

CNN-AE: Convolution Neural Network combined with Autoencoder approach to detect survival chance of COVID-19 patients

Fahime Khozeimeh

Deakin University

Danial Sharifrazi

Islamic Azad University

Navid Hoseini Izadi

Isfahan University of Technology

Javad Hassannataj Joloudari

University of Birjand

Afshin Shoeibi

Ferdowsi University of Mashhad

Roohallah Alizadehsani (✉ ralizadehsani@deakin.edu.au)

Deakin University

Juan M. Gorriz

University of Cambridge

Sadiq Hussain

Dibrugarh University

Zahra Alizadeh Sani

Iran University of Medical Sciences

Hossein Moosaei

University of Bojnord

Abbas Khosravi

Deakin University

Saeid Nahavandi

Deakin University

Sheikh Mohammed Shariful Islam

Deakin University

Research Article

Keywords: COVID-19, survival chance, CNN, Autoencoder, data augmentation, feature analysis

Posted Date: April 23rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-442013/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Scientific Reports on July 28th, 2021. See the published version at <https://doi.org/10.1038/s41598-021-93543-8>.

CNN-AE: Convolution Neural Network combined with Autoencoder approach to detect survival chance of COVID-19 patients

Fahime Khozeimeh^a, Danial Sharifrazi^b, Navid Hoseini Izadi^c, Javad Hassannataj Joloudari^d, Afshin Shoeibi^{e,f}, Roohallah Alizadehsani^{g,*}, Juan M. Gorriz^{g,h}, Sadiq Hussainⁱ, Zahra Alizadeh Sani^j, Hossein Moosaei^k, Abbas Khosravi^a, Saeid Nahavandi^a, Sheikh Mohammed Shariful Islam^{l,m,n}

^aInstitute for Intelligent Systems Research and Innovations (IISRI), Deakin University, Geelong, Australia

^bDepartment of Computer Engineering, School of Technical and Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran

^cDepartment of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

^dDepartment of Computer Engineering, Faculty of Engineering, University of Birjand, Birjand, Iran

^eComputer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran

^fFaculty of Electrical and Computer Engineering, Biomedical Data Acquisition Lab, K. N. Toosi University of Technology, Tehran, Iran

^gDepartment of Signal Theory, Networking and Communications, Universidad de Granada

^hDepartment of Psychiatry, University of Cambridge, Cambridge, UK

ⁱSystem Administrator, Dibrugarh University, Assam 786004, India

^jOmid hospital, Iran University of Medical Sciences, Tehran, Iran

^kDepartment of Mathematics, Faculty of Science, University of Bojnord, Iran; moosaei@ub.ac.ir

^lInstitute for Physical Activity and Nutrition, School of Exercise and Nutrition Sciences, Deakin University, Geelong, VIC, 3220, Australia

^mCardiovascular Division, The George Institute for Global Health, Newtown, Australia

ⁿSydney Medical School, University of Sydney, Camperdown, Australia

*Corresponding author: Roohallah Alizadehsani Email: ralizadehsani@deakin.edu.au

Abstract

Today, the use of artificial intelligence methods to diagnose and predict infectious and non-infectious diseases has attracted so much attention. Currently, COVID-19 is considered a new virus which has caused so many deaths worldwide. Due to the pandemic nature of COVID-19, the automated tools for the clinical diagnostic of this disease are highly desired. Convolutional Neural Networks (CNNs) have shown outstanding classification performance on image datasets. Up to our knowledge, COVID computer aided diagnosis systems based on CNNs and clinical information have been never analyzed or explored to date. Moreover, Most of existing literature on COVID-19 focuses on distinguishing infected individuals from non-infected ones. In this paper, we propose a novel method named CNN-AE to predict survival chance of COVID-19 patients using a CNN trained on clinical information. To further increase the prediction accuracy, we use the CNN in combination with an autoencoder. Our method is one of the first that aims to predict survival chance of already infected patients. We rely on clinical data to carry out the prediction. The motivation is that the required resources to prepare CT images are expensive and limited compared to the resources required to collect clinical data such as blood pressure, liver disease, etc.

We evaluate our method on a publicly available clinical dataset of deceased and recovered patients which we have collected. Careful analysis of the dataset properties is also presented which consists of important features extraction and correlation computation between features. Since most of COVID-19 patients are usually recovered, the number of deceased samples of our dataset is low leading to data imbalance. To remedy this issue, a data augmentation procedure based on autoencoders is proposed. To demonstrate the generality of our augmentation method, we train random forest and Naïve Bayes on our dataset with and

without augmentation and compare their performance. We also evaluate our method on another dataset for further generality verification. Experimental results reveal the superiority of CNN-AE method compared to the standard CNN as well as other methods such as random forest and Naïve Bayes. COVID-19 detection average accuracy of CNN-AE is 96.05% which is higher than CNN average accuracy of 92.49%. To show that clinical data can be used as a reliable dataset for COVID-19 survival chance prediction, CNN-AE is compared with a standard CNN which is trained on CT images.

Keywords: COVID-19, survival chance, CNN, Autoencoder, data augmentation, feature analysis

1 Introduction

Nowadays medical centers are bloated with huge amount of data which are collected from patients. Medical biomarkers, demographic data and images modalities can help and support medical specialists to diagnose infectious diseases [1], Alzheimer [2], Parkinson [3], Coronary artery disease [4], etc. However, these data must be processed and analyzed in order to become usable information for the specialists. Automated solutions based on artificial intelligence have the potential to carry out the required process efficiently [5].

Recently, a new type of coronavirus (COVID-19) has emerged which has caused many death worldwide [6-9]. The virus outbreak has been observed for the first time in late 2019 [10, 11]. COVID-19 primarily aims the patients' lungs [12, 13]. Therefore, if the virus is not properly diagnosed in the early stages of infection, it can severely damage the lungs [14]. Although the mortality rate of the virus is low, it must not be overlooked due to being highly contagious. The virus threat becomes more serious when the medical centers' resources cannot provide service to the large number of people who get infected each day [15].

Survival chance prediction of infected people is as important as early detection of the virus. Under resource scarcity, the medical centers can take into account patients' conditions and use the available resources wisely. Existing literature on COVID-19 detection has proven that deep neural networks are very effective for early detection of COVID-19 [16]. Therefore, deep networks may be useful for survival chance prediction as well. In this study, we rely on clinical dataset including gender, age, blood type, etc. to perform diagnostic analysis on COVID-19 virus. To the best of our knowledge, this is the first time that a survival chance predictor is proposed for COVID-19 patients using clinical features. To evaluate the effectiveness of the proposed method, we compare its performance against a standard CNN trained on image data. Our contributions are as follows:

- Survival chance prediction for COVID-19 patients based on clinical features
- Preparing clinical dataset to predict survival chance of COVID-19 patients for the first time
- Presenting careful analysis of the dataset characteristics including impact of features on mortality rate and correlation between each feature pair
- Making our dataset publicly available
- Combining Autoencoder with CNN to increase prediction accuracy
- Proposing a data augmentation procedure to balance number of samples of different classes of dataset. Our data augmentation method is generic and applicable to any other dataset.

The remaining sections of the paper are organized as follows: related literature is reviewed in Section 2. Section 3 describes our dataset. Required background is briefly reviewed in Section 4. The proposed methodology is explained in Section 5. Experimental results are presented in section 6. Discussion, conclusion, and future works are presented in sections 7 and 8, respectively.

2 Literature Review

This paper tries to predict survival chance of COVID-19 patients using clinical features. Therefore, we review COVID-19 detection methods which rely on clinical features. We also review methods on mortality estimation of infected patients.

To contain the COVID-19 threat as soon as possible, the researchers have tackled this virus from multiple directions. Some researchers have focused on fast and accurate detection of infected patients from non-infected ones. As an example, Wu et al. [17] extracted 11 vital blood indices through Random Forest (RF) method to design an assistant discrimination tool. Their method demonstrated an accuracy of 96.97% and 97.95% for the test set and cross-validation set respectively. The assistant tool was well-equipped to perform preliminary investigation of the suspected patients and suggested them quarantine and timely treatment. In another attempt, Rahman et al. [18] reviewed various studies on treatment, complications, seasonality, symptoms, clinical features and epidemiology of COVID-19 infection to assist the general people and medical staff in facilitating necessary guidance of the pandemic. Using a CNN they tried to detect infected patients in order to isolate them from the healthy ones.

In addition to distinguishing infected patients from non-infected ones, it is also important to determine whether infected ones have severe condition or not. Hence, Chen et al. [19] studied 148 severe and 214 non-severe COVID-19 patients from Wuhan, China using their laboratory test results and symptoms as features to devise a RF. The task of the RF was to classify COVID-19 patients to severe and non-severe types using the features. Using laboratory results and symptom as input, their model yielded accuracy over 90%. Some of the key features they identified were LDH, IL-6, absolute neutrophil count, D-Dimer, diabetes, gender, cardiovascular disease, hypertension and age.

Some other researchers focused on mortality risk prediction of the patients. Gao et al. [20] proposed a mortality risk prediction model for COVID-19 (MRPMC) that applied clinical data to stratify patients by mortality risk that predicted mortality 20 days in advance. Their ensemble framework was based on four machine learning techniques including Neural Network (NN), Gradient Boosted Decision Tree [21], Support Vector Machine (SVM) and Logistic Regression. Their model demonstrated accurate and expeditious mortality risk stratification of COVID-19 patients.

Zhu et al. [22] presented a risk stratification score system and a multilayer perceptron (MLP) with six dense layers to predict mortality. 78 clinical variables were identified and prediction performance was compared with pneumonia severity index (PSI), CURB-65 score, and COVID-19 severity score. They derived the top five predictors of mortality- lactate dehydrogenase, C-reactive protein, neutrophil:lymphocyte ratio, O₂ Index and D-dimer. Their model was proved to be effective in resource-constrained and time-sensitive environment.

The power of XGBoost algorithm has also been leveraged for mortality risk prediction. As an example, Yan et al. [23] collected blood samples of 485 infected patients from China to detect key predictive biomarkers of COVID-19 mortality. They employed a XGBoost classifier which was able to predict the mortality of patients with 90% accuracy more than 10 days in advance. In another study, Bertsimas et al. [24] developed a data-driven mortality risk calculator for in-hospital patients. Laboratory, clinical, and demographic variables were accumulated at the time of hospital admission. Again, they applied XGBoost for mortality prediction on patients. Taking a different approach, Abdulaal et al. [25] devised a point-of-admission mortality risk scoring system utilizing a MLP for COVID-19 patients. The network exploited patient specific features including present symptoms, smoking history, comorbidities and demographics

and predicted the mortality risk based on them. The mortality prediction model demonstrated a specificity of 85.94%, sensitivity of 87.50% and accuracy of 86.25%.

