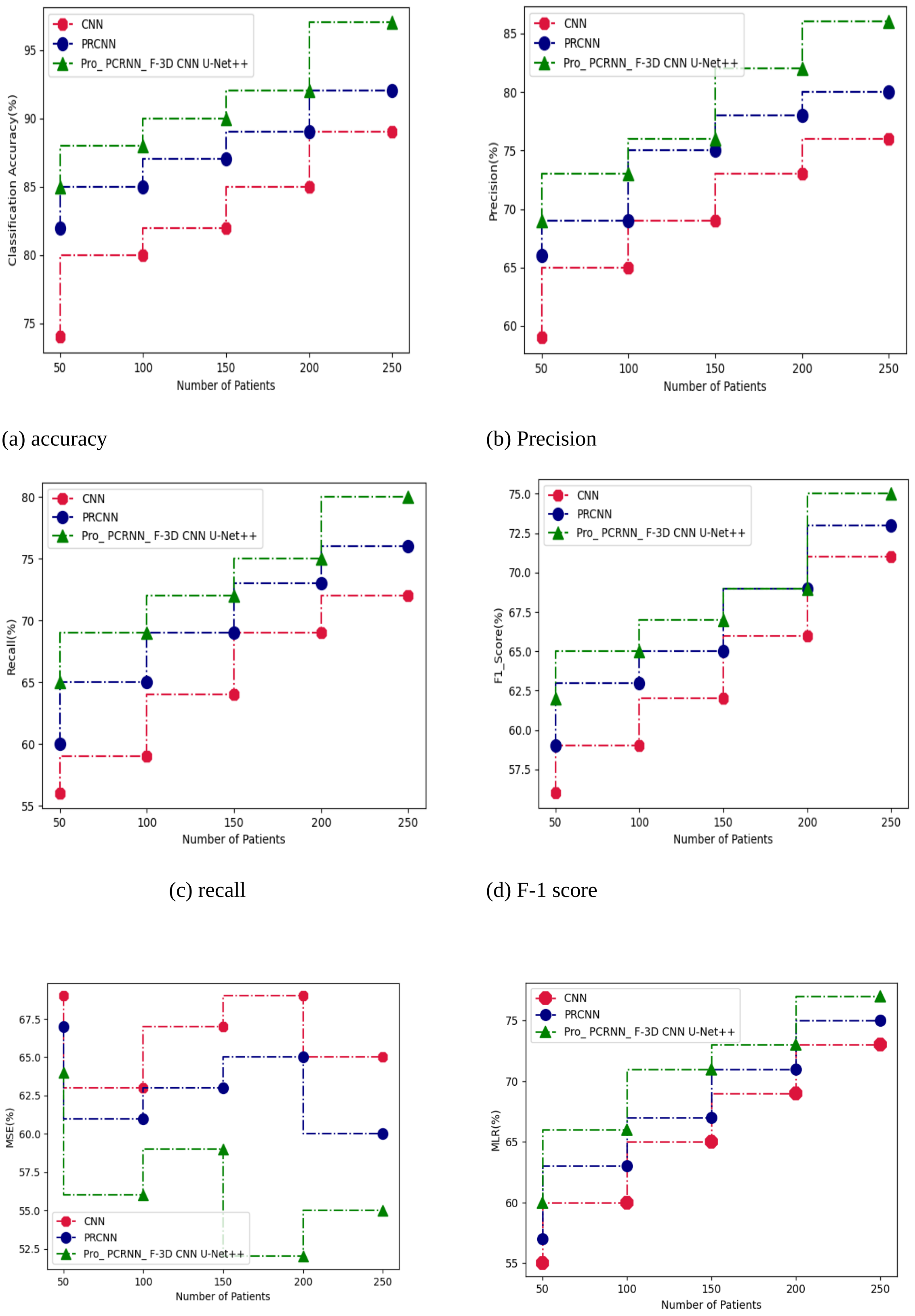
Dataset	Chest X-ray Input Image	Pre-processed image	Segmented image	F-3D CNN U-Net++ based classified image
CORD-19 Dataset				
COVID-19 X-ray Dataset				
CheXNet Dataset				

The above table-1 shows comparative analysis in chest X-ray image classification for various input datasets. The comparison has been made based on the image classification for identifying the risk of covid for diabetes patients from the collected input image of diabetes. Among all the techniques compared above, the proposed F-3D CNN U-Net++ based classification has obtained the enhanced and précised output in analysing the input chest X-ray image.

