BioMedInformatics

MDPI

Article

An Efficient COVID-19 Mortality Risk Prediction Model Using Deep Synthetic Minority Oversampling Technique and Convolution Neural Networks

Rajkumar Soundrapandiyan [1], Adhiyaman Manickam [2], Moulay Akhloufi [2,*], Yarlagadda Vishnu Srinivasa Murthy [1], Renuka Devi Meenakshi Sundaram [3] and Sivasubramanian Thirugnanasambandam [4]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India; rajkumars@vit.ac.in (R.S.); vishnu.murthy@vit.ac.in (Y.V.S.M.)
[2] Perception, Robotics and Intelligent Machines (PRIME), Department of Computer Science, Univeristy of Moncton, Moncton, NB E1A 3E9, Canada; adhiyaman.manickam@umoncton.ca
[3] Department of Computer Applications, Sri Krishna Arts and Science College, Coimbatore 641008, India; renukadevim@skasc.ac.in
[4] Wavicle Data Solutions, Coimbatore 641028, India; sivasubramanian.thiru@wavicledata.com
* Correspondence: moulay.akhloufi@umoncton.ca

Abstract: The COVID-19 virus has made a huge impact on people's lives ever since the outbreak happened in December 2019. Unfortunately, the COVID-19 virus has not completely vanished from the world yet, and thus, global agitation is still increasing with mutations and variants of the same. Early diagnosis is the best way to decline the mortality risk associated with it. This urges the necessity of developing new computational approaches that can analyze a large dataset and predict the disease in time. Currently, automated virus diagnosis is a major area of research for accurate and timely predictions. Artificial intelligent (AI)-based techniques such as machine learning (ML) and deep learning (DL) can be deployed for this purpose. In this, compared to traditional machine learning techniques, deep Learning approaches show prominent results. Yet it still requires optimization in terms of complex space problems. To address this issue, the proposed method combines deep learning predictive models such as convolutional neural network (CNN), long short-term memory (LSTM), auto-encoder (AE), cross-validation (CV), and synthetic minority oversampling techniques (SMOTE). This method proposes six different combinations of deep learning forecasting models such as CV-CNN, CV-LSTM+CNN, IMG-CNN, AE+CV-CNN, SMOTE-CV-LSTM, and SMOTE-CV-CNN. The performance of each model is evaluated using various metrics on the standard dataset that is approved by The Montefiore Medical Center/Albert Einstein College of Medicine Institutional Review Board. The experimental results show that the SMOTE-CV-CNN model outperforms the other models by achieving an accuracy of 98.29%. Moreover, the proposed SMOTE-CV-CNN model has been compared to existing mortality risk prediction methods based on both machine learning (ML) and deep learning (DL), and has demonstrated superior accuracy. Based on the experimental analysis, it can be inferred that the proposed SMOTE-CV-CNN model has the ability to effectively predict mortality related to COVID-19.

Keywords: artificial intelligence; auto-encoder; convolutional neural network; cross-validation; long short-term memory; synthetic minority; oversampling

Citation: Soundrapandiyan, R.; Manickam, A.; Akhloufi, M.; Murthy, Y.V.S.; Sundaram, R.D.M.; Thirugnanasambandam, S. An Efficient COVID-19 Mortality Risk Prediction Model Using Deep Synthetic Minority Oversampling Technique and Convolution Neural Networks. *Biomedinformatics* **2023**, *3*, 339–368. https://doi.org/10.3390/biomedinformatics3020023

Academic Editor: Yasunari Matsuzaka

Received: 28 March 2023
Revised: 19 April 2023
Accepted: 28 April 2023
Published: 5 May 2023

1. Introduction

The COVID-19 virus is a new type of virus that was first identified in Wuhan, China, in November 2019. It quickly spread around the world, causing a pandemic that has affected over 76 million people and resulted in the deaths of over 6 million people as of 20 April 2023. While most people experience mild to moderate respiratory illness and recover

without special treatment, older individuals and those with certain medical conditions are at higher risk of suffering from the virus, which increases the mortality rate of COVID-19 [1]. Symptoms of COVID-19 are similar to those of the flu and can be difficult to detect at an early stage, making efficient testing and treatment systems crucial [2]. The current method for detecting COVID-19, RT-PCR testing, is time-consuming and expensive, with a low detection rate. Hence, researchers and physicians are turning to artificial intelligence (AI)-based applications for various purposes, such as early disease prediction, disease diagnosis, professional decision assistance, patient support, and mortality risk assessment for serious illnesses.

Lately, AI is rapidly growing with machine learning (ML) and deep learning (DL) algorithms. DL algorithms have shown promising results in various medical applications such as classifying skin cancer, detecting breast cancer, classifying pneumonia, and segmenting lungs. DL algorithms such as convolutional neural networks (CNNs), long short-term memory (LSTM), and auto-encoder are the most widely used techniques these days [3,4]. This study investigates the effectiveness of both CNN and LSTM models in forecasting COVID-19 mortality. The incorporation of data-augmentation techniques has further improved the performance of these models, which is discussed in Section 4.3. To assess the mortality risk of COVID-19 patients, these models are compared and evaluated against current predictive models in both the deep learning and machine learning domains.

1.1. Contributions

The main contributions of this research are:

- Generating the following six deep learning predictive models will help identify people with the COVID-19 disease that are at higher risk of mortality.
    - i CV-CNN: A clinical dataset of 4711 individuals is used in building this model, and it is trained using 10-fold cross-validation.
    - ii CV-LSTM+CNN: The LSTM method and a CNN model are combined to create this model. Additionally, a 10-fold cross-validation method is utilized in its training.
    - iii IMG-CNN: This model is a CNN and is trained using converted images of the clinical dataset where each image corresponds to one record.
    - iv AE+CV-CNN: This model is built by combining an auto-encoder and CNN model with 10-fold cross-validation.
    - v SMOTE-CV-LSTM: This model is established by integrating SMOTE and LSTM techniques along with 10-fold cross-validation.
    - vi SMOTE-CV-CNN: This model is established by integrating SMOTE and CNN techniques along with 10-fold cross-validation.
- Estimating a patient's probability of survival based on their medical records.
- Locating important biomarkers that can tell us the severity of the diseases.
- The medical dataset is transformed as images and is applied to the proposed IMG-CNN algorithm.
- Assessing the suggested model and comparing it with earlier research work.
- Improving the model performance using data-augmentation techniques.

1.2. Organization of Paper

The rest of the paper is organized as follows: Section 2 describes the literature survey on predicting mortality risk and severity. Section 3 presents the application scenario of the proposed models. Section 4 presents the methodology in which Section 4.1 discusses the dataset, Section 4.2 explains the preprocessing process, Section 4.3 explores the data-augmentation techniques, and Section 4.4 elaborately presents the proposed models. Section 5 presents the experimental results and discussions. Finally, Section 6 concludes the paper.

2. Literature Survey

Using deep learning modules, Tekerek [5] suggested an architecture for anomaly-based web attack detection in a web application. The datasets that were considered were CSIC2010v2 datasets. As one of the most well-known datasets, it is frequently used for online application security. The preprocessing that they used can be broken down into six steps: cleaning, missing values, noisy data, integration, reduction, and transformation. The model that was considered for this application was a CNN model. CNN is usually used in image-processing studies. It is a special case of neural networks where at least one of the layers is a convolutional layer. The final evaluation was performed by considering the F1-score, precision, recall, accuracy, false negative rate (FNR), false positive rate (FPR), true negative rate (TNR), and true positive rate (TPR).

Xu et al. [6] actively collected more and more COVID-19 patient data. They provide real-time case information for people to use. Both formal government websites and peer-reviewed scientific papers that present primary data serve as the sources for the data. The official websites of the health departments and the social media profiles of government and public health organizations are examples of government sources. The data provided by them have been used by numerous authors to train their models. The database has 32 features. After some extraction, the features can go up to 83. The authors used one hot encoding on the data of the symptoms. The authors also provide a fixed dataset that does not have real-time case information. In the paper, they discuss their data collection and augmentation techniques.

Banoei et al. [7] considered a dataset containing 250 clinical features. In this dataset, only 108 clinical features, comorbidities, and blood markers recorded at the moment of admission from a hospitalized cohort of patients are subjected to multivariate predictive analysis. The mortality risk of COVID-19 individuals was forecasted using a partial least squares-based model. The model was able to forecast patient mortality with good accuracy (AUC $>$ 0.85) and average predictive power (Q2 = 0.24).

The use of supervised learning techniques and their capacity to decipher intricate patterns in actual medical data were the main topics of Bikku's [8] research. Perceptions with multiple layers make up the model. A feed-forward neural network called an MLP has an output layer, one or more hidden layers, and at least one input layer. Each stratum serves a distinct purpose. To determine the most important health variables and forecast the mortality risk of COVID-19 patients, an integrated predictive model utilizing a decision tree (DT), support vector machine (SVM), logistic regression (LR), random forest (RF), and K-nearest neighbor (KNN) was put into place. The UCI ML repository provided the sample that was used. It consisted of records of more than 2,670,000 patients affected by COVID-19. It was collected from 146 countries across the globe. In order to determine other metrics, such as accuracy, precision, recall, specificity, and F1-score, the assessment is carried out using a confusion matrix. The model's obtained accuracy is 89.98%.

Yan et al. [9] used blood sample data belonging to 485 patients infected by COVID-19 in Wuhan, China. The objective was to find crucial predictive biomarkers of disease mortality. The feature importance was identified by ranking the features using a multi-tree XGBoost. Lactic dehydrogenase, lymphocytes, and C-reactive protein are some crucial disease indicators. This information is used as input into a decision tree algorithm to identify patients who require urgent medical care more than 10 days in advance. The patient information included blood sample information of all patients collected between 10 January and 18 February 2020. Using categorization accuracy, precision, sensitivity, recall, and F1-scores, the model's performance was assessed. The model's precision was a little bit higher than 90%.

A novel technique, the CNN-AE, which is a CNN trained with clinical data from patients with COVID-19 damage, was suggested by Khozeimeh et al. [10]. The dataset used by the authors is a very unbalanced one. It was a sample of 320 patients out of which 300 patients recovered and 20 died. This would result in a very poor model, however, with the inclusion of an auto-encoder model that produces more artificial samples of lower class

(dead patients). The 20 instances of the lower class are given as input to an AE model that compresses and decompresses the data. The outcome of the model is 20 artificial samples. By repeating this process several times, the authors were able to balance the dataset. This dataset produced high accuracy in training. The precision of the suggested model was 96.05%.

Pourhomayoun and Shakibi [2] used several machine learning techniques for mortality risk prediction. Some of these are SVM, artificial neural networks, DT, LR, and KNN. In the process, they were also able to identify the most important symptoms that would be useful for prediction. A total of 2,670,000 samples from 146 different nations made up the information used for this procedure. Out of these, 307,000 samples were labeled samples. The performance of each technique was tabulated and presented. The evaluation criteria were accuracy, receiver operating characteristic (ROC), AUC, and the confusion matrix.

Khan et al. [11] implemented several machine learning algorithms as well as deep learning algorithms. Decision tree, logistic regression, random forest, extreme gradient boosting, and K-nearest neighbor are some of the machine learning methods that were used. A DL model contains six layers: an input layer, a CONV layer, a max pooling layer, a ReLU layer, a fully connected layer, and an output layer with softmax implemented. The dataset used for this purpose had 2,676,000 samples collected from 146 countries. The data were split using K-fold cross-validation that was then used for each model training.

Tezza et al. [12] used machine learning techniques to predict mortality risk. The algorithms used were recursive partition tree, gradient boosting machine, random forest, and SVM. The data used came from patients who were admitted to the COVID-19 referral facility 'Ospedali Riuniti Padova Sud' in the Veneto area of Italy. The patients being considered were diagnosed with COVID-19 after undergoing a PCR test. The number of patients was 374 with a median age of 74. The algorithms were evaluated using specificity, sensitivity, and ROC curve measures. The final results concluded that the random forest technique outperformed the other algorithms. Their results concluded that the RF algorithm produced the best performance with a ROC of 0.84. The study also revealed important biomarkers for predicting mortality risk-age, oxygen saturation levels, creatine, AST, and hemoglobin.

According to Wang et al. [13], the XGBoost algorithm is effective in producing mortality prediction models for COVID-19. The First People's Hospital in Wuhan's Jiangxia District provided the dataset for this research. The model was trained using a cohort of 296 individuals with information on their age, history of hypertension, and occurrence of coronary heart disease. The dataset had around 96% recovered patients and thus was highly unbalanced. The model achieved an AUC of 83% and provided some insight into the biomarkers that are key in predicting COVID-19 severity. Some of these were lymphocyte count, age, D-dimer, oxygen saturation, and glomerular filtration rate.