Since the symptoms of different viruses may be similar to some extent, there has been an attempt to distinguish multiple viruses from each other in [26]. To this end multiple classical machine learning algorithms were trained to classify textual clinical reports into four classes of SARS, ARDS, COVID-19, and Both (SARS, COVID-19). Feature engineering has also been carried out using report length, Bag of words (BOW) and Term frequency/inverse document frequency (TF/IDF) techniques. Multinomial Naïve Bayes and Logistic regression outperformed other classifiers with a testing accuracy of 96.2%. The summary of the reviewed works are presented in Table 1.

Most of the existing literatures on COVID-19 rely on CT and X-ray images to achieve their objectives. Based on the review presented above, it is apparent that existing works based on clinical data are rather scarce. Hence we were motivated to present yet another study using clinical data for mortality risk assessment. The difference between our method and existing literature on mortality risk assessment is twofold. First, we present a new approach to carry out the assessment. Second, some of the clinical features that we have considered have never been used before which is why we release our dataset publicly. As will be discussed in the rest of the paper, clinical data are cost effective compared to CT images and classifiers trained on clinical data provide good performance which is almost as good as training on CT images. To justify this claim, we compare our method performance (trained on clinical data) with a standard CNN which is trained on CT images.

Table 1. Summary of reviewed literature

Ref	Method	objective
Gao et al. [20]	An Ensemble of NN, grad boosted decision tree, SVM, and logistic regression	mortality risk prediction
Zhu et al. [22]	MLP	mortality risk prediction
Yan et al. [23]	XGBoost classifier	mortality risk prediction
Bertsimas et al. [24]	XGBoost classifier	mortality risk prediction
Abdulaal et al. [25]	MLP	mortality risk prediction
Wu et al. [17]	RF	COVID-19 detection
Rahman et al. [18]	CNN	COVID-19 detection
Khanday et al. [26]	Multinomial Naïve Bayes and Logistic regression	Patients classification to four classes {SARS, ARDS, COVID-19, Both (SARS, COVID-19)}
Chen et al. [19]	RF	COVID-19 severity classification

3 Background

The proposed method consists of two modules which are classifier and data augmenter. The classification is carried out using a CNN. The data augmentation is realized using 10 autoencoders. In this section, we briefly review the main concepts of CNN and autoencoder.

3.1 CNN

CNNs are massively used in image based learning applications. Thanks to automatic feature extraction mechanism of CNNs, they can discover valuable information from training samples. CNNs are usually designed with several convolutional, pooling, and fully-connected layers [27]. As illustrated in Figure 1, the feature extraction is done via convolving the input with convolutional kernels. The pooling

layer reduces the computational volume of the network without making a noticeable change in the resolution of feature map. In CNNs, usually the size of the pooling layers decreases as the number of layers increases. Two of the most popular types of pooling layers are max pooling and average pooling [28].

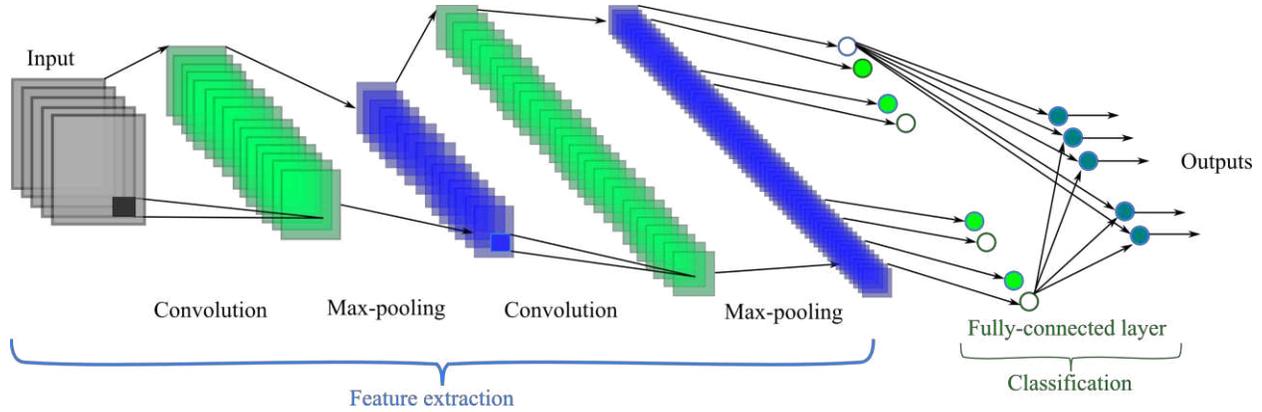


Figure 1. A CNN schematic

3.2 Autoencoders

Autoencoders belong to the realm of unsupervised learning since they do not need labeled data for their training. In a nutshell, an autoencoder compresses input data to a lower dimensional latent space and then reconstructs the data by decompressing the latent space representation. Similar to Principle Component Analysis (PCA), autoencoders perform dimensionality reduction in the compression phase. However, unlike PCA that relies on linear transformation, autoencoders carry out nonlinear transformation using deep neural networks [29]. The architecture of a typical autoencoder is presented in Figure 2.

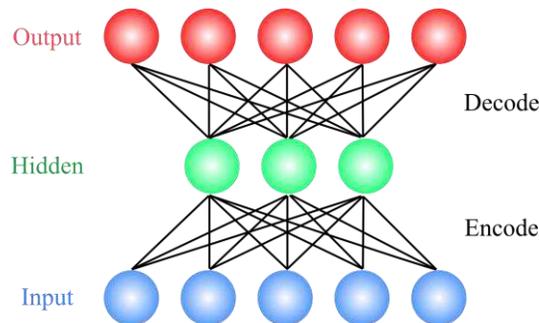


Figure 2. Autoencoder architecture: high dimension input data is encoded (compressed) to form latent (hidden) space which has a lower dimension compared to the original input. The latent representation is reconstructed (decoded) to yield decompressed output.

3.3 Information gain

We review Information gain (IG) since in section 4, it is used to determine the degree to which each feature of our dataset contributes to the patients' death. IG calculates the entropy reduction due to splitting a dataset D based on a given value a of random variable A :

$$IG(D, A = a) = H(D) - H(D|A = a),$$

where $H(D)$ and $H(D|A = a)$ are entropy on dataset D and conditional entropy on D given $A = a$, respectively. The conditional entropy is computed as:

$$H(D|A = a) = \sum_{v \in \text{values}(A)} \frac{|D_{A=a}|}{|D|} H(D_{A=a}), \quad (1)$$

where $D_{A=a} \subset D$ is the set of samples with variable $A = a$. Moreover, $|D_{A=a}|$ and $|D|$ denote the cardinality of subset $D_{A=a}$ and set D , respectively. In equation (1), the sum is computed over all possible values of A .

4 Description of our clinical dataset

The dataset we have collected in this paper contains 320 cases, 300 of which are recovered patients and the remaining 20 cases represent deceased ones. The percentage of female cases is 55%. The mean age of the collected cases is 49.5 years old and standard deviation is 18.5. The patients have referred to Tehran Omid hospital in Iran from 2020-03-03 to 2020-04-21. Ethical approval of these data was also obtained from the Omid hospital. To gather the cases, patients' history collected by doctors, questionnaire filled by patients, laboratory test, vital sign measurement, and imaging investigation were used. The description of the dataset features is presented in Table 2. Our dataset is publicly available in [30]. Institutional approval was acquired for the use of patient datasets used in research studies for diagnostic and therapeutic purposes. Approval was granted on the grounds of existing datasets. Informed consent was obtained from all participants of this study. All methods were carried out in accordance with relevant guidelines and regulations.

Table 2. Description of dataset features which are used for classification

Feature Number	Feature Name	Range
1	Gender	{Male, Female}
2	Age	11-95 years old
3	Blood Type	{A-, A+, B-, B+, AB-, AB+, O-, O+}
4	BCG Vaccine	{Yes, No}
5	CBC	{Normal, Abnormal}
6	Diabetes	{Yes, No}
7	blood pressure	{Yes, No}
8	Asthma	{Yes, No}
9	Heart disease	{Yes, No}
10	kidney disease	{Yes, No}
11	Respiratory disease	{Yes, No}
12	Cancer	{Yes, No}
13	Corticosteroids	{Yes, No}
14	Transplant	{Yes, No}
15	HEM	{Yes, No}
16	Immunodeficiency	{Yes, No}
17	Liver disease	{Yes, No}
18	Rheumatological disease	{Yes, No}
19	Chest pain	{Yes, No}
20	Fever	{Yes, No}
21	Trembling or Shakes	{Yes, No}
22	Weakness	{Yes, No}
23	Sweating	{Yes, No}
24	Sore throat	{Yes, No}
25	Dyspnea	{Yes, No}

26	Dry cough	{ Yes, No }
27	Cough with sputum	{ Yes, No }
28	Fatigue, whole body hurts	{ Yes, No }
29	Anosmia	{ Yes, No }
30	Ageusia	{ Yes, No }
31	Anorexie	{ Yes, No }
32	Eczema	{ Yes, No }
33	Conjunctivitis (Pink eye)	{ Yes, No }
34	Blindness and Tunnel vision	{ Yes, No }
35	Vertigo	{ Yes, No }
36	Nausea/Diarrhea	{ Yes, No }
37	Tobacco	{ Yes, No }

Considering that our dataset has not been released before, it is vital to assess the degree to which each dataset feature contributes to patients' death. Such an analysis provides researchers with valuable insight on the characteristics of the collected data. Various feature selection methods are available to determine the weight of each feature in classification of dataset samples. We chose information gain [31] which is one of the widely used feature selection methods [32]. In Figure 3, the importance of each feature (i.e. information gain) is shown as a bar. The age has a much larger information gain (0.149) compared to other features. Thus age is not included in Figure 3 to make the comparison of other features importance easier. According to the bar chart, (after age) cancer, heart, and kidney diseases are the second, third, and fourth most important feature related to patients' death. Hence, it is clear that the patients with poor health conditions are more vulnerable against COVID-19. It is worth noting that Figure 3 does not include the features with information gain zero.

We have also inspected the interplay between the dataset features to find out the potential correlation between them. To this end the grid in Figure 4 is presented which can be thought as a heat map showing positive/negative correlation between features. Each cell $c(i, j)$ in the grid of Figure 4 represents the correlation of features in i -th row and j -th column. As the cell color approaches red the positive correlation between the feature pairs is higher. For example Anosmia and Ageusia have high positive correlation which means they are usually observed simultaneously. Anosmia is the loss of ability to smell and Ageusia is the inability to taste with tongue.