Table-2 Comparative analysis of chest X-ray processing based parameters

Dataset	Techniques	Classification accuracy	Precision	Recall	F1 score	MSE	MLR
CORD-19 Dataset	CNN	88	76	72	71	65	73
	PRCNN	92	80	76	73	60	75
	Pro_ PCRNN_ F-3D CNN U-Net++	97	86	80	75	55	77
COVID-19 X-ray Dataset	CNN	80	79	75	65	55	75
	PRCNN	85	83	79	70	50	78
	Pro_ PCRNN_ F-3D CNN U-Net++	92	88	82	75	45	80
CheXNet Dataset	CNN	86	82	78	75	52	73
	PRCNN	90	86	82	80	49	76
	Pro_ PCRNN_ F-3D CNN U-Net++	96	90	85	85	48	85

The above table-2 shows the comparative analysis for chest X-ray image processing based parameters in terms of accuracy, precision, recall, F-1 score, MSE and MLR. Comparative analysis is carried out for various datasets based on COVID X-ray datasets.



(a) accuracy

(b) Precision

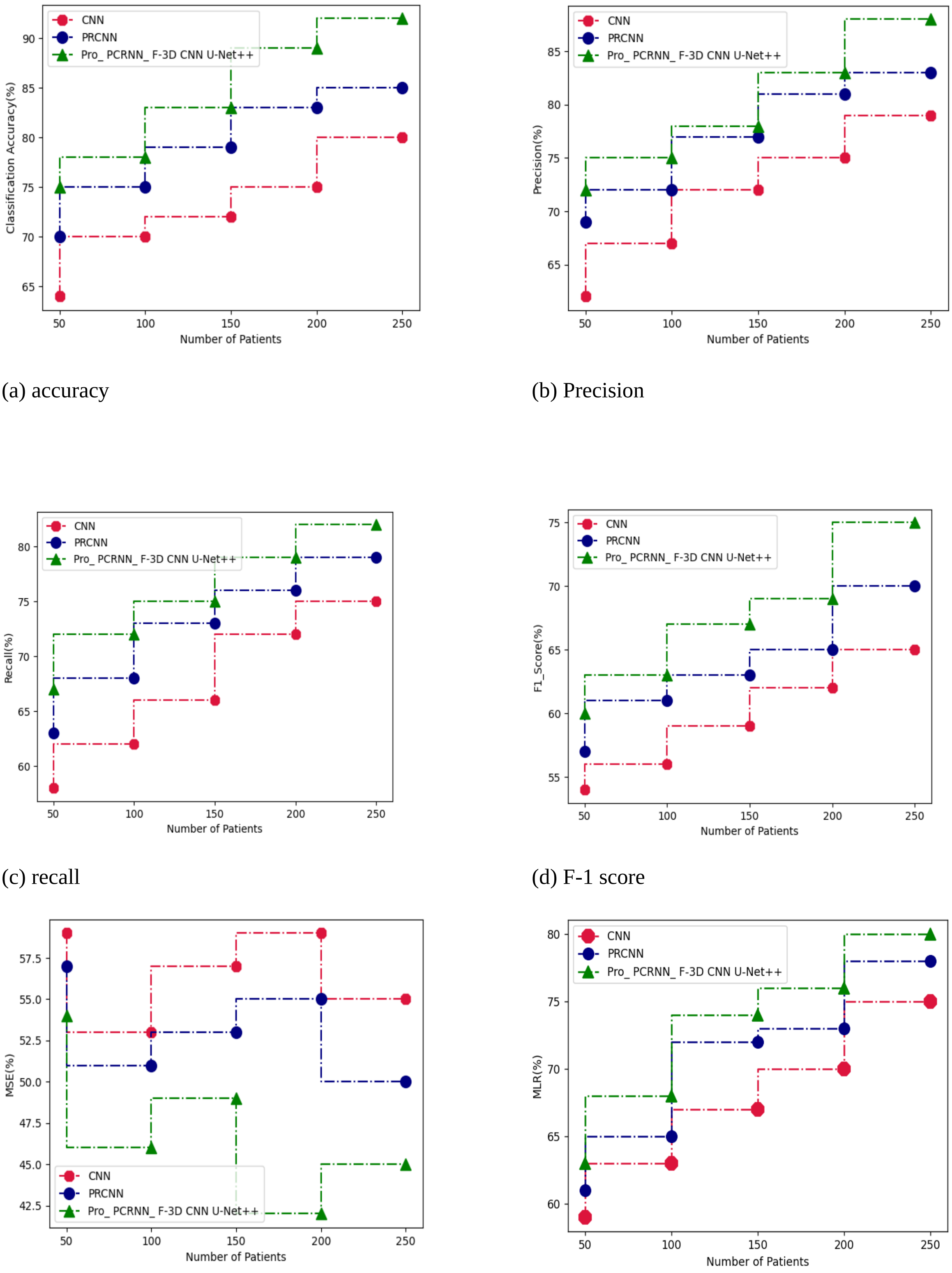
(c) recall

(d) F-1 score

(e) MSE

(d) MLR

Figure-3 Comparative analysis of CORD-19 Dataset in terms of (a) accuracy, (b) precision, (c) recall, (d) F-1 score, (e) MSE and (f) MLR