An XGBoost-based prognostic prediction model was trained using a sample of 375 affected patients and 201 survivors [14]. The data were collected from the Tongji Hospital in Wuhan, China. The model achieved more than 90% accuracy on mortality prediction and revealed key biomarkers such as LDH, hs-CRP, and lymphocyte count.

The SVM method was used by Sun et al. [15] to create a predictive model for the severity prediction of COVID-19. The data were collected from the Shanghai Public Health Clinical Center, which comprised a cohort of more than 330 patients. The model could successfully distinguish between mild and severe cases with an AUC value of 97.57%. This study recognizes four features out of 220 features as the most important to predict COVID-19 severity. They were age, GSH, CD3 ratio, and total protein.

The XGBoost algorithm was used by Rechtman et al. [16] to create a predictive model that was trained on patient data from a New York City healthcare system. A dataset of 8770 patients with 7656 survivors was used in the study. The Mount Sinai health facility in New York City is where the information was gathered. The model produced an AUC of 0.86 and concluded that age, gender, and high body mass index (BMI) are key features in predicting disease severity.

Early mortality risk detection in COVID-19 patients using an ML method was covered by Hu et al. [17]. The 183 patient entries in the dataset used for this purpose are from the Sino-French New City branch of Wuhan's Tongji hospital. They used details of an additional 64 patients to externally validate their predictive model. The authors used ten different approaches out of which only five proved to be effective. These included flexible discriminant analysis, partial least squares regression, elastic net model, and LR and RF. The performance of all the algorithms was similar with an AUROC value of 88.1%. Three important biomarkers—hs-CRP level, cell count, and D-Dimer levels—were discovered.

An algorithm created by Zhou et al. [18] can forecast the severity of a COVID-19 infection. The dataset used contains details of 377 patients (172 severe), and it is collected from Wuhan's Central Hospital. The model performed with an AUC of 87.9%. Key biomarkers identified include age, CRP, and D-dimer levels.

A prediction model developed by Li et al. [19] utilizes clinical and laboratory data to forecast the mortality risk of COVID-19 patients. The GitHub dataset and the Wolfram dataset were both used by the writers. The forecast model is constructed using an auto-encoder. Other algorithms, including LR, RF, SVM, SVM one-class models, and isolation forest, are used to evaluate the model's performance. The study also revealed that persons with chronic conditions were at a higher risk of COVID-19-related death.

In order to approximate the infection severity in COVID-19 patients, Zhu et al. [20] suggested a model. The dataset is collected from Ningbo's Hwa Mei Hospital and has 126 records. The LR method was used to build the model, which has an AUC value of 90.0%. The model was effective in pinpointing a number of causes of severe instances of COVID-19. The neutrophil–lymphocyte ratio, fibrinogen, sialic acid, CRP, and relative pressure of oxygen were a few of these. Table 1 provides an overview of the current COVID-19 diagnostic techniques.

Table 1. Summary of the existing COVID-19 diagnostic methods.

Study	Method	ML/DL	Performance
[8]	SVM, NN and RF	ML	89.98% Accuracy
[9]	XGBoost	ML	Accuracy > 90%
[10]	CNN	DL	96.05% (Avg Accuracy)
[12]	RPART, SVM, CBM and RF	ML	RF Model Performance (84% ROC)
[13]	XGBoost	ML	83% AUC Clinical Model 88% AUC Laboratory Model
[14]	SVM	ML	91% specificity 91% sensitivity
[15]	SVM	ML	97.57% AUC
[16]	XGBoost	ML	86% AUC
[17]	Partial least squares regression, elastic net model, RF, Bagged FDA and LR	ML	88.1% AUC 79.4 % Specificity 83.9% Sensitivity
[18]	LR	ML	73.7% Specificity 88.6% Sensitivity
[19]	AE, LR, RF, SVM, one-class SVM, isolation forest and local outlier factor	ML	97% Accuracy and 73% AUC
[20]	LR	ML	90% AUC
[21]	RF and ANN	ML	90. 83% Accuracy
[22]	Deep-Risk	DL	94.14% Accuracy

Table 1 suggests that machine learning methods have been more frequently used than deep learning methods for COVID-19 mortality prediction using clinical datasets. In order to address these limitations, alternative six deep learning models were implemented.

The later sections of this paper describe the application of the proposed models followed by the dataset used and the methodology. The pseudo-code and a thorough description of each model are given. After this, the results are discussed along with the inferences and conclusion.

3. Application

We discuss the application aspects of this specific research in this section. This study falls under the category of predictive analytics in the healthcare industry. The introduction of artificial intelligence in healthcare has drawn a lot of attention to predictive analytics as well. Using predictive analytics in healthcare has a number of benefits, some of which are discussed in this section. Figure 1 represents the application scenario of the proposed efficient COVID-19 mortality risk prediction system.

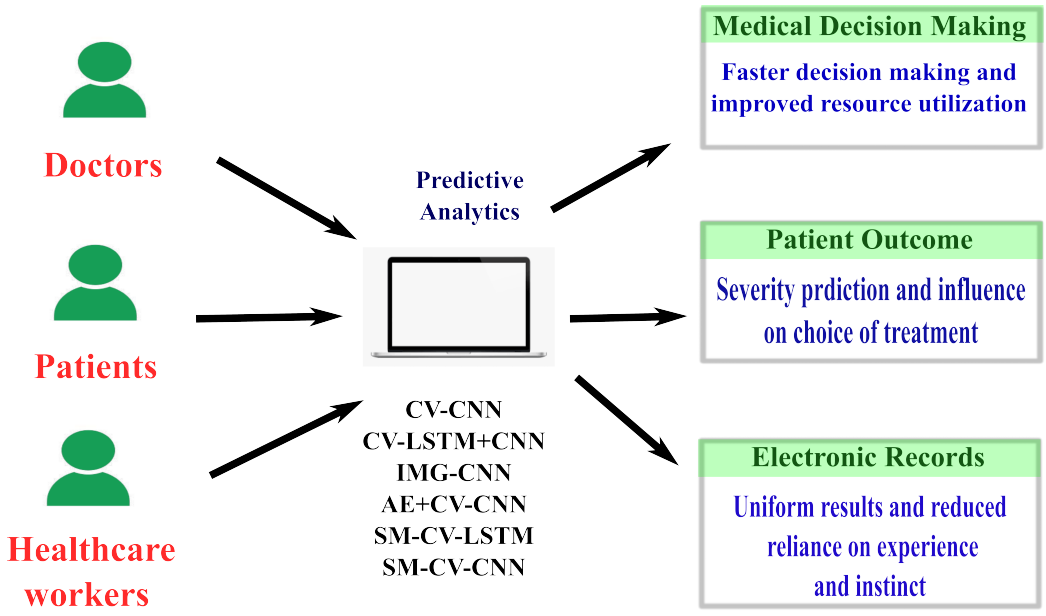

Figure 1. Application scenario of the proposed efficient COVID-19 mortality risk prediction system.

3.1. Medical Decision Making

Using predictive analytics to support decision making is its most significant addition to the healthcare sector. The amount of scientific information gathered about COVID-19 in just a small amount of time is extraordinary. However, uncertainty in medicine is unavoidable and one can never make a generalized assumption. The only method being used to diagnose COVID-19 is the viral PCR test, which has proven to have a lower rate of false negatives. However, information about the severity of the infection is still unavailable. Using the proposed prediction models in this study to predict the mortality risk of infected persons can be life-saving. It facilitates decision making and, to some degree, influences the choice of treatment. This would allow medical institutions to decide how they utilize their available resources.

3.2. Improving Patient Outcomes

All medical institutions store large amounts of patient information, which includes data about chronic health conditions, family illnesses, and so on. With the help of this information, patient results can be improved through mortality prediction. The models proposed in this study can identify warning signs before the infection becomes severe. The severity of the infection can vary from person to person. By identifying key biomarkers

that contribute to COVID-19 severity, our models can identify patients who are at a greater risk of fatality.

3.3. Healthcare Workers

Medical professionals spend around 62% of their time per patient reviewing their electronic health records. By incorporating the proposed models into the healthcare system, the process can be transformed into an efficient one. Studies [23] have indicated that 90% of medical workers prefer an AI-based approach compared to relying on their instinct. They also assist the workers in the diagnostic process.

4. Methodology

4.1. Dataset

The dataset being used for this research is from a healthcare surveillance software package [24,25]. The data were collected from a single healthcare system over a certain period. The dataset has a total of 4711 records, out of which 75% are records of recovered patients, whereas 25% are of deceased patients. Each record has a total of 85 features.

The Montefiore Medical Center/Albert Einstein College of Medicine Institutional Review Board gave its approval for the data gathering. The data are anonymized as a result of their approval of the waiver of patient-informed permission.

4.2. Preprocessing

The proposed models require the dataset to be preprocessed and is presented in three different forms:

Dataset 1: This dataset will be used as input for the CV-CNN and CV-LSTM+CNN models. All the features with more than two unique values with the exception of demographic features and score features are identified. These features have to be normalized. For this purpose, we make use of StandardScaler() from the SciKit Learn library. Followed by this, we introduce a new feature (severity_class) that reduces the severity feature from 12 different categories to 4. The top 35 variables that correlate to the 'Death' feature (headache, cough, sore throat, fever, chest pain, dizziness, shortness of breath, pneumonia, diarrhea, respiratory distress, fatigue, anorexia, cardiac disease, gasp, sputum, hypoxia, eye irritation, chills, somnolence, somnolence, septic, emesis, rhinorrhea, lesions on chest radiographs, hypertension, hypertension, heart attack, expectoration, conjunctivitis, shock, kidney failure, myalgia, old, obnubilation, myelofibrosis) are extracted to be used for our models.

Dataset 2: This dataset, a novel method of data encoding, serves as the input for the IMG-CNN model. Every row of the initial dataset is transformed into an image. This is achieved by considering the numerical value for each feature as a pixel intensity value. The top 81 features out of 85 are considered so that we can form a 9×9 image. Each pixel in this image corresponds to a particular feature. The numerical value for that feature determines the pixel intensity. The conversion is from CSV to grayscale. A sample of these images is given in Figure 2. These are then stored in the appropriate directories to be used by the model.

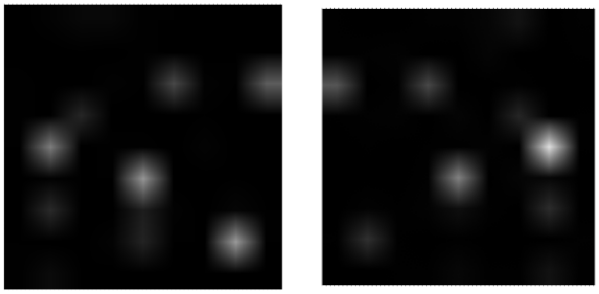

Figure 2. Sample images.

Dataset 3: This dataset will be a balanced version of dataset 1. A balanced dataset can yield better results. The dataset is balanced using two different methods. Method 1 is uses auto-encoders. Samples from the 'Dead' class are taken and passed through an auto-encoder model. The 84-dimensional data are compressed and reconstructed several times to produce augmented samples. This is used as input in the AE+CV-CNN model. The second method involves oversampling using SMOTE. Samples from the minority class are increased in a balanced way, resulting in a perfectly balanced dataset.

4.3. Data Augmentation

Three different techniques were considered for data augmentation. They were simple auto-encoders, variational auto-encoders and oversampling. On using auto-encoders to partially balance the dataset, it was identified that most of the new samples being generated contained many duplicates. Since the validity of a model trained on such a dataset is questionable, variational auto-encoders were used. On using variational auto-encoders, no duplicates were generated. However, the decompression process resulted in an output where most of the binary variables had decimal values. The outcome variable was also not in sync with the actual dataset. As a result, variational auto-encoders were rejected for this dataset. Then, SMOTE was used for data augmentation. SMOTE is a resampling technique that aims to balance datasets with a highly unbalanced ratio by creating synthetic samples in the minority class. This technique generates new samples of data in the minority class by interpolating between samples of this class that are in close vicinity of each other. SMOTE increases the number of minority class examples within an imbalanced dataset and enables the classifier to achieve better generalizability. To apply SMOTE, the desired amount of oversampling, N, should be set to an integer number, and three main steps should be taken iteratively, including randomly selecting a sample that belongs to the minority class, selecting the K (default 5) nearest neighbors of this sample, and selecting N of these K neighbors randomly for interpolation and generating new samples. The synthetic generation of new samples in SMOTE differs from the multiplication algorithm to avoid the issue of overfitting. Figure 3 provides an intuition of how SMOTE works [26]. SMOTE was used because of produced good samples that were also validated by checking with the original dataset. The performance of the models was also improved by using SMOTE.