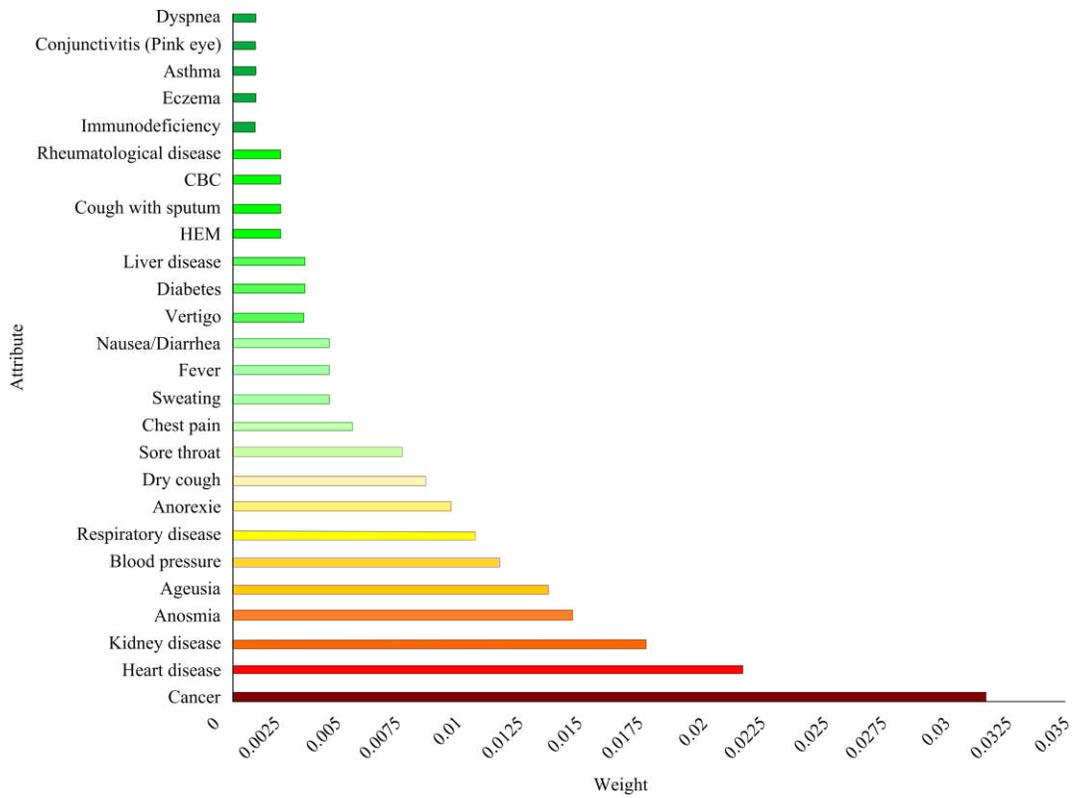


Figure 3. Features effect on mortality rate based on information gain

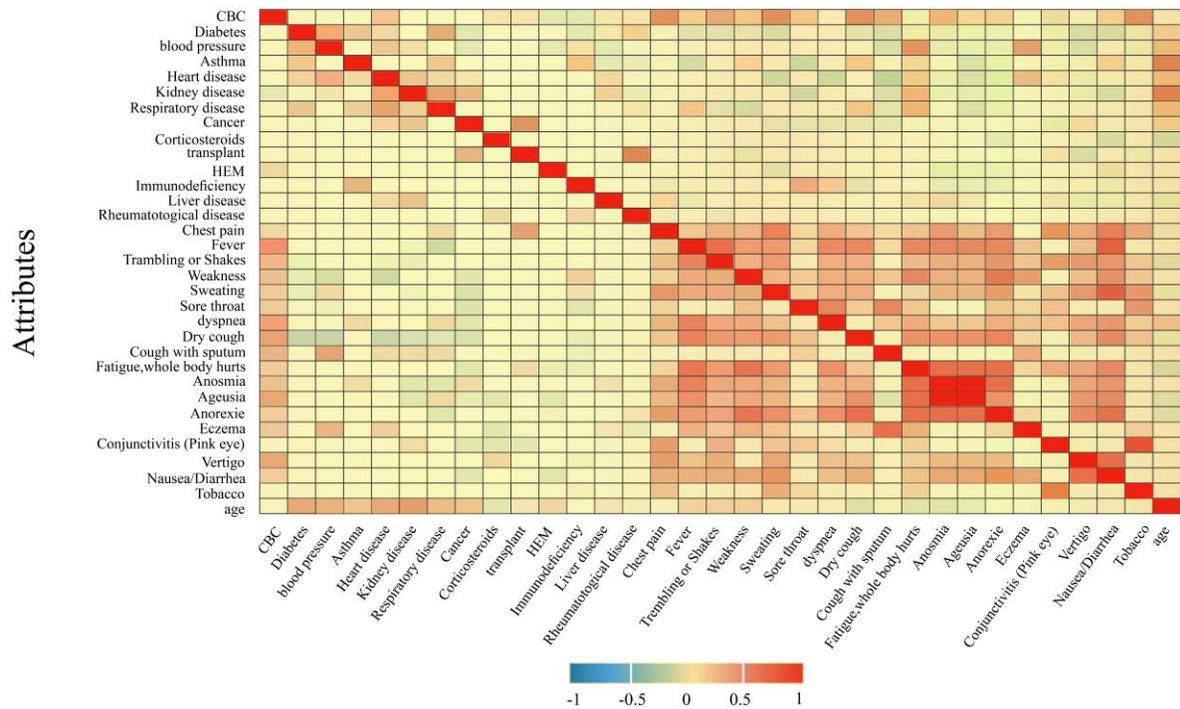


Figure 4. Correlation between dataset features

5 Proposed Methodology

In this study, the survival chance prediction of COVID-19 patients who have referred to Omid hospital in Tehran is investigated. The classification is based on features obtained from patients' information. In the database provided, the number of recovered people is 300 and the number of deceased ones is 20. It is clear that the number of recovered patients is much higher than the number of deceased ones. In order to classify accurately, it is necessary to balance recovered to deceased ratio of dataset samples. To do so, the number of instances of the lower class is increased such that the number of data in both classes is approximately equal. In order to increase the number of data of deceased people, an autoencoder model has been used. To carry out the data augmentation, the 20 samples of deceased class are fed to the autoencoder to undergo compression and decompression routine. The output of this process is 20 reconstructed samples which are similar to the original ones but not identical. Therefore, we have augmented the original 20 samples with 20 reconstructed ones. Training the autoencoder 10 times using different training and validation sets yields 10 autoencoders with similar architecture but different parameters. Each of the 10 autoencoders can generate 20 reconstructed deceased samples yielding overall reconstructed samples of 200 which will be added to the original deceased samples. To gain a better insight on what the autoencoders do, a sample vector before and after reconstruction is presented in table Table 3. For the majority of "1" components of input vector c , the autoencoder has outputted values near 1 as components/elements of the reconstructed vector \hat{c} . Similarly most of the reconstructed components corresponding to original components "0" have values near "0" which shows that the reconstruction process is sound.

Table 3. An example of reconstruction performed by autoencoder: vectors c is the original sample and \hat{c} is its reconstructed counterpart

$c[1:10]$	1	0	0	1	0	0	0	0	0	0
$\hat{c}[1:10]$	0.9940	0.1291	0.0001	0.4697	0.1581	0.0240	0.0525	0.0068	0.0061	0.0202
$c[11:20]$	0	0	0	1	0	0	0	0	1	1
$\hat{c}[11:20]$	0	0	0.0003	0.4004	0.0004	0.0596	0.0040	0.0027	0.9516	0.4450
$c[21:30]$	0	0	0	1	1	0	1	0	0	0
$\hat{c}[21:30]$	0.1305	0.0018	0.0042	0.9565	0.5750	0.0029	0.9281	0.0111	0.0140	0.0966
$c[31:39]$	0	0	0	0	0	0	0	0	1	-
$\hat{c}[31:39]$	0.0087	0.0004	0	0.0110	0.0024	0	0.0017	0.0015	0.9814	-

The details of the augmentation process are explained in more detail in sub-section 5.1. It is worth noting that our augmentation procedure is generic and can be applied to any other dataset.

5.1 Implementation details of CNN-AE

The proposed method (CNN-AE) consists of multiple steps which are summarized in Figure 5. The detailed explanation of these steps is presented below:

1. Design 10 autoencoders with identical configuration but different initial parameters for data augmentation.
2. Train each of the 10 autoencoders on 300 samples representing the recovered patients. We want to have 10 models with different parameters at the end of the training. To this end, we divide the 300 samples to ten groups of 30 samples $\{g_j, j = 1, 2, \dots, 10\}$ where g_j is the j -th group of samples. To train i -th model, g_i is set aside for validation and the nine remaining groups $\{g_j, j \in \{1, 2, \dots, 10\} - \{i\}\}$ (270 samples) are used for training. Recall that each model is initialized with different parameters, trained on partially different training samples, and validated on totally

different validation set. Therefore, the proposed training procedure will yield 10 different autoencoders.

3. The 20 deceased samples are fed to each of the 10 trained autoencoders. The samples undergo the compression and decompression routine of autoencoders. The decompression procedure is lossy so the 20 reconstructed samples (after decompression) are not identical to the original ones. Moreover, the 10 trained autoencoders exhibit different behavior on the same input data since their parameters are different from each other. Therefore, feeding the same 20 samples to the 10 autoencoders will yield 200 new samples which belong to the deceased class. The motivation behind the explained procedure is data augmentation to remedy lack of enough samples for the deceased class.
4. The 200 reconstructed samples are attached to 320 original ones to yield a dataset of 520 samples.
5. Design a CNN model to classify 520 samples as recovered or deceased.
6. Train the CNN model using all 520 samples. We apply 10-fold cross validation during the training. Hence, the training sample size is 468 (samples of 9 folds) and the test sample size is 52 (samples of 1 fold).
7. Use the trained CNN to classify the test data.