(a) accuracy

(b) Precision

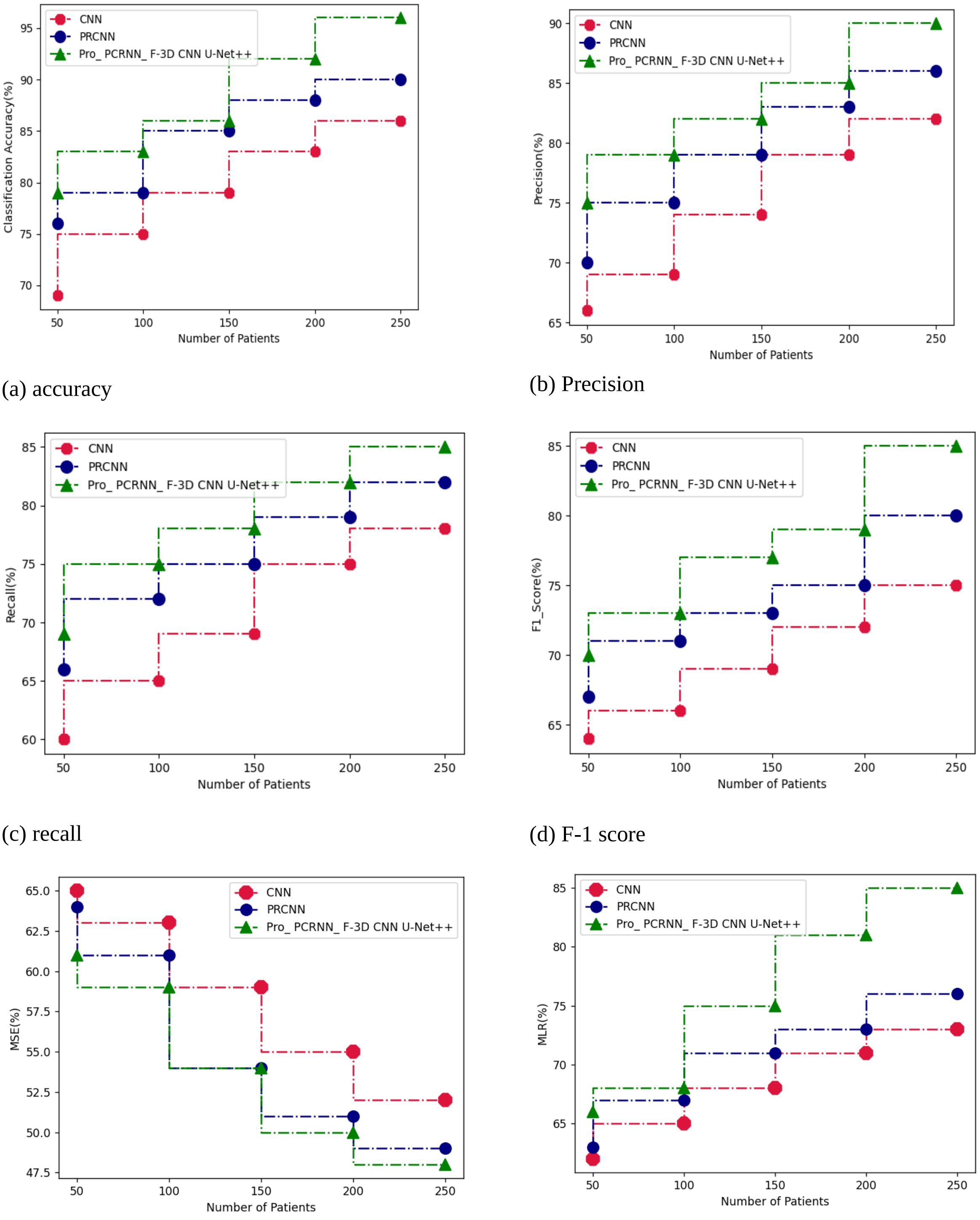
(c) recall

(d) F-1 score

(e) MSE

(d) MLR

Figure-4 Comparative analysis of COVID-19 X-ray Dataset in terms of (a) accuracy, (b) precision, (c) recall, (d) F-1 score, (e) MSE and (f) MLR



(a) accuracy

(b) Precision

(c) recall

(d) F-1 score

(e) MSE

(d) MLR

Figure-5 Comparative analysis of CheXNet Dataset in terms of (a) accuracy, (b) precision, (c) recall, (d) F-1 score, (e) MSE and (f) MLR

The above figure- 3,4, and 5 shows the comparative analysis of chest X-ray images for various dataset in terms of accuracy, precision, recall, F-1 score, MSE and MLR. Existing techniques compared are PRCNN and CNN with proposed techniques. For all the datasets analysed the proposed technique obtained optimal results based on enhancing the accuracy and minimizing the errors.