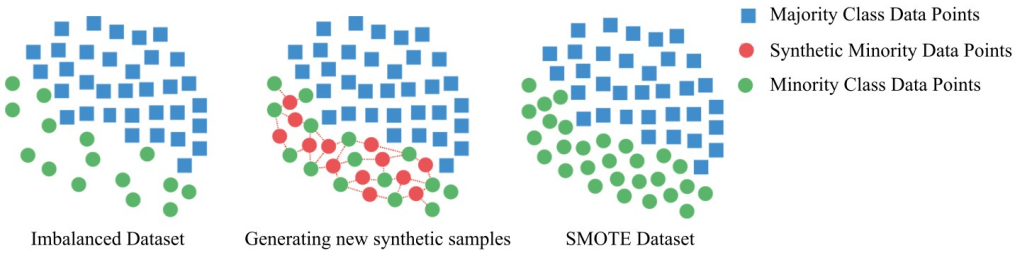

Figure 3. The visual representation of SMOTE oversampling for an imbalanced classification.

4.4. Proposed Models

Six deep learning forecasting models are taken into consideration in this study. Later, efficiency improvements are made using data augmentation. They all belong to various domains of deep learning. One of these models combines LSTM and CNN while utilizing recurrent neural networks (RNN). Another model uses unsupervised learning concepts for data augmentation and makes use of auto-encoders. The other models are regular CNN models with different input types and SMOTE.

4.4.1. CV-CNN Model

The CV-CNN model is the initial model put forth. It is a CNN model that was developed using K-fold cross-validation and dataset 1. For each test set, we conduct 10-fold cross-validation with 10 epochs in our instance. The following describes how 10-fold cross-validation functions. The dataset is split to have 10 folds with an equal number of records

in each fold. The dataset is then divided into a 9:1 training and validation group split. On each fold, the model is trained for a predetermined number of epochs. Figure 4 displays the conceptual framework of the CV-CNN model. Algorithm 1 provides the CV-CNN model abstract code.

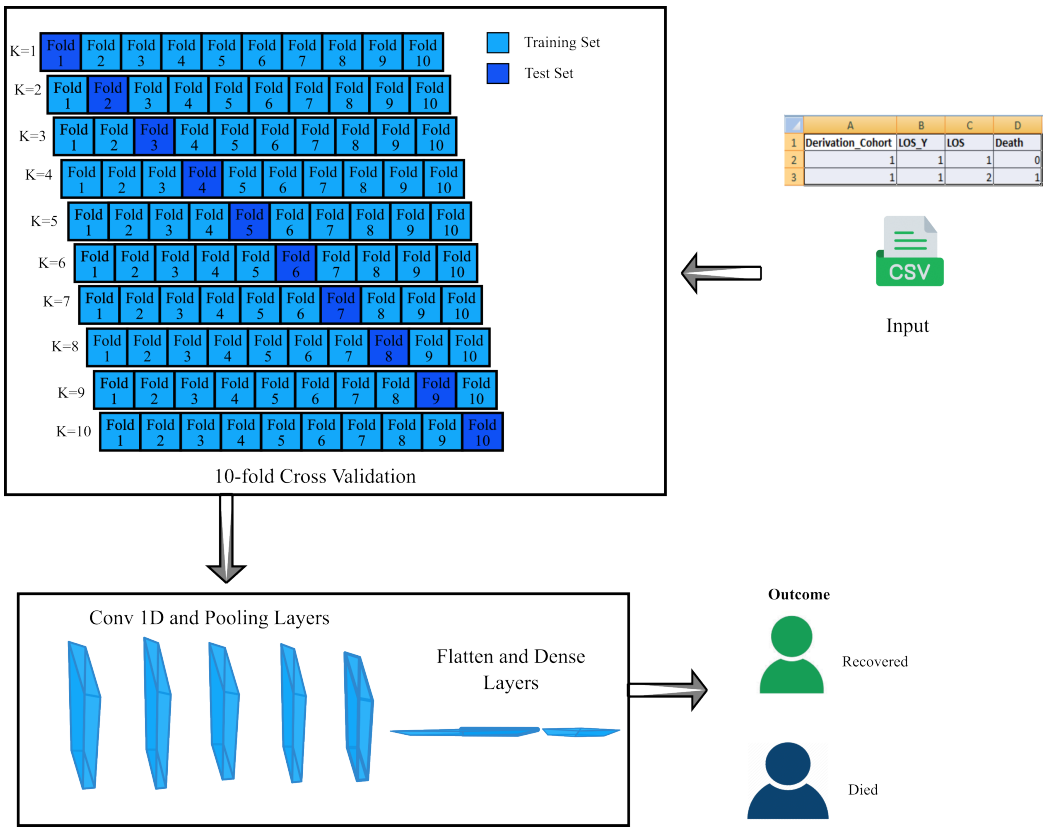

Figure 4. Conceptual framework of the CV-CNN model.

Algorithm 1 CV-CNN Model Classification

Input Clinical dataset DF: (4710 records), training epochs N, number of folds K
Output Classification

```
Create LSTM+CNN model with required parameters
Perform train–test split on DF (DTrain, DTest)
//K fold cross-validation. Partition DTrain into K subsets F1,...,Fk
for k = 1 to K do
   Dtrain = DF-Fk
   Dvalid = Fk
End for
For e = 1 to N, perform
   CNN.train(Dtrain)
   CNN.validate(Dvalid)
End for
Test = CV-CNN.test(DTest)
Classification_output = classify(CV-CNN_model, test)
End procedure CV-CNN_model
```

Three convolutional layers and four maximum pooling layers make up the CNN model itself. The one-dimensional convolution and pooling layers are used because the information is in CSV format. The input size for the model is (35,1). After the feature extraction is performed, a flattened layer is used, followed by dense layers. Given that the issue is a binary classification problem, the final output layer consists of a single neuron

with a sigmoid activation function. A summary of the CV-CNN model is given in Figure 5. The total number of trainable parameters is 806,529.

Layer (type)	Output shape	Params #
conv1d (Conv1D)	(None, 33, 256)	1024
max_pooling1d (MaxPooling1D)	(None, 32, 256)	0
conv1d_1 (Conv1D)	(None, 30, 256)	196,864
max_pooling1d_1 (MaxPooling1D)	(None, 29, 256)	0
conv1d_2 (Conv1D)	(None, 27, 256)	196,864
max_pooling1d_2 (MaxPooling1D)	(None, 26, 256)	0
max_pooling1d_3(MaxPooling1D)	(None, 25, 256)	0
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 64)	409,664
dense_1 (Dense)	(None, 32)	2080
dense_2 (Dense)	(None, 1)	33
Total params: 806,529 Trainable params: 806,529 Non-trainable params: 0		

Figure 5. Summary of CV-CNN model [22].

4.4.2. CV-LSTM+CNN Model

The dataset used for this model is dataset 1. It is a form of a recurrent neural network with the ability to retain key information from training in its long-term memory. The model can take input with a minimum of three dimensions. For this purpose, the input data are reshaped into (7,5,1). Due to this, the top 35 features from dataset 1 are considered. The training of the model is once again performed using 10-fold cross-validation as explained above. The conceptual framework of the CV-LSTM+CNN model is given in Figure 6. The pseudo-code for the CV-LSTM+CNN model execution is given in Algorithm 2. A summary of the CV-LSTM+CNN model is given in Figure 7.

Algorithm 2 CV-LSTM+CNN Model Classification

Input Clinical dataset DF: (4710 records), training epochs N, number of folds K
Output Classification

```
Create CV-LSTM+CNN model with required parameters
Perform train–test split on DF (DTrain, DTest)
// K-fold cross-validation. Partition DTrain into K subsets F1,...,Fk
For k = 1 to K do
   Dtrain = DF-Fk
   Dvalid = Fk
End for
For e = 1 to N do
   CV-LSTM+CNN.train(Dtrain)
   CV-LSTM+CNN.validate(Dvalid)
End for
Test = CV+LSTM-CNN.test(DTest)
Classification_output = classify(CV+LSTM-CNN_model, test)
End procedure CV+LSTM-CNN_model
```

In order to implement the LSTM aspect of the model, a TimeDistributed layer is used. The activation function used in this layer is 'tan h'. Since the input has been resized, the dimensions of the data changed. As a result, Conv2D and AveragePooling2D are used.

The input is then passed to a flattened layer followed by the output layer with a sigmoid activation function. The total number of trainable parameters is 830,209.

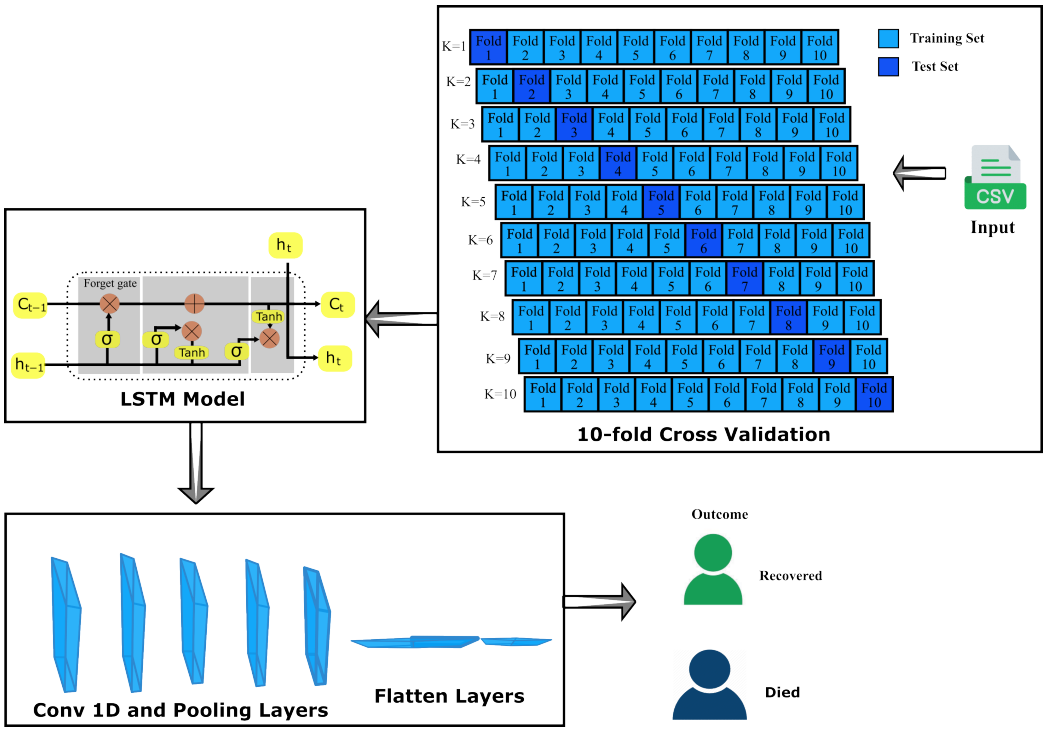

Figure 6. Conceptual framework of the CV-LSTM+CNN model.

Layer (type)	Output shape	Params #
reshape (Reshape)	(None, 7, 5, 1)	0
time_distributed (TimeDistributed)	(None, 7, 5, 256)	264,192
dropout (Dropout)	(None, 7, 5, 256)	0
time_distributed_1 (TimeDistributed)	(None, 7, 5, 256)	262,400
conv1d_1 (Conv1D)	(None, 7, 3, 256)	196,864
average_pooling2d (AveragePooling2D)	(None, 3, 1, 256)	0
flatten (Flatten)	(None, 786)	0
dropout_1 (Dropout)	(None, 786)	0
dense (Dense)	(None, 128)	98,432
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 1)	65
Total params: 803,209 Trainable params: 830,209 Non-trainable params: 0		

Figure 7. Summary of the CV+LSTM-CNN model [22].

4.4.3. IMG-CNN Model

The image CNN model requires dataset 2. The preprocessed dataset is organized in such a way that training and testing data are available with its corresponding labels. The ImageDataGenerator tool is used to take input from the directories where the images are stored. A total of 4710 records belonging to two different classes are used for training and testing. A CNN model is built and trained for 50 epochs. The conceptual framework of the IMG-CNN model is given in Figure 8. The pseudo-code of the IMG-CNN model is given in Algorithm 3. The IMG-CNN model summary is given in Figure 9.