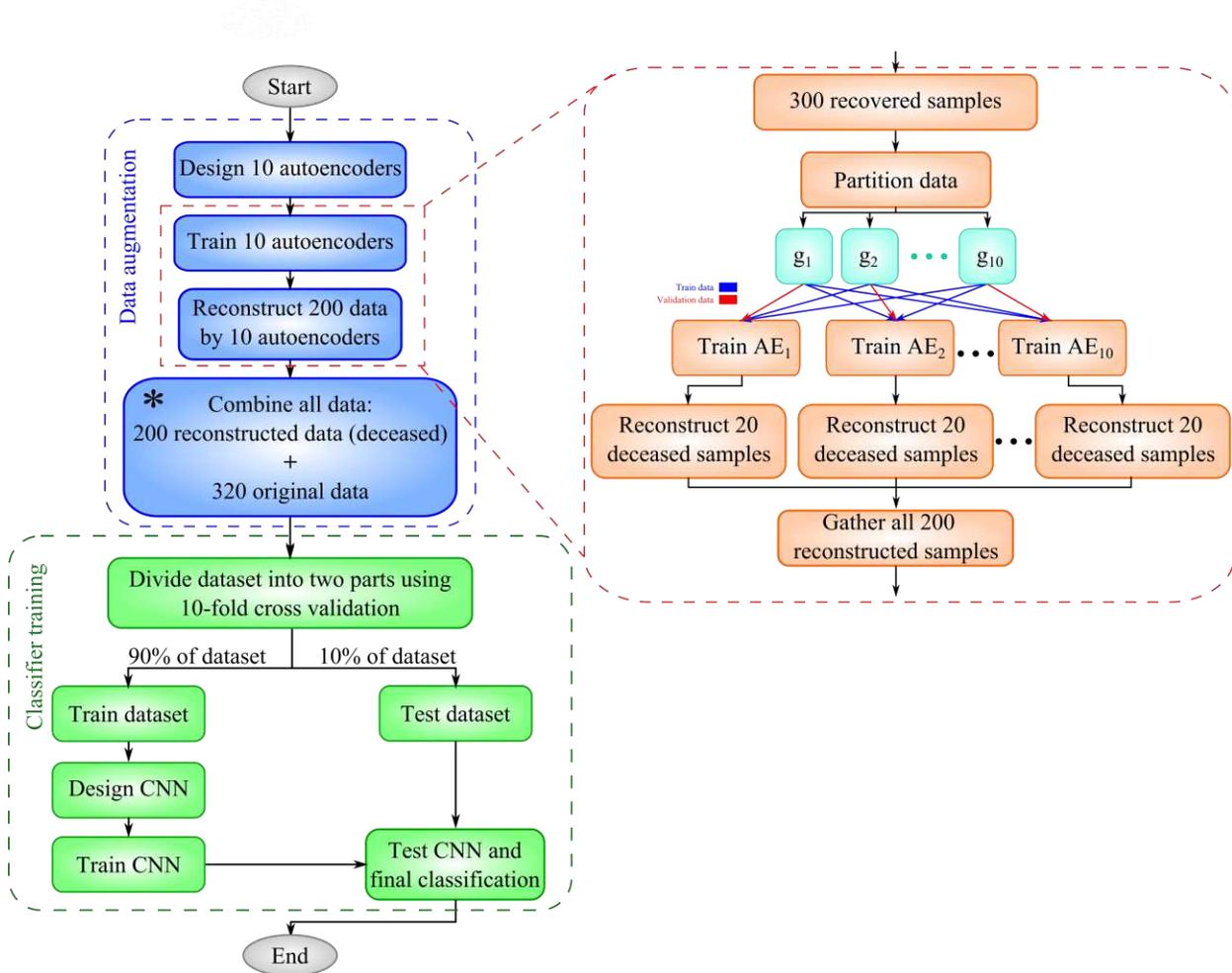


Figure 5. The steps for implementing the proposed method.

To implement the proposed method, we used Python language and Keras library which has TensorFlow backend. In this study, the dataset contains 320 samples of infected cases. The number of recovered cases is 300 and the remaining 20 cases represent the deceased ones. Moreover, we have also generated 200 reconstructed deceased cases to balance the recovered to deceased ratio of our dataset. After the reconstruction phase, our dataset contained 520 cases. We have used 10-fold cross validation. Moreover, 80% of 9 of the folds are used for training and the remaining 20% are used for validation. The implementation details of CNN and AE are illustrated in Figure 6 and Figure 7, respectively.

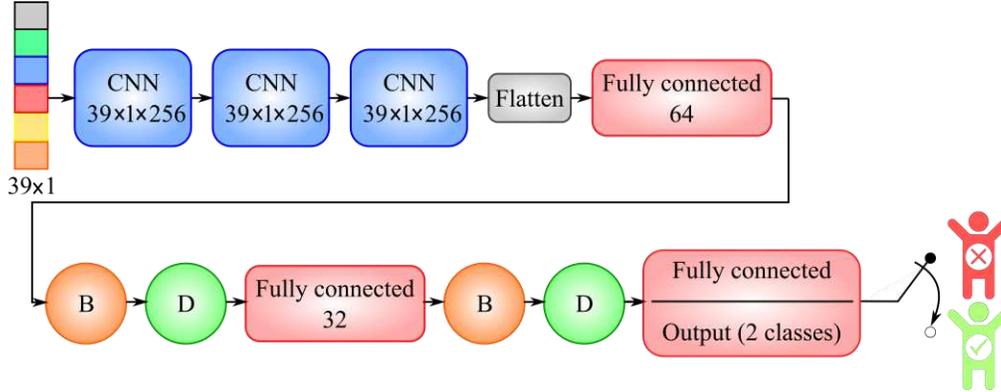


Figure 6. Implementation details of CNN: circles with “B” letter represent batch normalization layers and circles with “D” letter represent dropout layers with probability 0.5.

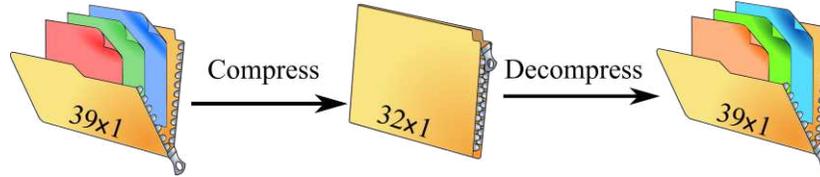


Figure 7. Implementation details of Autoencoder

6 Experiments

In this section, the experimental results are presented. The implementation details of CNN and autoencoders are explained in section 6.1. We report the performance of the proposed method (CNN-AE) and compare it with a CNN in section 6.2.

6.1 Experiments setup/details

Our experiments consist of two scenarios. In the first scenario, our method (CNN-AE) is compared with a standard CNN which is trained on clinical data. The architecture of the CNN is presented in

Table 4. To have a fair comparison, we use the same CNN architecture in our method. Moreover, implementation details of autoencoders used in CNN-AE are presented in Table 5.

Table 4. Implementation details of the CNN which has been trained on clinical data.

Hyper-parameters	Values
Input dimension	39×1 (39 medical features)

Number of convolution layers	3
Number of fully connected layers	3
Number of filters for each convolution layer	256
Size of convolutional kernels	3×3
Strides size	1
Activation function for hidden layers	ReLU
Activation function of last layer	Sigmoid
Adam hyper-parameters	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning rate	0.001
Loss function	Binary Cross Entropy (BCE)
Number of neurons of fully connected layers	64, 32, 2
Dropout probability	0.5
Number of epochs	100

Table 5. Autoencoder implementation details

Hyper-parameters	values
Input dimension	39×1 (39 medical features)
Number of neurons of first layer	39×1
Number of neurons of second layer	32×1
Number of neurons of third layer	39×1
First and second layers activation function	ReLU
Third layer activation function	Sigmoid
Adam hyper-parameters	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning rate	0.001
Loss function	Binary Cross Entropy (BCE)
Number of epochs	100

In the second phase of our experiments, we compare CNN-AE (trained on clinical data) with a standard CNN which is trained on image data. The CNN architecture is presented in Figure 8. After multiple trials, we ended up with the best set of CNN hyperparameters which are presented in Table 6.

Table 6. Implementation details of the CNN which has been trained on image data

Hyper-parameters	Values
Number of convolutional kernels of first layer	64
Number of convolutional kernels of second layer	128
Number of convolutional kernels of third layer	256
Size of convolutional kernels	3×3
Strides size	2
Input dimension	100×100
Output dimension	2
Number of convolution layers	3
Number of fully connected layers	2
Activation function for convolutional and fully connected layers	ReLU
Activation function of last layer	Sigmoid
Adam hyper-parameters	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning rate	0.001
Loss function	Binary Cross Entropy (BCE)
Number of neurons of forth layer (fully connected)	256

Number of neurons of fifth layer (fully connected)	128
Dropout probability	0.5
Number of epochs	30
Batch size	128

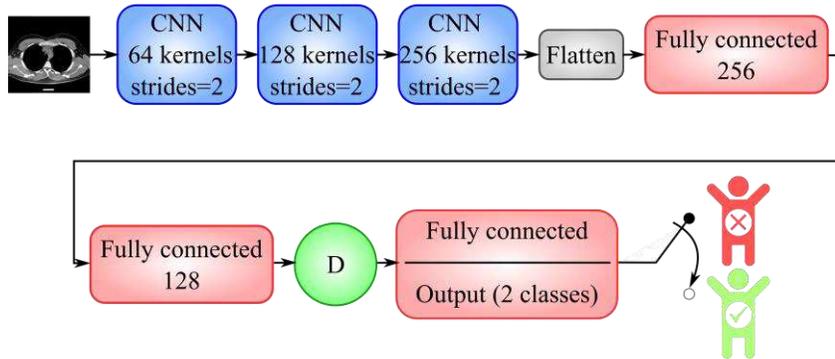


Figure 8. Implementation details of the CNN which is trained on CT images.

6.2 Experimental results

In this section, we seek to answer two important questions about the proposed method. First, we compare our method performance with a standard CNN which is trained on clinical data. This experiment reveals the effect of the proposed data augmentation technique using multiple autoencoders. Next, we train a standard CNN for the same purpose (survival chance prediction) but using CT images. This experiment determines how well CT images can represent patients' survival chance using a CNN as the predictor.

6.2.1 Inspecting data augmentation approach

As mentioned in section 5.1, we use 10 autoencoders to augment the available dataset. Data augmentation is critical to successful training when the number of samples from different classes is unbalanced. Data imbalance can bring any powerful classifier to its knees even a state of the art CNN which is the motivation for our data augmentation technique.

To investigate the effectiveness of our data augmentation procedure, we have trained a CNN on original dataset and our CNN-AE on the augmented one. The original dataset has only 20 samples with deceased label while the recovered samples are 300. Comparing 300 with 20 reveals severe data imbalance from which CNN has suffered during training as shown in

Table 7. On the other hand, using augmented dataset with 300 recovered samples and 220 deceased ones has facilitated the CNN training leading to better accuracy (

Table 7). Moreover, AUC measure of CNN-AE is almost twice the AUC of CNN. The specificity measure of CNN is mostly zero which stems from the fact that CNN has been unable to distinguish deceased samples from the recovered ones due to insufficient number of deceased samples in the original dataset. Looking at

Table 7, it is clear that CNN-AE training takes more time which is due to the time it takes to train 10 autoencoders required for data augmentation.

Table 7. Comparison of CNN and CNN-AE using different evaluation metrics based on 10-fold cross validation.