5. CONCLUSION

This paper proposesanovel method in predicting COVID- 19 to analyse the death rate due to thepresence of diabetes. The aim is to collect the dataset of COVID- 19 patients and process them for both numerical and image data. Pre-processing and segmentation have been carried out for both numerical and image data of diabetes patients with COVID-19. Data has been classified based on numerical and image data, the numerical data has been classified using PCRNN and image data classification is done using F-3D-CNN-U net++. By diagnosing the diabetes range and coronavirus affected part of thelungs, the death rate has been calculated. The experimental results show in terms of accuracy, precision, recall, F-1 score, true positive, false positive rate, MSE, MLR for the proposed technique with various COVID-19 datasets.

REFERENCES

1. Asraf, A., Islam, M. Z., Haque, M. R., & Islam, M. M. (2020). Deep learning applications to combat novel coronavirus (COVID-19) pandemic. *SN Computer Science*, *1*(6), 1-7.

2. Lalmuanawma, S., Hussain, J., & Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons & Fractals*, *139*, 110059.

3. Ho, T. T., Park, J., Kim, T., Park, B., Lee, J., Kim, J. Y., ... & Choi, S. (2021). Deep learning models for predicting severe progression in COVID-19-infected patients: a retrospective study. *JMIR medical informatics*, *9*(1), e24973.

4. Kivrak, M., Guldogan, E., & Colak, C. (2021). Prediction of death status on the course of treatment in SARS-COV-2 patients with deep learning and machine learning methods. *Computer Methods and Programs in Biomedicine*, *201*, 105951.

5. Lassau, N., Ammari, S., Chouzenoux, E., Gortais, H., Herent, P., Devilder, M., ... & Blum, M. G. (2021). Integrating deep learning CT-scan model, biological and clinical variables to predict severity of COVID-19 patients. *Nature communications*, *12*(1), 1-11.

6. Ayyoubzadeh, S. M., Ayyoubzadeh, S. M., Zahedi, H., Ahmadi, M., & Kalhori, S. R. N. (2020). Predicting COVID-19 incidence through analysis of google trends data in Iran: data mining and deep learning pilot study. *JMIR public health and surveillance*, *6*(2), e18828.

7. Hassan, A., & Mahmood, A. (2018). The convolutional recurrent deep learning model for sentence classification. *Ieee Access*, *6*, 13949-13957.

8. Radiuk, P. (2020). Applying 3D U-Net architecture to the task of multi-organ segmentation in computed tomography. *Applied Computer Systems*, *25*(1), 43-50.

9. Alimadadi, A., Aryal, S., Manandhar, I., Munroe, P. B., Joe, B., & Cheng, X. (2020). Artificial intelligence and machine learning to fight COVID-19.

10. Sankaranarayanan, S., Balan, J., Walsh, J. R., Wu, Y., Minnich, S., Piazza, A., ... & Jenkinson, G. (2021). Covid-19 mortality prediction from deep learning in a large multistate electronic health record and laboratory information system data set: Algorithm development and validation. *Journal of medical Internet research*, *23*(9), e30157.

11. R. Zagrouba, M. Adnan Khan, A. ur-Rahman et al., “Modelling and simulation of COVID-19 outbreak prediction using supervised machine learning,” Computers, Materials & Continua, vol. 66, no. 3, pp. 2397–2407, 2021.

12. A. Kumar, A. Sharma, and A. Arora, “Anxious depression prediction in real-time social data,” in Proceedings of the International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019, pp. 1–7, Dehradun, India, July 2019

13. S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, “Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: a review,” Chaos, Solitons & Fractals, vol. 139, p. 110059, 2020.

14. M. E. H. Chowdhury, T. Rahman, A. Khandakar et al., “An early warning tool for predicting mortality risk of COVID-19 patients using machine learning,” 2020, http://arxiv.org/abs/ 2007.15559.

15. M. Nemati, J. Ansary, and N. Nemati, “Machine-learning approaches in COVID-19 survival analysis and dischargetime likelihood prediction using clinical data,” Patterns, vol. 1, no. 5, p. 100074, 2020.

16. L. Yan, H.-T. Zhang, Y. Xiao et al., “Prediction of criticality in patients with severe Covid-19 infection using three clinical features: a machine learning-based prognostic model with clinical data in Wuhan,” medRxiv, 2020.

17. K. C. Y. Wong and H.-C. So, “Uncovering clinical risk factors and prediction of severe COVID-19: a machine learning approach based on UK biobank data,” medRxiv, 2020.

18. L. Sun, F. Song, N. Shi et al., “Combination of four clinical indicators predict the severe/critical symptom of patients infected COVID-19,” Journal of Clinical Virology, vol. 128, p. 104431, 2020.