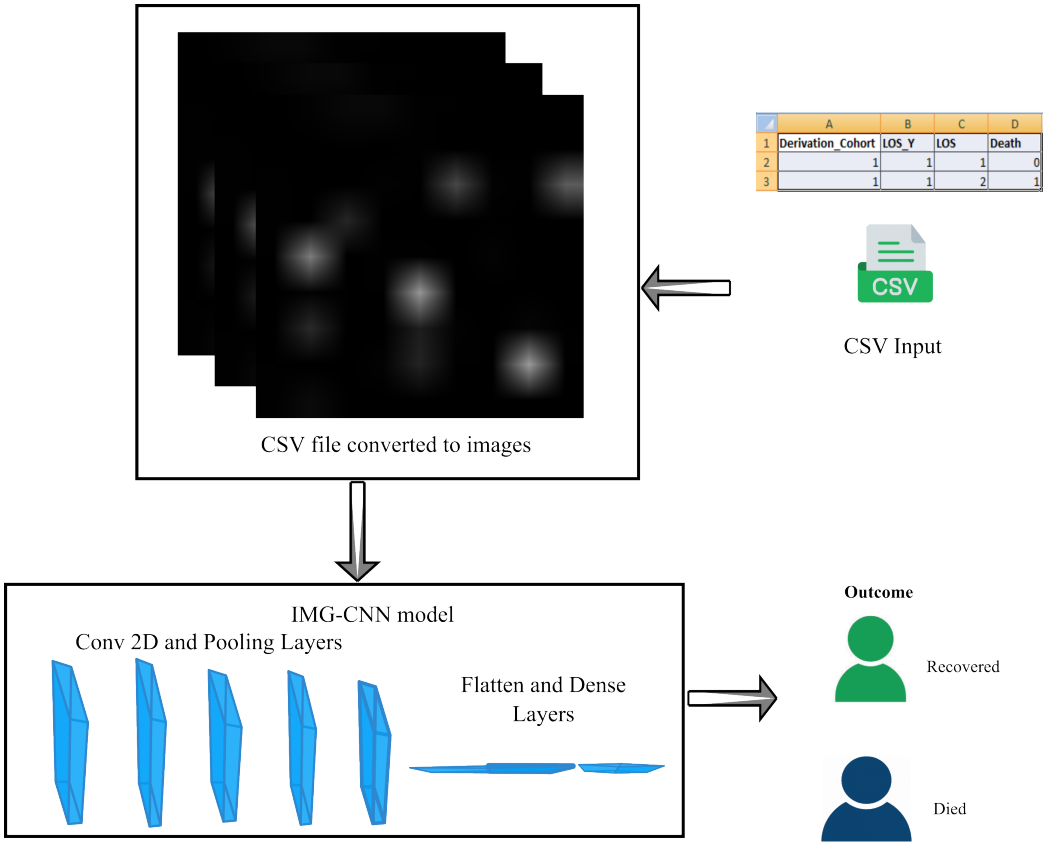

Figure 8. Conceptual framework of the IMG-CNN model.

Algorithm 3 IMG-CNN Model Classification

Input Clinical dataset DF: (4710 records), training epochs N, number of folds K
Output Classification

```
Create IMG-CNN model with required parameters
For r = 1 to R do
    Convert each record to an image
    Add image to image dataset (DS-IMG)
End for
Partition DS-IMG into train-gen and test-gen
Partition train-gen to IMG-train and IMG-val
For e = 1 to N, perform
    IMG-CNN.train(IMG-train)
    IMG-CNN.validate(IMG-val)
End for
test = IMG-CNN.test(test-gen)
Classification_output = classify(IMG-CNN_model, test)
End procedure IMG-CNN_model
```

4.4.4. AE+CV-CNN Model

This model utilizes dataset 3. The lower-class data, in our case 'Death' = 1, is passed through the auto-encoder model trained on this data. The auto-encoder compresses the 84-dimensional data into 50 dimensions. It then reconstructs these data back to 84 dimensions, but the reconstructed data are different from the original data. This would yield synthetic samples of the original data and can be used to balance the dataset. A balanced dataset would definitely produce better results than an unbalanced one. These data are then used as input in the existing CV-CNN model, thus creating an AE+CV-CNN model. The conceptual framework of the AE+CV-CNN model is shown in Figure 10. The pseudo-code

of the AE+CV-CNN model is shown in Algorithm 4. The AE+CV-CNN model summary is specified in Figure 11. The CNN model is the same as Figure 5.

Layer (type)	Output shape	Params #
conv2d (Conv2D)	(None, 148, 148, 16)	160
max_pooling2d (MaxPooling2D)	(None, 74, 74, 16)	0
conv2d_1 (Conv2D)	(None, 72, 72, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_2 (Conv2D)	(None, 34, 34, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 64)	0
flatten (Flatten)	(None, 18496)	0
dense (Dense)	(None, 512)	9,470,464
dense_1 (Dense)	(None, 1)	513
Total params: 9, 494, 273 Trainable params: 9, 494, 273 Non-trainable params: 0		

Figure 9. Summary of the IMG-CNN model [10].

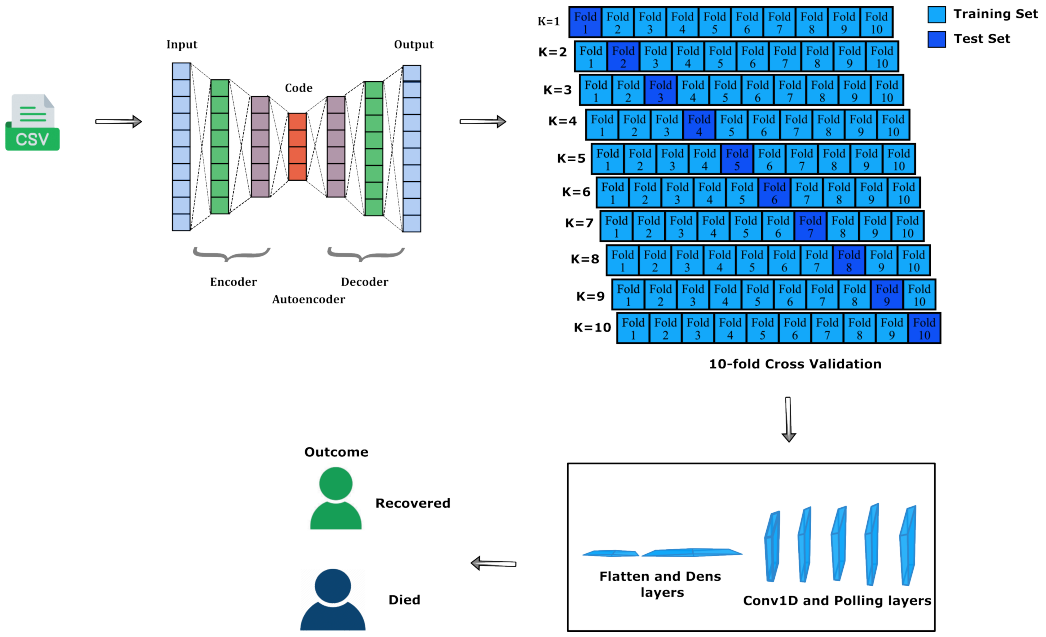

Figure 10. Conceptual framework of the AE+CV-CNN model.

Layer (type)	Output shape	Params #
input_1 (InputLayer)	(None, 84)	0
dense (Dense)	(None, 128)	10,880
dense_1 (Dense)	(None, 50)	6450
dense_2 (Dense)	(None, 128)	6528
dense_3 (Dense)	(None, 84)	10,836
Total params: 34, 694 Trainable params: 34, 694, 273 Non-trainable params: 0		

Figure 11. Summary of the AE+CV-CNN model.

Algorithm 4 AE+CV-CNN Model Classification

Input Clinical dataset DF: (4710 records), auto-encoder training epochs AN, training epochs N, number of folds K
Output Classification

```
Create AE+CV-CNN model with required parameters
Split DF as DF-Dead, DF-Recovered
Partition DF-Dead into DFD-Train and DFD-Predict
For ae = 1 to AE, perform
  AE.train(DFD-Train)
End for
DFAE = AE.predict(DFD-Predict)
DS = DF+DFAE
Create CNN model with required parameters
Perform train–test split on DS(DTrain, DTest)
Partition DTrain into K subsets(F1,...,FK)
For K = 1 to k do
  Dtrain = DF-Fk
  Dvalid = Fk
End for
For e = 1 to N do
  CNN.train(Dtrain)
  CNN.validate(Dvalid)
Test = AE+CV-CNN.test(DTest)
Classification_output = classify(AE+CV-CNN_model, test)
End procedure AE+CV-CNN_model
```

4.4.5. SMOTE-CV-LSTM Model

This model makes use of dataset 3. In this case, data augmentation is performed using SMOTE. The number of samples belonging to both classes is equal. Thus, there are 50% dead patient records and 50% recovered patient records. Once the data have undergone further preprocessing, they are used as input for the LSTM model. The improved performance was seen and is summarized in the Results section. The conceptual framework of the SMOTE-CV-LSTM model is shown in Figure 12.

4.4.6. SMOTE-CV-CNN Model

This model is similar to the SMOTE-CV-LSTM model. The preprocessing and the dataset used are the same, but the difference is that the data are used as input in the CV-CNN model. In both the SMOTE models, the actual model is the same as the one shown in Figures 4 and 10, except that the input data are first balanced using oversampling and then are used as input for the model. The CV-CNN model already showed good performance, and when combined with SMOTE, it produced a very reliable model. The conceptual framework of the SMOTE-CV-CNN model is shown in Figure 13.

In the case of these two models, the input used was a balanced dataset generated by oversampling. The imblearn library was used to import SMOTE. The data were classified into X and Y (independent variables and target variables). Using SMOTE with random state 45, we generated 2415 synthetic samples of the minority class. The samples generated proved to be more valid than the ones generated using other data-augmentation techniques. This dataset was then used as input in the CV-CNN and CV-LSTM models. The pseudo-code of the complete implementation is given in Algorithms 5 and 6. The model summary is the same as the ones given in Figures 5 and 7.

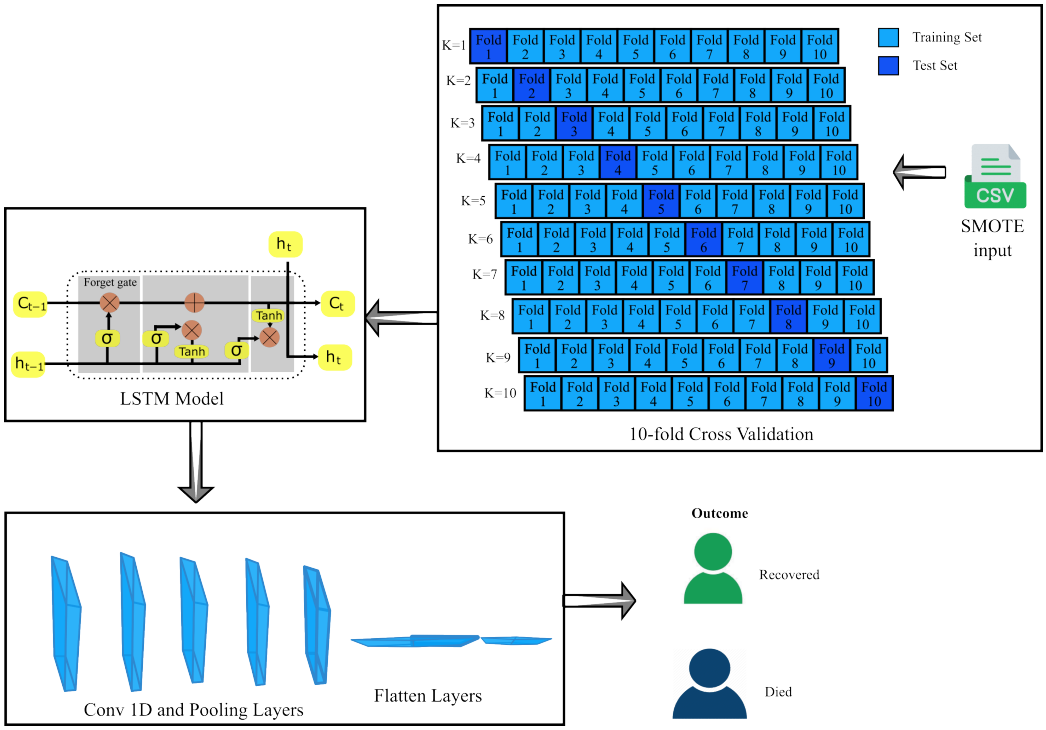

Figure 12. Conceptual framework of the SMOTE-CV-LSTM model.