Methods	Fold No.	Accuracy (%)	PPV (%)	Recall (%)	Specificity (%)	F1-score (%)	AUC (%)	Loss	Total training time (s)
CNN	1	93.75	94	100	0	97	50	0.2157	27.04
	2	93.75	94	100	0	97	50	0.2104	20.48
	3	93.75	94	100	0	97	50	0.2733	21.34
	4	90.63	97	94	0	95	46.77	0.6140	21.49
	5	96.88	97	100	66.67	98	83.33	0.0916	21.70
	6	90.63	97	94	0	95	46.77	0.2164	22.00
	7	93.75	94	100	0	97	50	0.1642	21.51
	8	93.75	94	100	0	97	50	0.1816	21.83
	9	81.25	96	81	80	88	80.74	0.7327	21.97
	10	96.77	97	100	0	98	50	0.1214	22.75
95% confidence interval over 10 folds		92.49±2.75	95.4±0.88	96.9±3.73	14.67±19.00	95.9±1.82	55.76±8.54	0.282±0.13	22.21±0.37
CNN-AE	1	98.08	97	100	94.74	99	97.37	0.0925	33.01
	2	94.23	94	97	90.91	95	93.79	0.2600	31.50
	3	100	100	100	100	100	100	0.0096	31.35
	4	96.15	96	96	95.83	96	96.13	0.2600	31.60
	5	93.27	91	97	85	94	90.94	0.4017	31.90
	6	92.31	94	94	90.48	94	92.01	0.3678	31.99
	7	98.08	97	100	95.45	98	97.73	0.0858	32.30
	8	96.15	94	100	90.91	97	95.45	0.2027	32.90
	9	94.23	92	96	92.59	94	94.3	0.1572	33.13
	10	98.04	97	100	95.45	98	97.73	0.0614	33.83
95% confidence interval over 10 folds		96.05±1.48	95.2±1.63	98±1.33	93.14±2.52	96.5±1.27	95.55±1.70	0.19±0.07	32.35±0.49

In

Table 7, CNN-AE method with average accuracy of **96.05%** outperforms CNN method with average accuracy of **92.49%**. Moreover, thanks to the augmented data, our method has been able to reduce training/validation loss faster than CNN which is evident in Figure 9. Similarly, CNN-AE has managed to reach higher accuracy faster than CNN which can be seen in plots of Figure 10. During training, our method exhibits much variation in validation plots compared to CNN. The reason is that CNN has quickly overfit to small number of deceased samples but CNN-AE had to deal with more versatile augmented samples. Therefore, CNN-AE had harder time during training but could gain better overall performance.

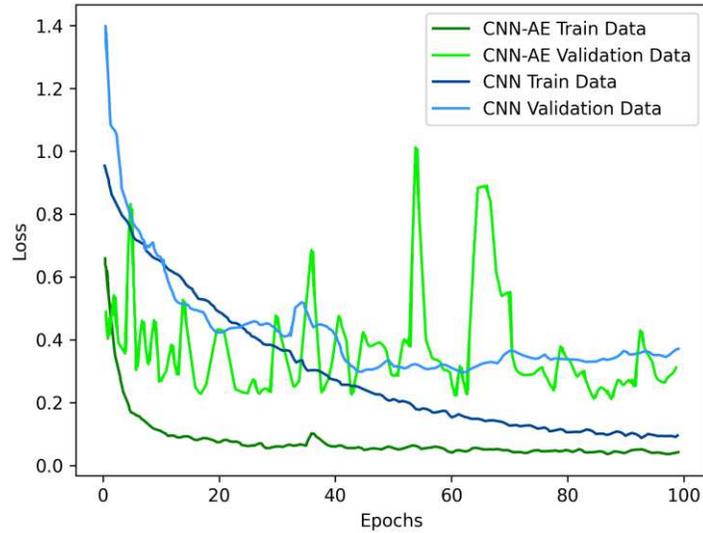


Figure 9. Loss plots of CNN and CNN-AE methods during training on our dataset

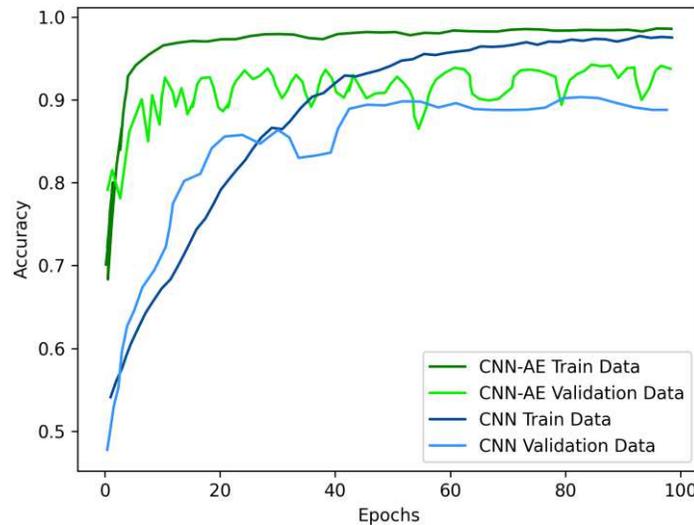


Figure 10. Accuracy plots of CNN and CNN-AE methods during training on our dataset

6.2.2 Comparison with image-based CNN

In this section we evaluate the performance of a standard CNN which is trained on a dataset of CT images. The CT images have been taken from the same patients for whom the clinical dataset has been collected. Therefore, the results of this section reveal how well a CNN trained on CT images performs compared to a CNN trained on clinical data. It is worth noting that most of the conducted experiments in the COVID-19 literature revolve around classifying infected and non-infected people using CT images. However, this section sheds some light on how well a CNN can predict survival chance of already infected patients based on CT images.

The dataset consists of 2822 CT image of recovered patients and 2269 CT images of deceased ones. The CT image dataset size is much greater than clinical dataset size. The reason is that the CT dataset contains multiple images per each patient. Since the number of samples of the two classes in the dataset is almost balanced, we did not apply our data augmentation technique to the CT dataset. Moreover, having multiple

images per each patient acts as some form of data augmentation. This is not the case for clinical dataset where for each patient only one value for each feature is present.

The loss and accuracy plots of training the CNN on CT images are presented in parts (a) and (b) of Figure 11. Moreover, in Table 8, the performance metrics are presented as 95% confidence intervals which have been computed over 10-fold cross validation. Looking at Figure 11 reveals that training on CT images yields good results. Therefore, CT images can be considered a reliable source for survival chance prediction of COVID-19 patients. Comparison of Figure 11b with CNN-AE plots in Figure 10 suggests that training a CNN on clinical data performs on par with training on CT data. Recall that the CT image dataset size is almost ten times of clinical dataset size. Yet, the CNN trained on clinical data can almost perform as well as the CNN trained on CT data. Hence, clinical data can be a good replacement for CT training samples whenever preparing CT images is difficult or expensive.

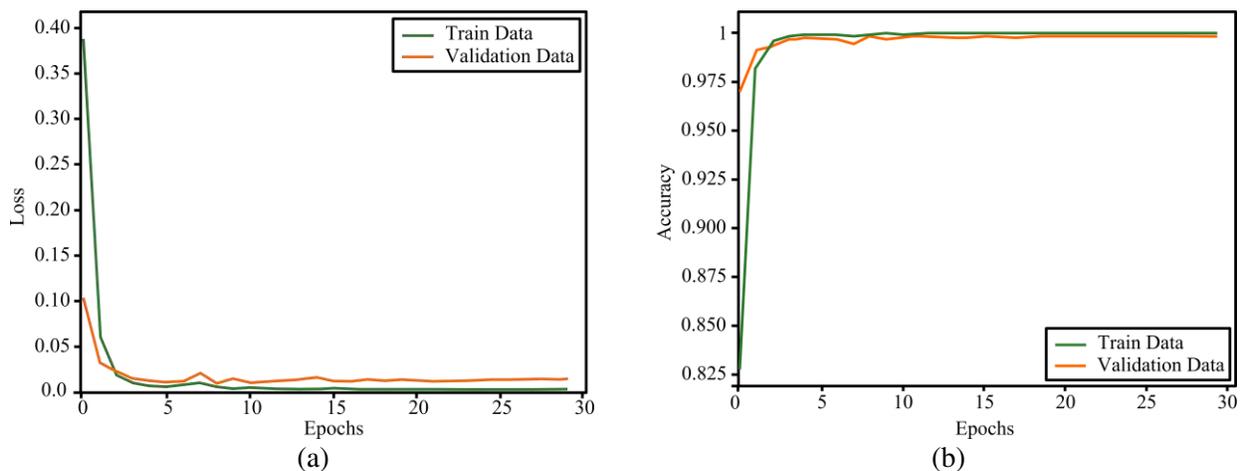


Figure 11. CNN trained on CT images: (a) loss plot, (b) accuracy plot

Table 8. Results of CNN trained on CT images

Accuracy (%)	PPV (%)	Recall (%)	Specificity (%)	F1-score (%)	AUC (%)	Loss
98.88 ± 1.09	98.9 ± 1.07	98.9 ± 0.91	98.10 ± 1.22	98.90 ± 0.87	98.89 ± 0.92	0.012 ± 0.005

6.3 Comparison with other methods

We have also compared our method performance with other commonly used classification methods such as Naïve Bayes classifier, random forest, logistic regression, and support vector machine (SVM). To get a better understanding of the working mechanism of these methods, first we review them briefly and next present the experimental results for them.

Naïve Bayes

Naïve Bayes classifier is based on Bayesian network which is a probabilistic graphical model. Given the observed event, a Bayesian network predicts the likelihood of the cause of that event. For example, given symptoms, the network can predict probabilities of the various diseases.

Radom forest

As the name implies, random forest is a set of decision trees which are used in an ensemble setup to get highly accurate classification results and avoid overfitting at the same time. Each of the trees is built using a random subset of available training data. The subset samples are draw randomly with replacement from the training data. Moreover tree nodes are split based on a random subset of the training data features.

During the testing phase, decision trees of the forest provide their judgment on the given test sample which are averaged to yield the final prediction [33, 34].

6.3.1 Methods performance

In this section we report experimental result using other classification methods such as random forest and Naïve Bayes. We also investigate the effect of using the proposed data augmentation technique during the training. Such an experiment demonstrates that our data augmentation method is generic and applicable with other methods as well. The performance statistics are presented with 95% confidence interval in Table 9. The confidence intervals are computed based on 10-fold cross validation. For each method we have performed two experiments. First, each of the methods is trained on the original dataset (without augmentation) the results of which are presented in first and third rows of Table 9. Once again the methods are trained but on the augmented dataset. The augmentation is carried out using our autoencoder based approach. The results are shown in second and fourth rows of Table 9. Both of random forest and Naïve Bayes have clearly benefitted from the augmentation performed on the training dataset. The augmented dataset has led to significant improvement of Naïve Bayes accuracy. The augmentation has mildly improved random forest accuracy but yielded much higher AUC. The last row of Table 9 is identical to the last row of

Table 7 which is restated here for ease of reference. It is evident that the proposed method CNN-AE has outperformed other methods both in terms of accuracy and AUC.

Table 9. Performance metrics for RF and Naïve Bayes with and without autoencoder based data augmentation

Methods	Accuracy (%)	PPV (%)	Recall (%)	Specificity (%)	F1-score (%)	AUC (%)
RF	92.36 ± 3.98	11.11 ± 2.36	36.67 ± 4.63	97.54 ± 2.36	5.56 ± 2.35	51.26 ± 4.12
RF+AE	94.44 ± 3.65	97.20 ± 3.19	90.50 ± 4.79	98.23 ± 1.34	93.70 ± 3.56	94.59 ± 3.72
Naïve Bayes	61.73 ± 6.93	14.30 ± 5.68	88.00 ± 5.65	42.65 ± 5.95	23.90 ± 2.39	74.39 ± 5.68
Naïve Bayes+AE	74.92 ± 5.49	63.80 ± 6.89	96.40 ± 3.47	60.74 ± 4.45	76.50 ± 4.29	78.46 ± 4.92
CNN-AE	96.05 ± 1.48	95.20 ± 1.63	98.00 ± 1.33	93.13 ± 2.52	96.50 ± 1.27	95.54 ± 1.70

7 Discussion

This paper focuses on survival chance prediction for COVID-19 patients. We performed experiments using clinical dataset as well as CT images dataset. Recall that the CT image dataset size is almost ten times of clinical dataset size. Yet, the CNN trained on clinical data could almost perform as well as the CNN trained on CT data which encourages using clinical data as an alternative for CT images.

Another aspect that may encourage using clinical training samples is the data collection cost. Preparing CT data may require high-end facilities which lead to increased data collection cost. Moreover, the required facilities to prepare CT data may not be available in deprived areas whereas the requirements to measure clinical data such as blood pressure, fever, CRP, etc. are accessible.

The proposed method can detect the severity of patients' condition based on clinical data and take preventive actions to minimize the mortality rate. As reviewed in section 2, few methods have studied mortality rate prediction using clinical data. Moreover, the existing methods have used features different than the ones we have used in our experiments. Therefore, the proposed method sheds some light on unexplored aspects of COVID-19 virus. In addition to the proposed method, our dataset can be considered as the second contribution of this paper since it is a good resource for further medical research. To the best of our knowledge we are the first to analyze the importance of the dataset features and their correlation as presented in Figure 3 and Figure 4. Using our dataset, the experts can study the relation between patients' medical conditions (e.g. blood pressure, diabetes) and COVID-19 death likelihood. As

a result, the medical experts can exercise more caution during treatment of patients who are more likely to die due to their medical condition. As the information gain values in Figure 3 suggest, there is a strong relation between mortality rate of COVID-19 patients and having critical diseases such as cancer, kidney and heart diseases. On the other hand, mild symptoms and/or diseases like dyspnea, conjunctivitis, and asthma are less likely to contribute to the mortality rate.

Like any other classification approach, the proposed method has some shortcomings as well. Due to using multiple autoencoders in the data augmentation phase, the training time of our method is longer than a standard CNN. Moreover, standard CNNs receive a single image sample as input and perform feature extraction automatically. On the contrary, we have collected multiple clinical features by hand for each patient which is harder to manage. Some of the features in our dataset have been asked directly from the patient so it is possible that the patient provides wrong information.

8 Conclusions and Future works

In this paper we investigated the possibility of using clinical data to train a CNN for survival chance prediction of COVID-19 patients. The experiments have been conducted on a clinical dataset which we have collected. Our dataset consists of new features that have not been studied before. We have made the dataset publicly available and analyzed its features using information gain and correlation. Such an analysis can aid the potential researchers and practitioners with their work on COVID-19 virus. To reduce the data imbalance of our dataset, we proposed a novel data augmentation method based on autoencoders. Our data augmentation approach is generic and applicable to other datasets as well. The effect of using data augmentation was investigated to evaluate its effectiveness. Using augmented data for training, CNN-AE has been able to reach accuracy $96.05 \pm 1.48\%$, recall $98.00 \pm 1.33\%$, and specificity $93.13 \pm 2.52\%$. However, training a CNN on dataset without augmentation has yielded accuracy $92.49 \pm 2.75\%$, recall $95.4 \pm 0.88\%$, and specificity $96.9 \pm 3.73\%$. It is clear that CNN-AE has benefitted the data augmentation and outperformed CNN.

We repeated the CNN training but this time using CT images of the same patients from clinical dataset. Comparing the performance of training on clinical data with training on CT data reveals that clinical data can be used as an alternative for CT images.

As future work, it is important to collect more data to be able to assess the proposed approach better. It is also important to investigate the use of other data augmentation methods and compare the results with our data augmentation.

Author contributions

Contributed to prepare the first draft: R.A., S.H., A.S., F.K., N.H.I., and J.H.J.

Contributed to editing the final draft: S.N., Z.A.S., A.K., S.M.S.I., H.M., and J.M.G.

Contributed to all analysis of the data and produced the results accordingly: D.S., N.H.I., and R.A.

Searched for papers and then extracted data: S.H., A.S., F.K., and J.H.J.

Provided overall guidance and managed the project: S.N., Z.A.S., A.K., S.M.S.I., H.M., and J.M.G.

Acknowledgement

This work was partly supported by the MINECO/ FEDER under the RTI2018-098913-B100, CV20-45250 and A-TIC-080-UGR18 projects.

References

- [1] R. Alizadehsani, Z. Alizadeh Sani, M. Behjati, Z. Roshanzamir, S. Hussain, N. Abedini, F. Hasanzadeh, A. Khosravi, A. Shoeibi, M. Roshanzamir, P. Moradnejad, S. Nahavandi, F. Khozeimeh, A. Zare, M. Panahiazar, U.R. Acharya, S.M.S. Islam, Risk factors prediction, clinical outcomes, and mortality in COVID-19 patients, *Journal of Medical Virology*, 93 (2021) 2307-2320.
- [2] U.R. Acharya, S.L. Fernandes, J.E. WeiKoh, E.J. Ciaccio, M.K.M. Fabell, U.J. Tanik, V. Rajinikanth, C.H. Yeong, Automated Detection of Alzheimer's Disease Using Brain MRI Images– A Study with Various Feature Extraction Techniques, *Journal of Medical Systems*, 43 (2019) 302.
- [3] S.L. Oh, Y. Hagiwara, U. Raghavendra, R. Yuvaraj, N. Arunkumar, M. Murugappan, U.R. Acharya, A deep learning approach for Parkinson's disease diagnosis from EEG signals, *Neural Computing and Applications*, 32 (2020) 10927-10933.
- [4] R. Alizadehsani, A. Khosravi, M. Roshanzamir, M. Abdar, N. Sarrafzadegan, D. Shafie, F. Khozeimeh, A. Shoeibi, S. Nahavandi, M. Panahiazar, A. Bishara, R.E. Beygui, R. Puri, S. Kapadia, R.-S. Tan, U.R. Acharya, Coronary artery disease detection using artificial intelligence techniques: A survey of trends, geographical differences and diagnostic features 1991–2020, *Computers in Biology and Medicine*, 128 (2021) 104095.
- [5] J.M. Górriz, J. Ramírez, A. Ortíz, F.J. Martínez-Murcia, F. Segovia, J. Suckling, M. Leming, Y.-D. Zhang, J.R. Álvarez-Sánchez, G. Bologna, P. Bonomini, F.E. Casado, D. Charte, F. Charte, R. Contreras, A. Cuesta-Infante, R.J. Duro, A. Fernández-Caballero, E. Fernández-Jover, P. Gómez-Vilda, M. Graña, F. Herrera, R. Iglesias, A. Lekova, J. de Lope, E. López-Rubio, R. Martínez-Tomás, M.A. Molina-Cabello, A.S. Montemayor, P. Novais, D. Palacios-Alonso, J.J. Pantrigo, B.R. Payne, F. de la Paz López, M.A. Pinninghoff, M. Rincón, J. Santos, K. Thurnhofer-Hemsi, A. Tsanas, R. Varela, J.M. Ferrández, Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications, *Neurocomputing*, 410 (2020) 237-270.
- [6] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan, China, *Jama*, 323 (2020) 1061-1069.
- [7] A. Shoeibi, M. Khodatars, R. Alizadehsani, N. Ghassemi, M. Jafari, P. Moridian, A. Khadem, D. Sadeghi, S. Hussain, A. Zare, Automated detection and forecasting of covid-19 using deep learning techniques: A review, *arXiv preprint arXiv:2007.10785*, (2020).
- [8] D. Sharifrazi, R. Alizadehsani, M. Roshanzamir, J.H. Joloudari, A. Shoeibi, M. Jafari, S. Hussain, Z.A. Sani, F. Hasanzadeh, F. Khozeimeh, A. Khosravi, S. Nahavandi, M. Panahiazar, A. Zare, S.M.S. Islam, U.R. Acharya, Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19 patients using X-ray images, *Biomedical Signal Processing and Control*, 68 (2021) 102622.
- [9] R. Alizadehsani, D. Sharifrazi, N.H. Izadi, J.H. Joloudari, A. Shoeibi, J.M. Gorriz, S. Hussain, J.E. Arco, Z.A. Sani, F. Khozeimeh, Uncertainty-Aware Semi-supervised Method using Large Unlabelled and Limited Labeled COVID-19 Data, *arXiv preprint arXiv:2102.06388*, (2021).
- [10] Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, R. Ren, K.S. Leung, E.H. Lau, J.Y. Wong, Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia, *New England Journal of Medicine*, (2020).
- [11] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao, Y. Hu, L. Zhang, G. Fan, J. Xu, X. Gu, Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China, *The lancet*, 395 (2020) 497-506.
- [12] V.M. Martín Giménez, F. Inserra, C.D. Tajer, J. Mariani, L. Ferder, R.J. Reiter, W. Manucha, Lungs as target of COVID-19 infection: Protective common molecular mechanisms of vitamin D and melatonin as a new potential synergistic treatment, *Life Sciences*, 254 (2020) 117808.