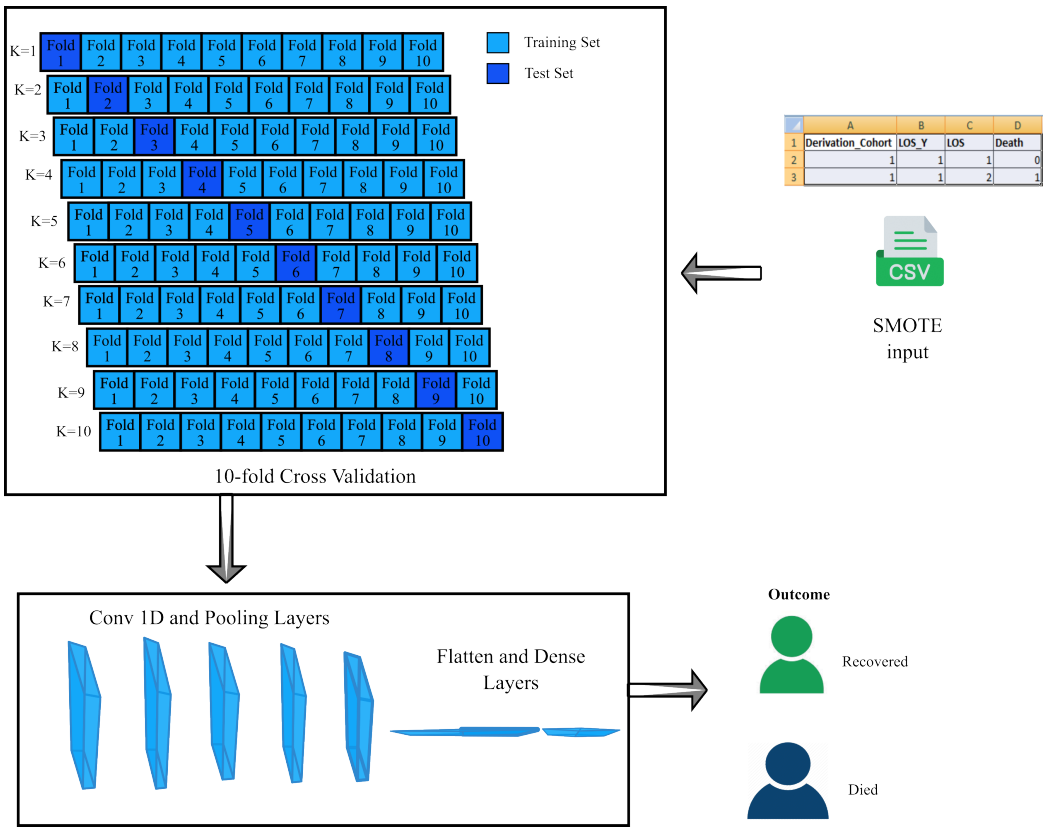

Figure 13. Conceptual framework of the SMOTE-CV-CNN model.

Algorithm 5 A SMOTE-CV-LSTM Model Classification

Input SMOTE generated dataset DF: (7124 records), training epochs N, number of folds K
Output Classification

```
Create LSTM+CNN model with required parameters
Perform Train–Test Split on DF (DTrain, DTest)
Partition DTrain into K subsets(F1,...,FK)
For K = 1 to k do
  Dtrain = DF-Fk
  Dvalid = Fk
End for
For e = 1 to N do
  LSTM+CNN.train(Dtrain)
  LSTM+CNN.validate(Dvalid)
Test = SMOTE-CV-LSTM.test(DTest)
Classification_output = classify(SMOTE-CV-LSTM_model, test)
End procedure SMOTE-CV-LSTM_model
```

Algorithm 6 A SMOTE-CV-CNN Model Classification

Input SMOTE dataset DF: (7126 records), training epochs N, number of folds K
Output Classification

```
Create LSTM+CNN model with required parameters
Perform Train–Test Split on DF (DTrain, DTest)
Partition DTrain into K subsets(F1,...,FK)
K fold cross validation
For K = 1 to k do
  Dtrain = DF-Fk
  Dvalid = Fk
End for
For e = 1 to N do
  CNN.train(Dtrain)
  CNN.validate(Dvalid)
Test = SMOTE-CV-CNN.test(DTest)
Classification_output = classify(SMOTE-CV-CNN_model, test)
End procedure SMOTE-CV-CNN_model
```

5. Experimental Results and Discussions

The experimental results obtained on the standard dataset and the performance analysis are presented in this section.

5.1. Performance Evaluation Metrics

Accuracy, precision, recall, and F1-score are used for performance evaluation. These metrics are defined as follows:

Accuracy: The number of examples correctly predicted from the total number of examples. It is defined in Equation (1)

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{1}$$

Precision: Represents the number of actual samples and is predicted as positive from the total number of samples predicted as positive. It is given in Equation (2)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall: The number of actual samples and is predicted as positive from the total number of samples that are actually positive. It is presented in Equation (3)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

F1-Score: Harmonic mean of precision, and it is defined in Equation (4)

$$\text{F1-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{4}$$

where TP, TN, FP, and FN indicate true positives, true negatives, false positives, and false negatives, respectively.

5.2. Key Information from Preprocessing

During the preprocessing stage, there were several observations from the dataset that were relevant. We were able to identify biological markers that are vital in predicting the survival chance of a patient. Some of these are given below.

From these data, one can infer that severity, age, and MAP are a few of the many important biomarkers that can help predict COVID mortality. In Figure 14, it is noted that the number of deaths is greater than the number of recoveries with higher severity scores. However, severity is an assigned feature and not a laboratory value. Figure 15 shows an age distribution of deaths and recoveries. Figure 16 shows the percentage of death by age. The number of deaths is greater as the age increases. Thus, age is an important biomarker. Figure 17 shows the percentage of deaths by MAP. The number of deaths is greater when MAP < 70.

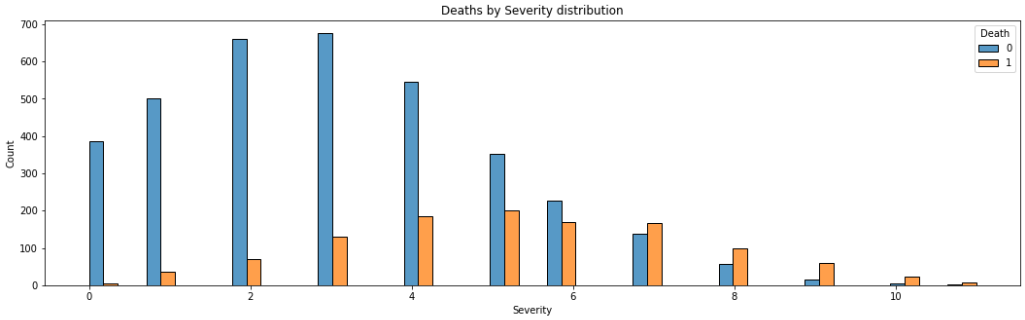

Figure 14. Deaths by severity distribution.

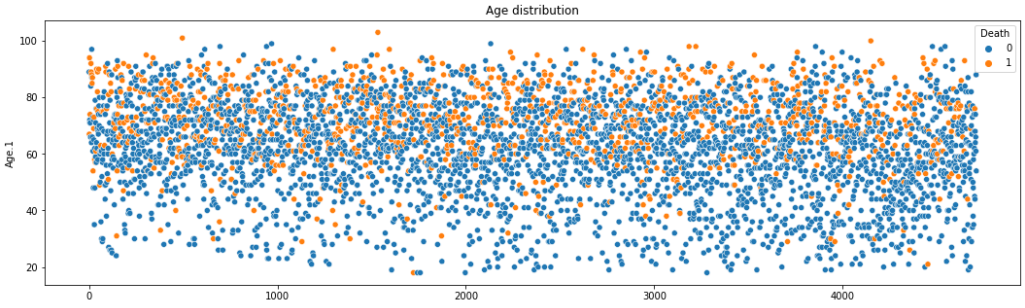

Figure 15. Age distribution of deaths and recoveries.

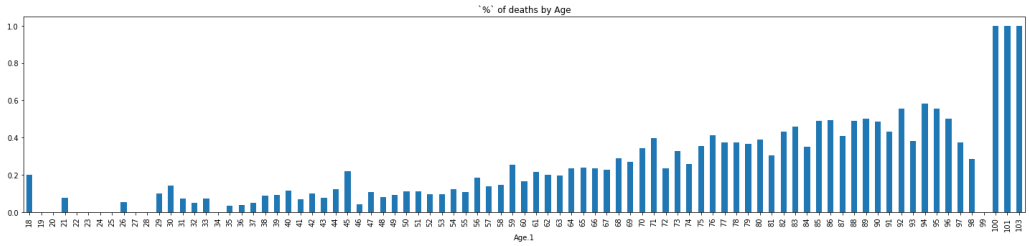

Figure 16. Percentage of death by age.

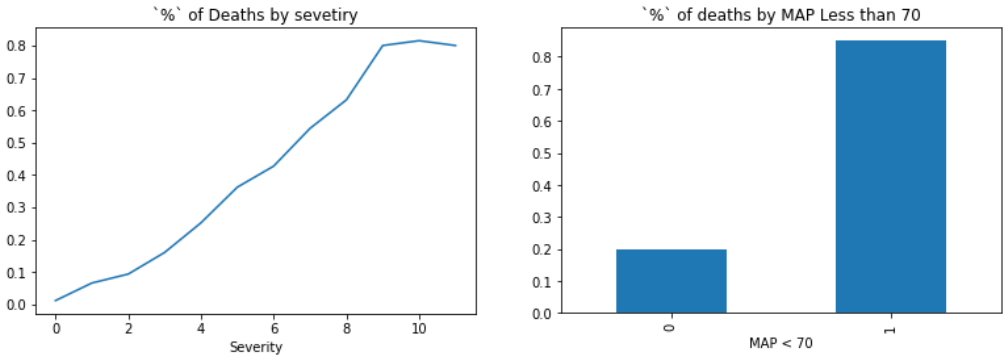

Figure 17. Percentage of death by severity (**left**), MAP < 70 (**right**).

5.3. Experimental Result of Models

The results from the implementations of the proposed models are given in the following sections.

5.3.1. Results of CV-CNN Model

The CV-CNN model produced 96.7% accuracy on the test data. In Figure 18, the confusion matrix of the CV-CNN model is presented. The CV-CNN model classification report values of precision, recall, and F1-score are given in Table 2. The AUC score for the CV-CNN model was 93%. In Table 3, the accuracy of the CV-CNN model for 10-fold cross-validation is given. Figure 19 represents the precision–recall curve of the CV-CNN model.

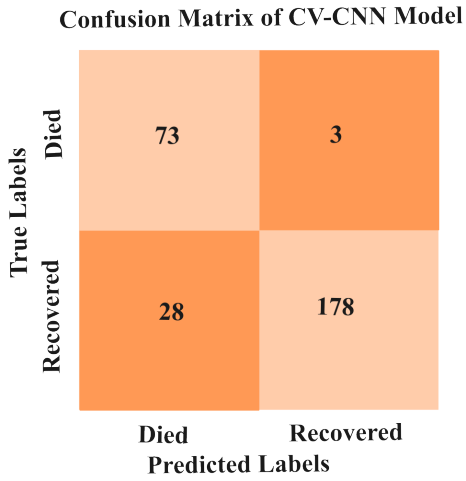

Figure 18. Confusion matrix of the CV-CNN model.

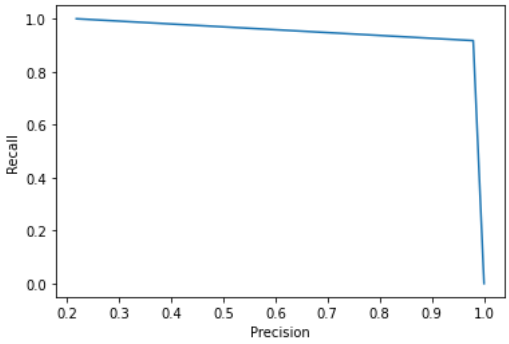

Figure 19. Precision–recall curve of the CV-CNN model.

Table 2. Classification report of the CV-CNN model.

Classification Report				
	Precision (%)	**Recall (%)**	**F1-Score (%)**	**Support**
Recovered	96.12	100.00	98.23	736
Died	98.86	86.45	92.87	206
Accuracy		97.20		942
Macro avg	97.76	93.36	95.32	942
Weighted avg	97.83	97.27	97.83	942

Table 3. Results of the CV-CNN model for 10-fold cross-validation.

Fold	**Accuracy (%)**	**Loss**	**Validation Accuracy (%)**
1	81.79	0.40	82.16
2	83.49	0.37	81.52
3	84.34	0.35	85.77
4	86.22	0.32	85.13
5	88.09	0.27	87.26
6	90.40	0.23	86.62
7	92.24	0.19	92.24
8	94.20	0.15	90.44
9	95.99	0.10	89.38
10	97.24	0.07	88.74

5.3.2. Results of CV-LSTM Model

The CV-LSTM model produced 85.1% accuracy on the test data. The confusion matrix of the CV-LSTM model is given in Figure 20. The precision, recall and F1-score values are shown in the classification report in Table 4. The CV-LSTM model earned an AUC score of 70.0%. Table 5 shows the results of the CV-LSTM model for 10-fold cross-validation. Figure 21 shows the precision–recall curve of the CV-LSTM model.