- [13] H. Asgharnezhad, A. Shamsi, R. Alizadehsani, A. Khosravi, S. Nahavandi, Z.A. Sani, D. Srinivasan, Objective Evaluation of Deep Uncertainty Predictions for COVID-19 Detection, arXiv preprint arXiv:2012.11840, (2020).
- [14] N. Zhang, L. Wang, X. Deng, R. Liang, M. Su, C. He, L. Hu, Y. Su, J. Ren, F. Yu, Recent advances in the detection of respiratory virus infection in humans, *Journal of medical virology*, 92 (2020) 408-417.
- [15] C. Iwendi, A.K. Bashir, A. Peshkar, R. Sujatha, J.M. Chatterjee, S. Pasupuleti, R. Mishra, S. Pillai, O. Jo, COVID-19 Patient health prediction using boosted random forest algorithm, *Frontiers in public health*, 8 (2020) 357.
- [16] J.H. Joloudari, M. Haderbadi, A. Mashmool, M. GhasemiGol, S.S. Band, A. Mosavi, Early detection of the advanced persistent threat attack using performance analysis of deep learning, *IEEE Access*, 8 (2020) 186125-186137.
- [17] J. Wu, P. Zhang, L. Zhang, W. Meng, J. Li, C. Tong, Y. Li, J. Cai, Z. Yang, J. Zhu, M. Zhao, H. Huang, X. Xie, S. Li, Rapid and accurate identification of COVID-19 infection through machine learning based on clinical available blood test results, medRxiv, (2020) 2020.2004.2002.20051136.
- [18] M. Rahman, M. Uddin, M. Wadud, A. Akhter, O. Akter, A Study on Epidemiological Characteristics and ML Based Detection of novel COVID-19, Available in <https://www.researchgate.net/publication/340246803>, (2020).
- [19] Y. Chen, L. Ouyang, F.S. Bao, Q. Li, L. Han, B. Zhu, Y. Ge, P. Robinson, M. Xu, J. Liu, An interpretable machine learning framework for accurate severe vs non-severe covid-19 clinical type classification, Available at SSRN 3638427, (2020).
- [20] Y. Gao, G.-Y. Cai, W. Fang, H.-Y. Li, S.-Y. Wang, L. Chen, Y. Yu, D. Liu, S. Xu, P.-F. Cui, S.-Q. Zeng, X.-X. Feng, R.-D. Yu, Y. Wang, Y. Yuan, X.-F. Jiao, J.-H. Chi, J.-H. Liu, R.-Y. Li, X. Zheng, C.-Y. Song, N. Jin, W.-J. Gong, X.-Y. Liu, L. Huang, X. Tian, L. Li, H. Xing, D. Ma, C.-R. Li, F. Ye, Q.-L. Gao, Machine learning based early warning system enables accurate mortality risk prediction for COVID-19, *Nature Communications*, 11 (2020) 5033.
- [21] S. Si, H. Zhang, S.S. Keerthi, D. Mahajan, I.S. Dhillon, C.-J. Hsieh, Gradient boosted decision trees for high dimensional sparse output, *International conference on machine learning*, PMLR, 2017, pp. 3182-3190.
- [22] J.S. Zhu, P. Ge, C. Jiang, Y. Zhang, X. Li, Z. Zhao, L. Zhang, T.Q. Duong, Deep-learning artificial intelligence analysis of clinical variables predicts mortality in COVID-19 patients, *Journal of the American College of Emergency Physicians Open*, 1 (2020) 1364-1373.
- [23] L. Yan, H.-T. Zhang, J. Goncalves, Y. Xiao, M. Wang, Y. Guo, C. Sun, X. Tang, L. Jing, M. Zhang, X. Huang, Y. Xiao, H. Cao, Y. Chen, T. Ren, F. Wang, Y. Xiao, S. Huang, X. Tan, N. Huang, B. Jiao, C. Cheng, Y. Zhang, A. Luo, L. Mombaerts, J. Jin, Z. Cao, S. Li, H. Xu, Y. Yuan, An interpretable mortality prediction model for COVID-19 patients, *Nature Machine Intelligence*, 2 (2020) 283-288.
- [24] D. Bertsimas, G. Lukin, L. Mingardi, O. Nohadani, A. Orfanoudaki, B. Stellato, H. Wiberg, S. Gonzalez-Garcia, C.L. Parra-Calderon, K. Robinson, COVID-19 mortality risk assessment: An international multi-center study, *PloS one*, 15 (2020) e0243262.
- [25] A. Abdulaal, A. Patel, E. Charani, S. Denny, N. Mughal, L. Moore, Prognostic Modeling of COVID-19 Using Artificial Intelligence in the United Kingdom: Model Development and Validation, *J Med Internet Res*, 22 (2020) e20259.
- [26] A.M.U.D. Khanday, S.T. Rabani, Q.R. Khan, N. Rouf, M. Mohi Ud Din, Machine learning based approaches for detecting COVID-19 using clinical text data, *International Journal of Information Technology*, 12 (2020) 731-739.
- [27] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, T. Sainath, Deep Learning for Audio Signal Processing, *IEEE Journal of Selected Topics in Signal Processing*, 13 (2019) 206-219.

- [28] M. Khodatars, A. Shoeibi, N. Ghassemi, M. Jafari, A. Khadem, D. Sadeghi, P. Moridian, S. Hussain, R. Alizadehsani, A. Zare, Deep Learning for Neuroimaging-based Diagnosis and Rehabilitation of Autism Spectrum Disorder: A Review, arXiv preprint arXiv:2007.01285, (2020).
- [29] M. Tschannen, O. Bachem, M. Lucic, Recent advances in autoencoder-based representation learning, arXiv preprint arXiv:1812.05069, (2018).
- [30] <https://www.kaggle.com/danielsharifrazi/covid19-numeric-dataset/settings>.
- [31] R. Alizadehsani, J. Habibi, M.J. Hosseini, H. Mashayekhi, R. Boghrati, A. Ghandeharioun, B. Bahadorian, Z.A. Sani, A data mining approach for diagnosis of coronary artery disease, Computer Methods and Programs in Biomedicine, 111 (2013) 52-61.
- [32] R. Alizadehsani, M. Roshanzamir, M. Abdar, A. Beykikhoshk, A. Khosravi, M. Panahiazar, A. Koohestani, F. Khozeimeh, S. Nahavandi, N. Sarrafzadegan, A database for using machine learning and data mining techniques for coronary artery disease diagnosis, Scientific Data, 6 (2019) 227.
- [33] C.R. Sekhar, Minal, E. Madhu, Mode Choice Analysis Using Random Forrest Decision Trees, Transportation Research Procedia, 17 (2016) 644-652.
- [34] M. Pal, Random forest classifier for remote sensing classification, International Journal of Remote Sensing, 26 (2005) 217-222.

Figures

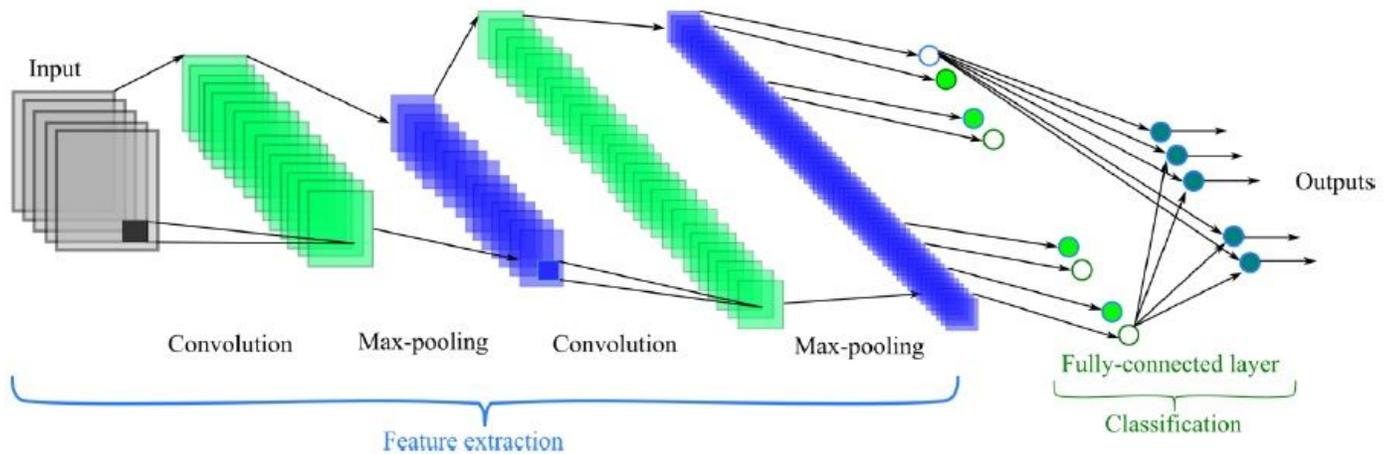


Figure 1

A CNN schematic

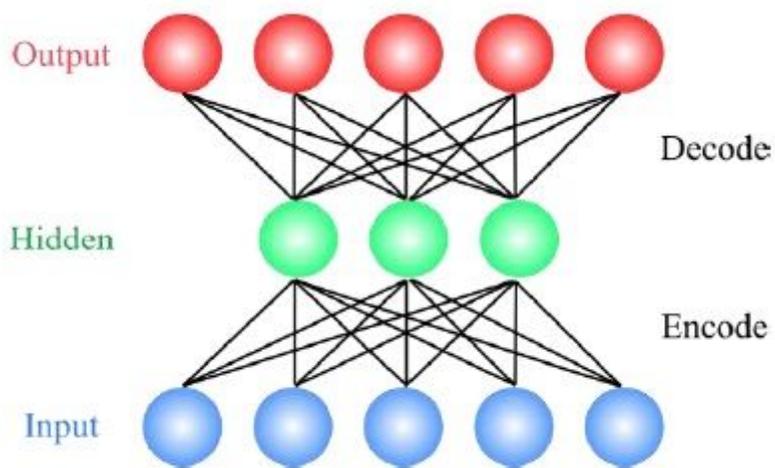


Figure 2

Autoencoder architecture: high dimension input data is encoded (compressed) to form latent (hidden) space which has a lower dimension compared to the original input. The latent representation is reconstructed (decoded) to yield decompressed output.