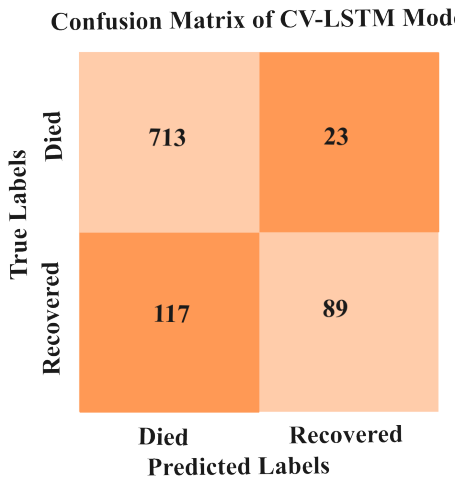

Figure 20. Confusion matrix of the CV-LSTM model.

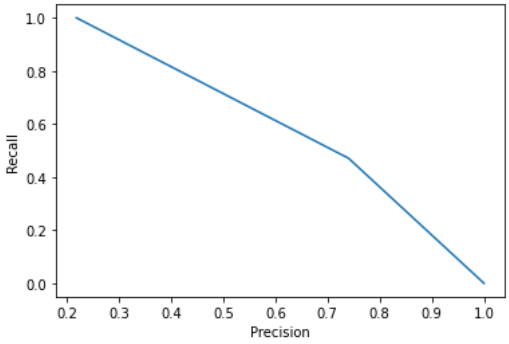

Figure 21. Precision–recall curve of the CV-LSTM model.

Table 4. Classification report of the CV-LSTM model.

Classification Report				
	Precision (%)	**Recall (%)**	**F1-Score (%)**	**Support (%)**
Recovered	86.00	97.00	91.21	736
Died	79.34	43.97	56.65	206
Accuracy		85.00		942
Macro avg	83.74	70.36	74.48	942
Weighted avg	84.71	85.62	83.82	942

Table 5. Results of the CV-LSTM model for 10-fold cross-validation.

Fold	Accuracy (%)	Loss	Validation Accuracy (%)
1	77.07	0.46	79.61
2	78.37	0.45	73.88
3	78.89	0.43	77.70
4	79.64	0.43	79.83
5	81.95	0.41	84.07
6	82.66	0.40	83.86

Table 5. *Cont.*

Fold	Accuracy (%)	Loss	Validation Accuracy (%)
7	83.09	0.40	86.19
8	82.92	0.39	84.92
9	84.24	0.37	77.28
10	84.64	0.36	77.91

5.3.3. Results of IMG-CNN Model

The IMG-CNN model produced a training accuracy of 95.7%. The accuracy of the test data was around 75%. The confusion matrix of the IMG-CNN model is given in Figure 22. The different evaluation metrics for the IMG-CNN model are shown in Table 6.

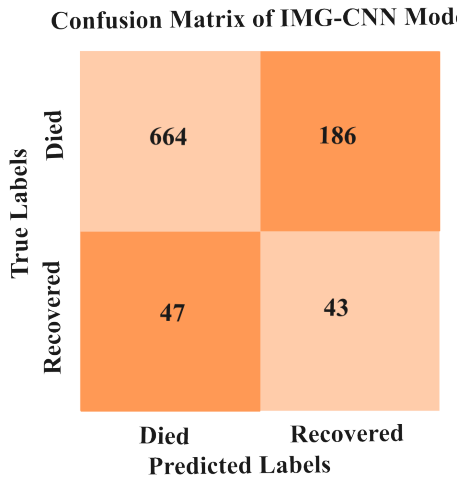

Figure 22. Confusion matrix of the IMG-CNN model.

Table 6. Results of the IMG-CNN model for different evaluation metrics.

	TP	FP	TN	FN	Loss	Accuracy (%)	Precision (%)	Recall (%)	AUC (%)
IMG-CNN Model	2540	244	629	199	0.26	87.74	91.24	92.73	93.95
	val_TP	**val_FP**	**val_TN**	**val_FN**	**val_Loss**	**val_Accuracy**	**val_Precision**	**val_Recall**	**val_AUC**
	664	186	43	47	1.05	75.21	78.12	93.39	64.16

5.3.4. Result of AE+CV-CNN Model

The AE+CV-CNN model produced 97.9% accuracy on the test data. Table 7 shows the classification report of the AE+CV-CNN model. Figure 23 represents the confusion matrix of the AE+CV-CNN model. The AUC score of the AE+CV-CNN model is 97.2%. Table 8 shows the result of the AE+CV-CNN model for 10-fold cross-validation. Figure 24 shows the precision–recall curve for the AE+CV-CNN model.

Table 7. Classification report for the AE+CV-CNN model.

Classification Report				
	Precision (%)	**Recall (%)**	**F1-Score (%)**	**Support (%)**
Recovered	97.00	100	98.23	730
Died	99.00	95.95	97.30	396
Accuracy		98.00		1126
Macro avg	98.04	97.82	98.01	1126
Weighted avg	98.07	98.97	98.99	1126

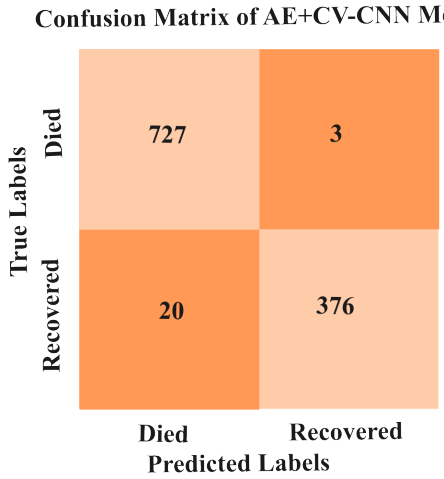

Figure 23. Confusion matrix of the AE+CV-CNN model.

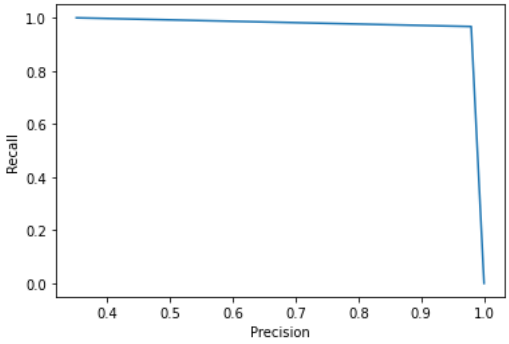

Figure 24. Precision–recall curve of the AE+CV-CNN model.

Table 8. Results of the AE+CV-CNN model for 10-fold cross-validation.

Fold	Accuracy (%)	Loss	Validation Accuracy (%)
1	85.04	0.33	80.81
2	86.24	0.31	83.12
3	87.57	0.29	84.54
4	88.65	0.26	88.27
5	90.19	0.22	84.54
6	92.15	0.19	87.38
7	94.10	0.14	89.52
8	96.03	0.10	85.79
9	96.33	0.09	95.20
10	96.63	0.08	100

5.3.5. Results of SMOTE-CV-LSTM and SMOTE-CV-CNN Models

On using SMOTE to generate artificial samples, the results of SMOTE-based models are shown in Figures 25–28. The validity of the generated samples proved to be better than the samples generated by the AE model. The same analysis performed on the original dataset is performed on the SMOTE-generated dataset, and the results remain true to the nature of the data.

SMOTE-CV-LSTM: The SMOTE-CV-LSTM model produced 86.74% accuracy on the test data. Table 9 shows the classification report of the SMOTE-CV-LSTM model for metrics precision, recall, and F1-score. Figure 29 represents the confusion matrix of the SMOTE-CV-LSTM model. The AUC score for the SMOTE-CV-LSTM model is 86.61%. Figure 30 shows the precision–recall curve for the SMOTE-CV-LSTM model. Table 10 shows the results of SMOTE-CV-LSTM model for 10-fold cross-validation.

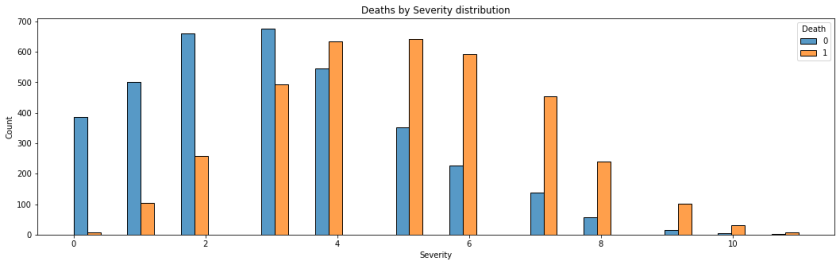

Figure 25. SMOTE data deaths by severity distribution.

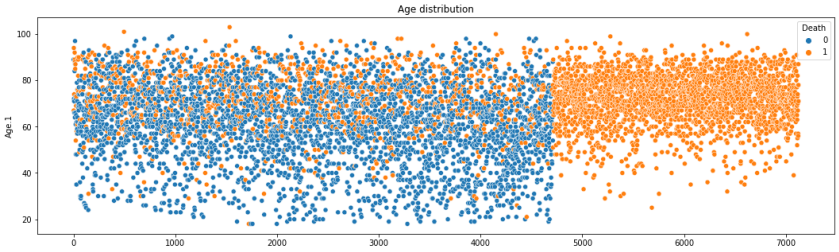

Figure 26. SMOTE data age distribution.

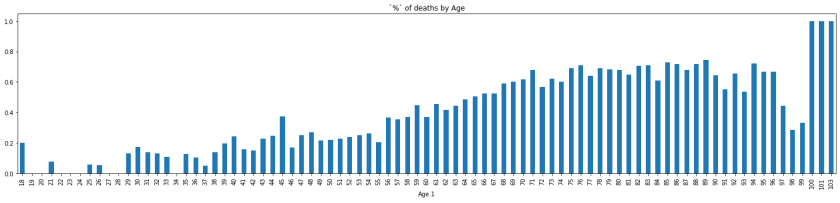

Figure 27. SMOTE data percentage of deaths by age.

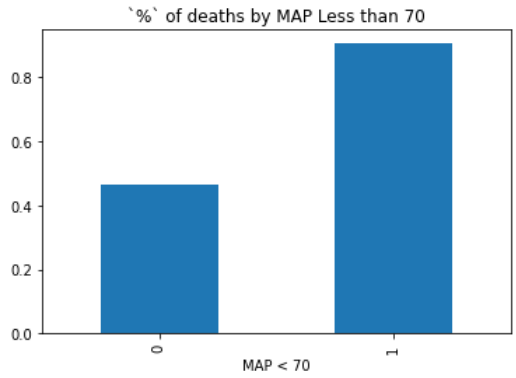

Figure 28. SMOTE data deaths by MAP < 70.

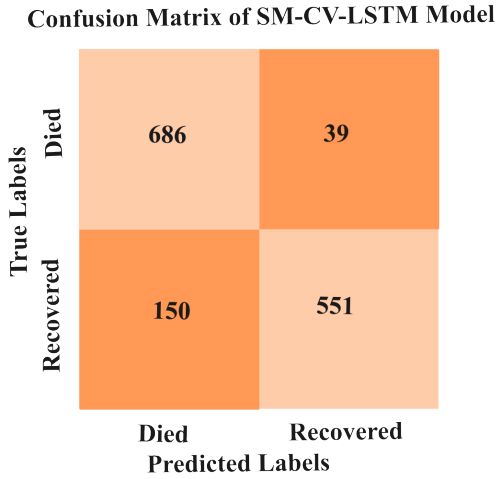

Figure 29. Confusion matrix of the SMOTE-CV-LSTM model.

Table 9. Classification report of SMOTE-CV-LSTM.

Classification Report				
	Precision (%)	**Recall (%)**	**F1-Score (%)**	**Support (%)**
Recovered	82.00	95.21	88.08	725
Died	93.09	79.06	85.98	701
Accuracy		87.09		1426
Macro avg	88.23	87.91	87.05	1426
Weighted avg	88.91	87.32	87.28	1426

Table 10. Results of the SMOTE-CV-LSTM model for 10-fold cross-validation.

Fold	Accuracy (%)	Loss	Validation Accuracy (%)
1	74.10	0.51	65.49
2	78.09	0.46	76.29
3	79.68	0.44	76.85
4	80.66	0.42	78.54
5	81.29	0.41	82.46
6	85.46	0.39	75.17
7	83.12	0.38	78.51
8	82.63	0.39	84.41
9	82.74	0.39	93.11
10	83.35	0.38	89.32

SMOTE-CV-CNN: The SMOTE-CV-CNN model produced 98.10% accuracy on the test data. The classification report of the SMOTE-CV-CNN model is presented in Table 11. Figure 31 represents the confusion matrix of the SMOTE-CV-CNN model. The SMOTE-CV-CNN model obtained an AUC score of 98.11%. Figure 32 shows the precision–recall curve for the SMOTE-CV-CNN model. Table 12 shows the results of the SMOTE-CV-CNN model for 10-fold cross-validation.

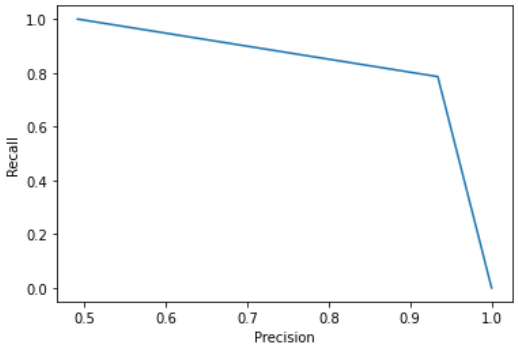

Figure 30. Precision recall curve of the SMOTE-CV-LSTM model.

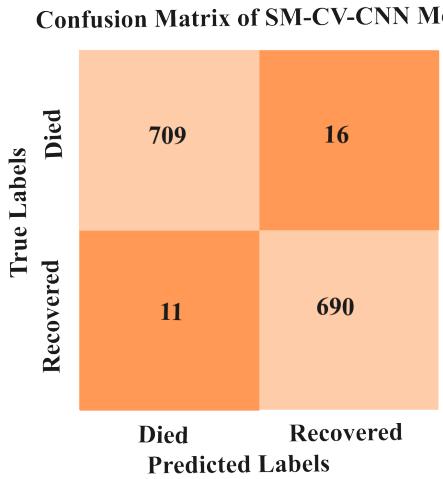

Figure 31. Precision–recall curve of the SMOTE-CV-CNN model.

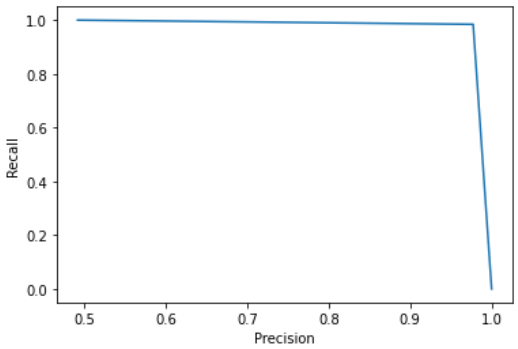

Figure 32. Confusion matrix of the SMOTE-CV-CNN model.

Table 11. Classification report of the SMOTE-CV-CNN model.

	Classification Report			
	Precision (%)	**Recall (%)**	**F1-Score (%)**	**Support (%)**
Recovered	98.00	98.08	98.15	725
Died	98.86	98.25	98.81	701
Accuracy		98.00		1426
Macro avg	98.34	98.08	98.08	1426
Weighted avg	98.00	98.07	98.91	1426

Table 12. Results of the SMOTE-CV-CNN model for 10-fold cross-validation.

Fold	Accuracy (%)	Loss	Validation Accuracy (%)
1	81.96	0.38	79.38
2	85.30	0.33	80.08
3	87.51	0.28	77.27
4	89.30	0.24	86.67
5	92.05	0.19	89.34
6	94.09	0.14	82.88
7	95.95	0.10	89.32
8	96.65	0.08	98.03
9	97.83	0.06	98.03
10	98.49	0.04	99.29

5.4. Inference

The Results section presents the outcome of the execution of the proposed models. A summary of all six model results is given in Table 13. The SMOTE-CV-CNN model produced the best performance after considering all the metrics. Therefore, it would produce reliable predictions in real-life scenarios. Table 14 compares the proposed SMOTE-CV-CNN model with the machine learning techniques proposed by other authors. Table 15 compares the proposed SMOTE-CV-CNN model with other deep learning techniques. In addition, Table 16 shows that the proposed SMOTE-CV-CNN model is compared with existing methods using accuracy metrics. Further, to interpret the resultant values easily, a graphical representation is given in this paper. Figure 33 displays a graphical comparison of all six model results. Similarly, Figures 34 and 35 show the graphical comparison of the proposed SMOTE-CV-CNN model with the existing ML and DL techniques.

Table 13. Results summary of the six proposed models.

Models	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
CV-CNN	97.32	93.76	95.34	96.76
CV-LSTM+CNN	82.56	70.43	73.33	85.18
IMG-CNN	78.32	93.43	85.67	75.22
AE+CV-CNN	98.54	97.39	97.20	97.90
SMOTE-CV-LSTM	88.22	87.47	87.91	86.74
SMOTE-CV-CNN	98.22	98.33	98.43	98.10

Table 14. Comparative analysis of the proposed SMOTE-CV-CNN model with existing ML techniques.

Study	Models	Accuracy (%)	AUC (%)
Pourhomayoun and Shakibi [2]	NN	89.98	93.76
	KNN	89.83	90.97
	SVM	89.02	88.18
	RF	87.93	94.34
	LR	87.91	93.98
	DT	86.87	93.97
Singh et al. [27]	SVM	95.70	95.80
Yoo et al. [28]	DT	98.00	98.00
Shi et al. [29]	LR	82.70	89.00
Proposed	SMOTE-CV-CNN	98.10	98.11

Table 15. Comparative analysis of the proposed SMOTE-CV-CNN model with existing deep learning techniques.

Study	Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
Khozeimeh et al. [10]	ACO-CNN	92.32	95.64	97.33	96.24	62.51
	ABC-CNN	93.10	94.57	97.43	96.01	53.33
	GA-CNN	92.18	94.78	97.83	96.01	57.25
	EHO-CNN	91.87	94.11	98.02	95.97	53.22
	PSO-CNN	91.85	95.03	97.85	95.51	61.59
	BOA-CNN	91.37	94.10	97.01	95.15	53.52
Proposed	SMOTE-CV-CNN	98.10	98.03	98.02	98.00	98.12

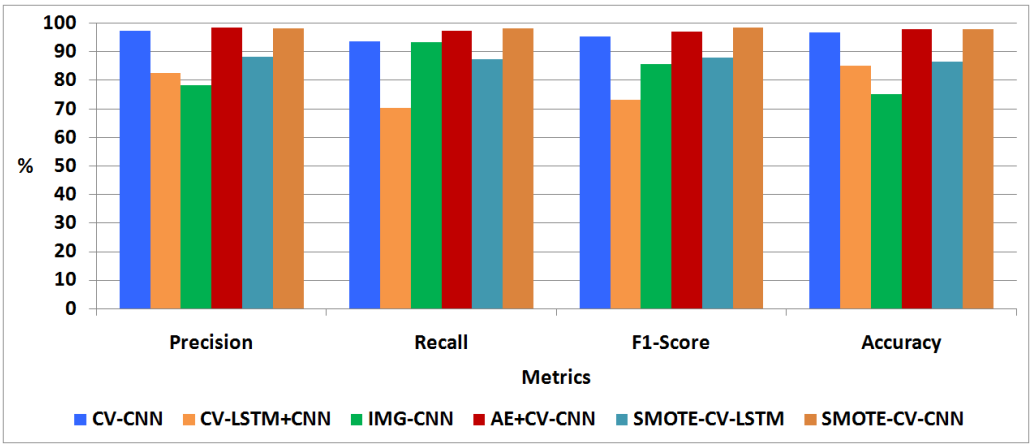

Figure 33. Graphical comparison of all six proposed models.

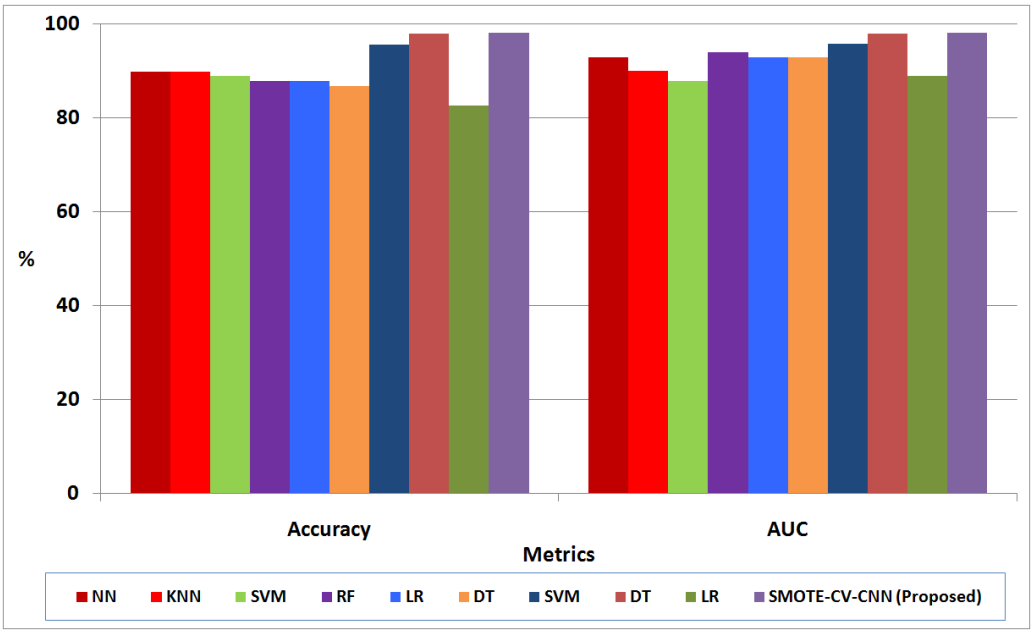

Figure 34. Graphical comparison of proposed SMOTE-CV-CNN model with the existing machine learning techniques using different metrics.

The results of our research revealed several key inferences. It is to be noted that the performance of the LSTM model was not as good as the CNNs. This shows that LSTMs are not suitable for the kind of predictive analytics that the problem scenario requires. Similarly, the IMG-CNN model was also unable to produce promising results. The CV-CNN model produced a considerably good performance. The CNN-AE model is a modification of the CV-CNN model with the addition of a balanced dataset. From the classification report of the LSTM model, it is noted that one of the reasons for low performance is due to the fact

that the data are not balanced. The metrics for the death records are not as good as the recovered records. Therefore, on using SMOTE data in the SMOTE-CV-LSTM model, the results are more balanced, and its performance is better than the CV-LSTM model. Like the CNN-AE model, the SMOTE-CV-CNN is also a modification of the CV-CNN model. The SMOTE-CV-CNN showed better performance than the CNN-AE model, and its results are also more reliable, as the data used are of better quality. Therefore, the SMOTE-CV-CNN model would produce good results for industry applications. It also has better performance than other machine learning techniques and most deep learning techniques. Therefore, it is a very efficient tool for predictive analytics.

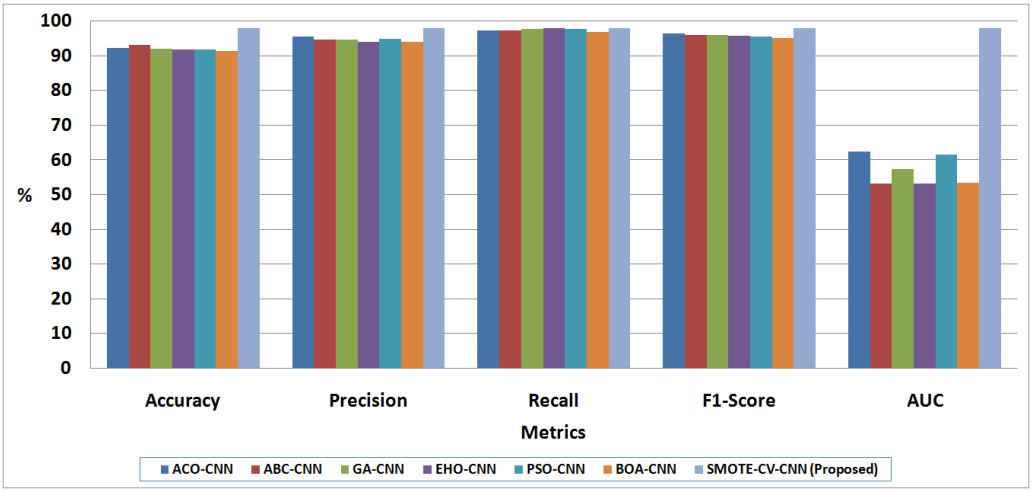

Figure 35. Graphical comparison of the proposed SMOTE-CV-CNN model with the existing deep learning techniques using different metrics.

Table 16. Comparison of the proposed SMOTE-CV-CNN model with existing methods using accuracy metrics.

Paper	Method	Accuracy (%)
Abbas et al. [30]	CNN	93.00
Che Azemin et al. [31]	CNN	71.90
Soda et al. [32]	Deep multimodal CNN	76.80
Varshni et al. [33]	CNN Models along with DenseNet-169 and SVM	80.02
Rahmat et al. [34]	Fully connected RCNN	62.00
Rahman et al. [35]	Different Pre-trained CNN	62.00
Proposed	SMOTE-CV-CNN	98.10

6. Conclusions

In this paper, an efficient COVID-19 mortality risk prediction system using DSMOTE and CNN were developed. Six different models were implemented for the automatic diagnosis of COVID-19, namely CV-CNN, CV-LSTM+CNN, IMG-CNN, AE+CV-CNN, SMOTE-CV-LSTM, and SMOTE-CV-CNN. Each model was trained and tested using the same clinical dataset to identify the recovered and death cases. The performances of these deep learning models were enhanced using various data augmentation, auto-encoder, and oversampling techniques. In the experiment, the CV-CNN model obtained an accuracy of 96.70%. Similarly, CV-LSTM+CNN, IMG-CNN, AE+CV-CNN, SMOTE-CV-LSTM, and SMOTE-CV-CNN attained accuracies of 85.10%, 75.22%, 97.90%, 86.74% and 98.10%, respectively. From the results, it was observed that the SMOTE-CV-CNN model outperformed the other proposed models. In addition, the proposed SMOTE-CV-CNN model was compared with the existing ML and DL models in terms of accuracy, AUC, precision, recall, and F1

score. The comparative analysis showed that the proposed SMOTE-CV-CNN model gives better accuracy than the existing models. Furthermore, an efficient COVID-19 mortality risk prediction system can be applied for the detection of other pneumonia diseases. Further, this work enlisted three main challenges in the current situation as:

- Available small-scale datasets restrain the detailed study by researchers.
- No dataset is available to provide the critical level of patients.
- Even though the proposed model earned high accuracy on the small dataset, it is not clear how this will perform on a large dataset.

Therefore, in order to effectively contribute to the healthcare industry in the future, mutations of COVID-19 should be observed, and the proposed framework could be updated accordingly.

Author Contributions: Conceptualization, R.S.; writing—original draft preparation, A.M. and R.S.; writing—review and editing, Y.V.S.M., R.D.M.S., S.T. and M.A.; supervision, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The data collection was authorized by the Montefiore Medical Center/Albert Einstein College of Medicine Institutional Review Board, which approved the waiver of patient-informed consent due to the retrospective design of the study.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this work come mainly from public dataset https://www.kaggle.com/datasets/harshwalia/mortality-risk-clinincal-data-of-covid19-patients (accessed on 13 February 2023), and they are anonymized. More details about the data are available under Section 4.1. The code of the proposed deep learning architecture will be made openly accessible upon publication. https://github.com/rajssnbeme/COVID19prediction (accessed on 13 February 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Asgharnezhad, H.; Shamsi, A.; Alizadehsani, R.; Khosravi, A.; Nahavandi, S.; Sani, Z.A.; Srinivasan, D.; Islam, S.M.S. Objective evaluation of deep uncertainty predictions for COVID-19 detection. *Sci. Rep.* **2022**, *12*, 815. [CrossRef]
2. Pourhomayoun, M.; Shakibi, M. Predicting mortality risk in patients with COVID-19 using machine learning to help medical decision-making. *Smart Health* **2021**, *20*, 100178. [CrossRef] [PubMed]
3. Ibrahim, D.M.; Elshennawy, N.M.; Sarhan, A.M. Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. *Comput. Biol. Med.* **2021**, *132*, 104348. [CrossRef] [PubMed]
4. Wu, J. Introduction to convolutional neural networks. *Natl. Key Lab Nov. Softw. Technol.* **2017**, *5*, 495.
5. Tekerek, A. A novel architecture for web-based attack detection using convolutional neural network. *Comput. Secur.* **2021**, *100*, 102096. [CrossRef]
6. Xu, B.; Gutierrez, B.; Mekaru, S.; Sewalk, K.; Goodwin, L.; Loskill, A.; Cohn, E.L.; Hswen, Y.; Hill, S.C.; Cobo, M.M.; et al. Epidemiological data from the COVID-19 outbreak, real-time case information. *Sci. Data* **2020**, *7*, 106. [CrossRef]
7. Banoei, M.M.; Dinparastisaleh, R.; Zadeh, A.V.; Mirsaeidi, M. Machine-learning-based COVID-19 mortality prediction model and identification of patients at low and high risk of dying. *Crit. Care* **2021**, *25*, 1–14. [CrossRef]
8. Bikku, T. Multi-layered deep learning perceptron approach for health risk prediction. *J. Big Data* **2020**, *7*, 1–14. [CrossRef]
9. Yan, L.; Zhang, H.-T.; Goncalves, J.; Xiao, Y.; Wang, M.; Guo, Y.; Sun, C.; Tang, X.; Jing, L.; Zhang, M.; et al. An interpretable mortality prediction model for COVID-19 patients. *Nat. Mach. Intell.* **2020**, *2*, 283–288. [CrossRef]
10. Khozeimeh, F.; Sharifrazi, D.; Izadi, N.H.; Joloudari, J.H.; Shoeibi, A.; Alizadehsani, R.; Gorriz, J.M.; Hussain, S.; Sani, Z.A.; Moosaei, H.; et al. Combining a convolutional neural network with auto-encoders to predict the survival chance of COVID-19 patients. *Sci. Rep.* **2021**, *11*, 15343. [CrossRef]
11. Khan, I.U.; Aslam, N.; Aljabri, M.; Aljameel, S.S.; Kamaleldin, M.M.A.; Alshamrani, F.M.; Chrouf, S.M.B. Computational intelligence-based model for mortality rate prediction in COVID-19 patients. *Int. J. Environ. Res. Public Health* **2021**, *18*, 6429. [CrossRef] [PubMed]
12. Tezza, F.; Lorenzoni, G.; Azzolina, D.; Barbar, S.; Leone, L.A.C.; Gregori, D. Predicting in-hospital mortality of patients with COVID-19 using machine learning techniques. *J. Pers. Med.* **2021**, *11*, 343. [CrossRef] [PubMed]
13. Wang, K.; Zuo, P.; Liu, Y.; Zhang, M.; Zhao, X.; Xie, S.; Zhang, H.; Chen, X.; Liu, C. Clinical and laboratory predictors of in-hospital mortality in patients with coronavirus disease-2019: A cohort study in wuhan, china. *Clin. Infect. Dis.* **2020**, *71*, 2079–2088. [CrossRef] [PubMed]

14. Booth, A.L.; Abels, E.; McCaffrey, P. Development of a prognostic model for mortality in covid 19 infection using machine learning. *Mod. Pathol.* **2021**, *34*, 522–531. [CrossRef]
15. Sun, L.; Song, F.; Shi, N.; Liu, F.; Li, S.; Li, P.; Zhang, W.; Jiang, X.; Zhang, Y.; Sun, L.; et al. Combination of four clinical indicators predicts the severe/critical symptom of patients infected COVID-19. *J. Clin. Virol.* **2020**, *128*, 104431. [CrossRef] [PubMed]
16. Rechtman, E.; Curtin, P.; Navarro, E.; Nirenberg, S.; Horton, M.K. Vital signs assessed in initial clinical encounters predict COVID-19 mortality in an NYC hospital system. *Sci. Rep.* **2020**, *10*, 21545. [CrossRef]
17. Hu, C.; Liu, Z.; Jiang, Y.; Shi, O.; Zhang, X.; Xu, K.; Suo, C.; Wang, Q.; Song, Y.; Yu, K.; et al. Early prediction of mortality risk among patients with severe COVID-19, using machine learning. *Int. J. Epidemiol.* **2020**, *49*, 1918–1929. [CrossRef]
18. Zhou, Y.; Yang, Z.; Guo, Y.; Geng, S.; Gao, S.; Ye, S.; Hu, Y.; Wang, Y. A new predictor of disease severity in patients with COVID-19 in Wuhan, China. *medRxiv* **2020**. [CrossRef]
19. Li, Y.; Horowitz, M.A.; Liu, J.; Chew, A.; Lan, H.; Liu, Q.; Sha, D.; Yang, C. Individual-level fatality prediction of COVID-19 patients using AI methods. *Front. Public Health* **2020**, *8*, 587937. [CrossRef]
20. Zhu, Z.; Cai, T.; Fan, L.; Lou, K.; Hua, X.; Huang, Z.; Gao, G. Clinical value of immune inflammatory parameters to assess the severity of coronavirus disease 2019. *Int. J. Infect. Dis.* **2020**, *95*, 332–339. [CrossRef]
21. Monjur, O.; Preo, R.B.; Shams, A.B.; Raihan, M.; Sarker, M.; Fairoz, F. COVID-19 Prognosis and Mortality Risk Predictions from Symptoms: A Cloud-Based Smartphone Application. *bioMed* **2021**, *1*, 114–125. [CrossRef]
22. Elshennawy, N.M.; Ibrahim, D.M.; Sarhan, A.M.; Arafa, M. Deep-Risk: Deep Learning-Based Mortality Risk Predictive Models for COVID-19. *Diagnostics* **2022**, *12*, 1847. [CrossRef] [PubMed]
23. Davenport, T.; Kalakota, R. The potential for artificial intelligence in healthcare. *Future Healthc. J.* **2019**, *6*, 94. [CrossRef] [PubMed]
24. Altschul, D.J.; Unda, S.R.; Benton, J.; de la Garza Ramos, R.; Cezayirli, P.; Mehler, M.; Eskandar, E.N. A novel severity score to predictinpatient mortality in COVID-19 patients. *Sci. Rep.* **2020**, *10*, 16726. [CrossRef] [PubMed]
25. Kaggle, Dataset. Available online: https://www.kaggle.com/datasets/harshwalia/mortality-risk-clinincal-data-of-covid19-patients (accessed on 13 February 2023).
26. Joloudari, J.H.; Marefat, A.; Nematollahi, M.A.; Oyelere, S.S.; Hussain, S. Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks. *Appl. Sci.* **2023**, *13*, 4006. [CrossRef]
27. Singh, M.; Bansal, S.; Ahuja, S.; Dubey, R.K.; Panigrahi, B.K.; Dey, N. Transfer learning—Based ensemble support vector machine model for automated COVID-19 detection using lung computerized tomography scan data. *Med. Biol. Eng. Comput.* **2021**, *59*, 825–839. [CrossRef]
28. Yoo, S.H.; Geng, H.; Chiu, T.L.; Yu, S.K.; Cho, D.C.; Heo, J.; Choi, M.S.; Choi, I.H.; Cung Van, C.; Nhung, N.V.; et al. Deep learning-based decision-tree classifier for COVID-19 diagnosis from chest X-ray imaging. *Front. Med.* **2020**, *7*, 427. [CrossRef]
29. Shi, W.; Peng, X.; Liu, T.; Cheng, Z.; Lu, H.; Yang, S.; Zhang, J.; Wang, M.; Gao, Y.; Shi, Y.; et al. A deep learning-based quantitative computed tomography model for predicting the severity of COVID-19: A retrospective study of 196 patients. *Ann. Transl. Med.* **2021**, *9*, 216. [CrossRef]
30. Abbas, A.; Abdelsamea, M.M.; Gaber, M.M. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *Appl. Intell.* **2021**, *51*, 854–864. [CrossRef]
31. Che Azemin, M.Z.; Hassan, R.; Mohd Tamrin, M.I.; Md Ali, M.A. COVID-19 deep learning prediction model using publicly available radiologist-adjudicated chest X-ray images as training data: Preliminary findings. *Int. J. Biomed. Imaging* **2020**, *2020*, 8828855. [CrossRef]
32. Soda, P.; D'Amico, N.C.; Tessadori, J.; Valbusa, G.; Guarrasi, V.; Bortolotto, C.; Akbar, M.U.; Sicilia, R.; Cordelli, E.; Fazzini, D.; et al. AI for COVID: Predicting the clinical outcomes in patients with COVID-19 applying AI to chest-X-rays. An Italian multicentre study. *Med. Image Anal.* **2021**, 74, 102216. [CrossRef] [PubMed]
33. Varshni, D.; Thakral, K.; Agarwal, L.; Nijhawan, R.; Mittal, A. Pneumonia detection using CNN based feature extraction. In Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 20–22 February 2019; IEEE: Coimbatore, India, 2019; pp. 1–7.
34. Rahmat, T.; Ismail, A.; Aliman, S. Chest X-ray image classification using faster R-CNN. *Malays. J. Comput.* **2019**, *4*, 225–236.
35. Rahman, T.; Chowdhury, M.E.; Khandakar, A.; Islam, K.R.; Islam, K.F.; Mahbub, Z.B.; Kadir, M.A.; Kashem, S. Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray. *Appl. Sci.* **2020**, *10*, 3233. [CrossRef]