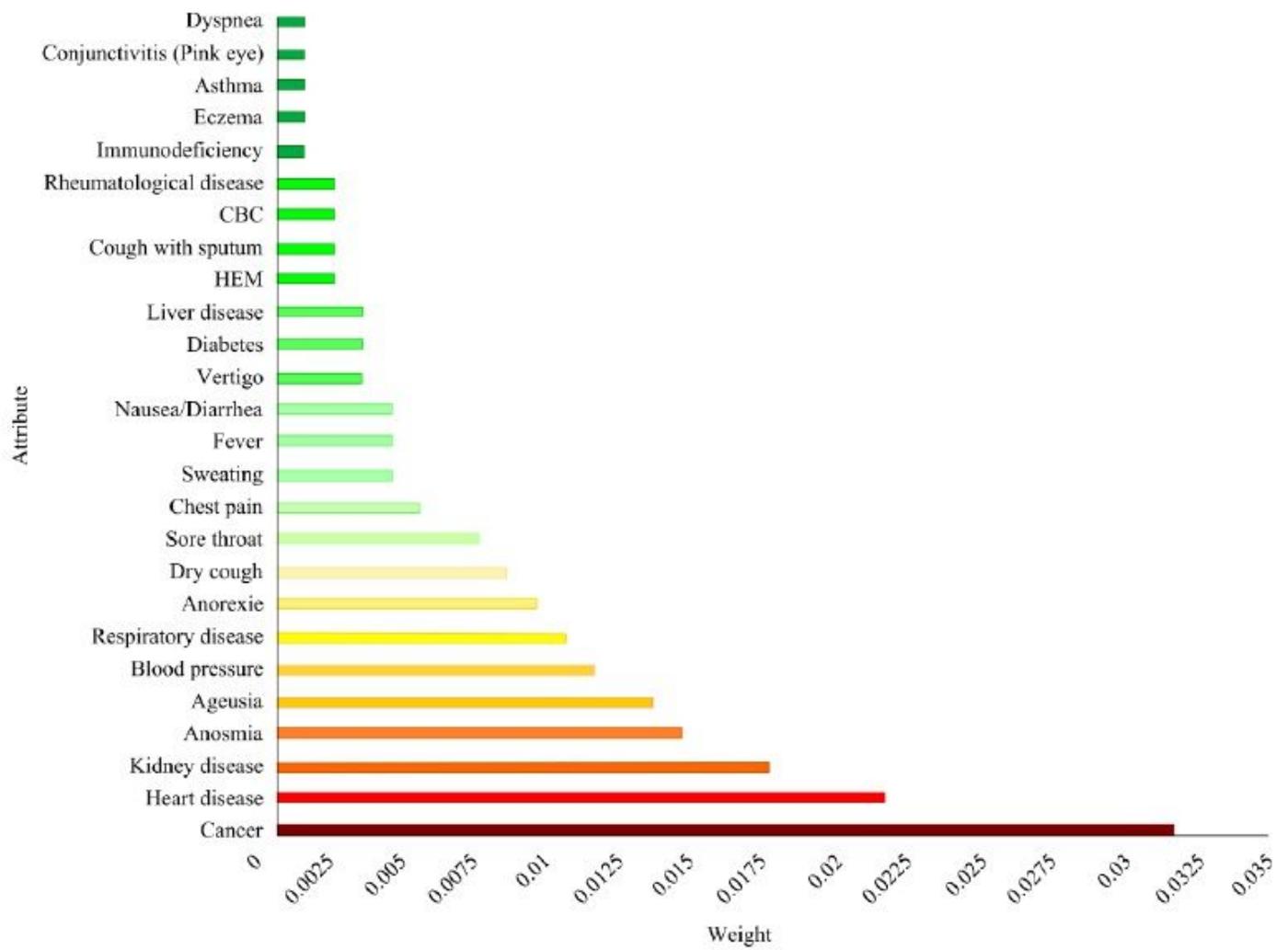


Figure 3

Features effect on mortality rate based on information gain

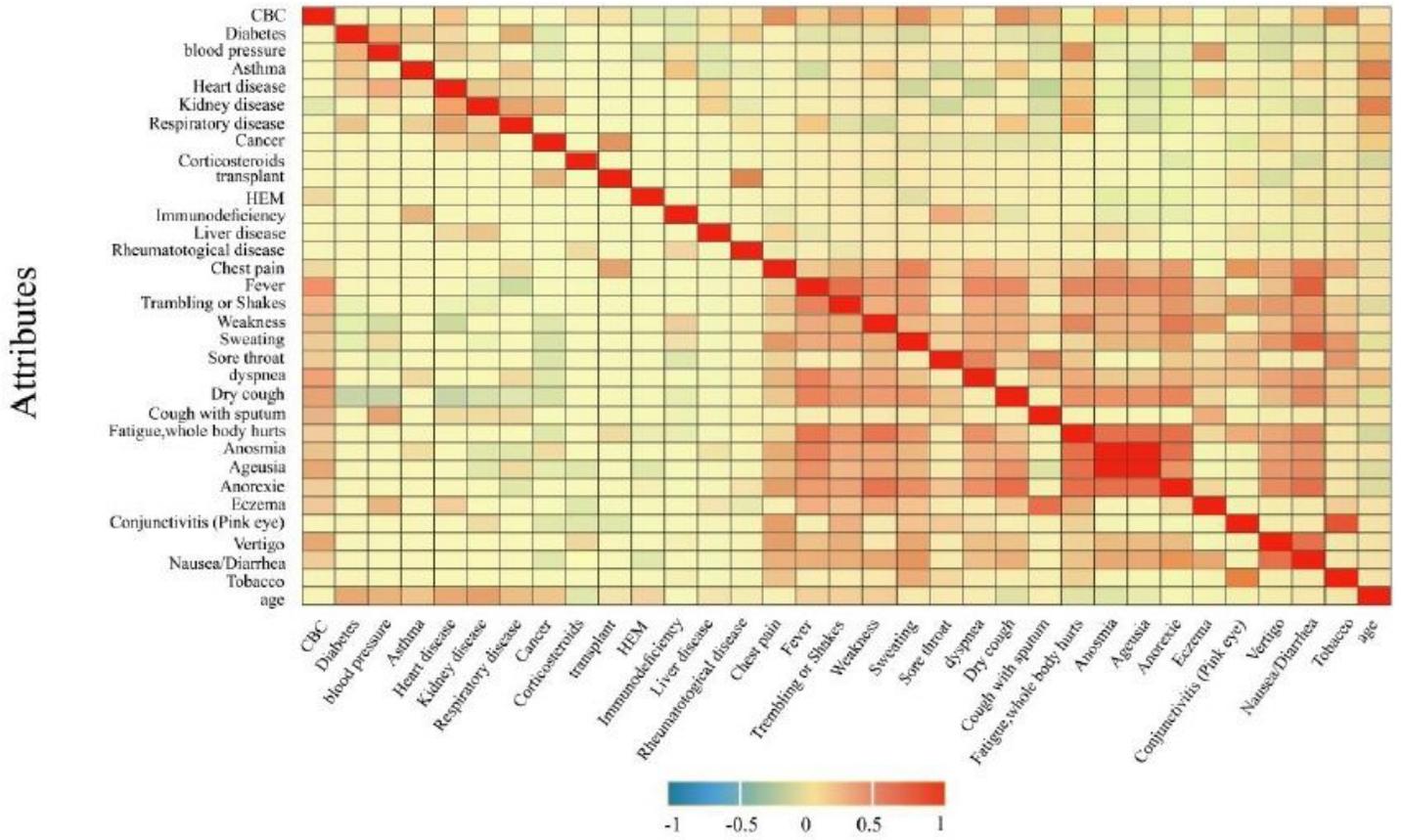


Figure 4

Correlation between dataset features

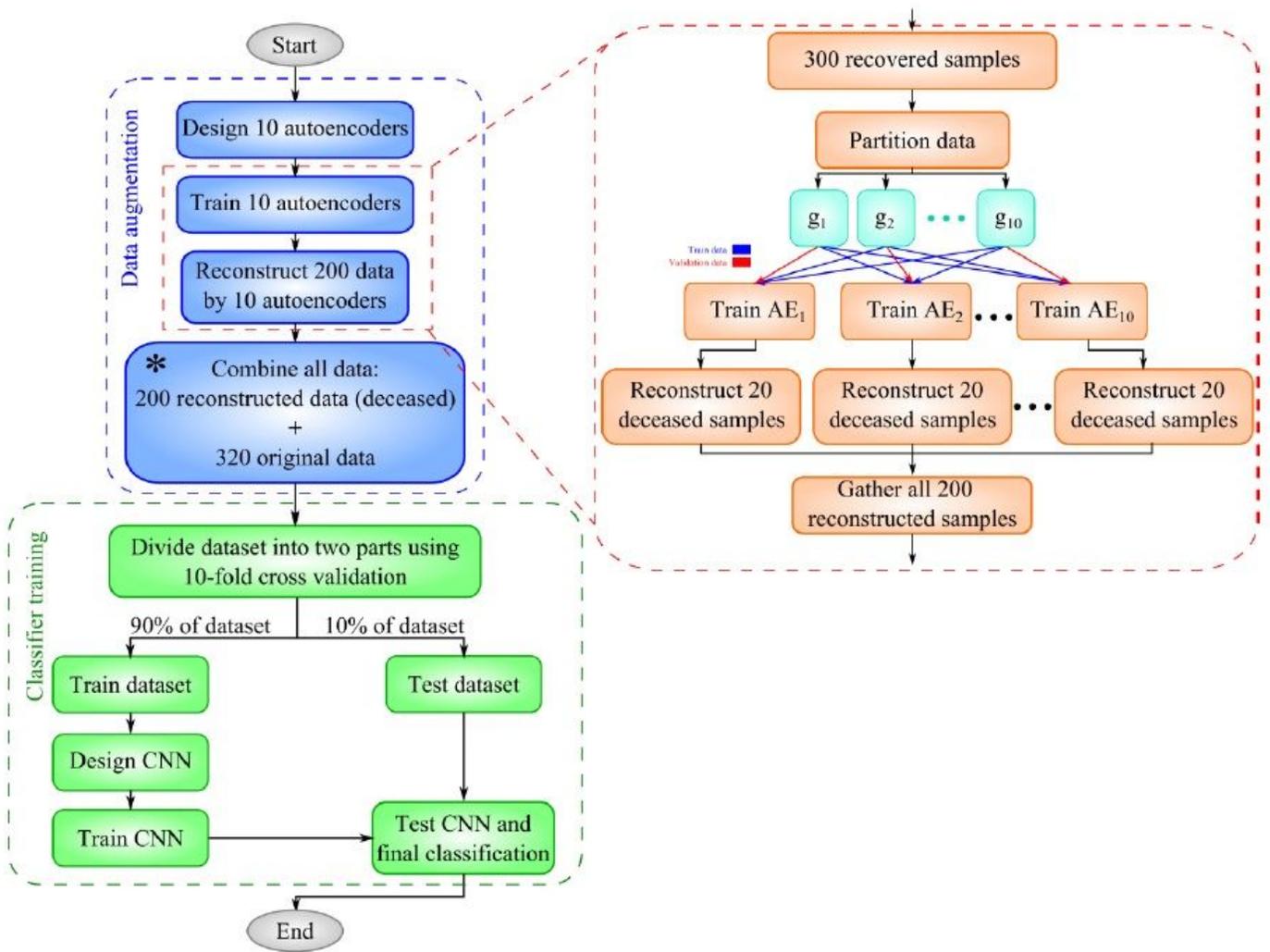


Figure 5

The steps for implementing the proposed method.

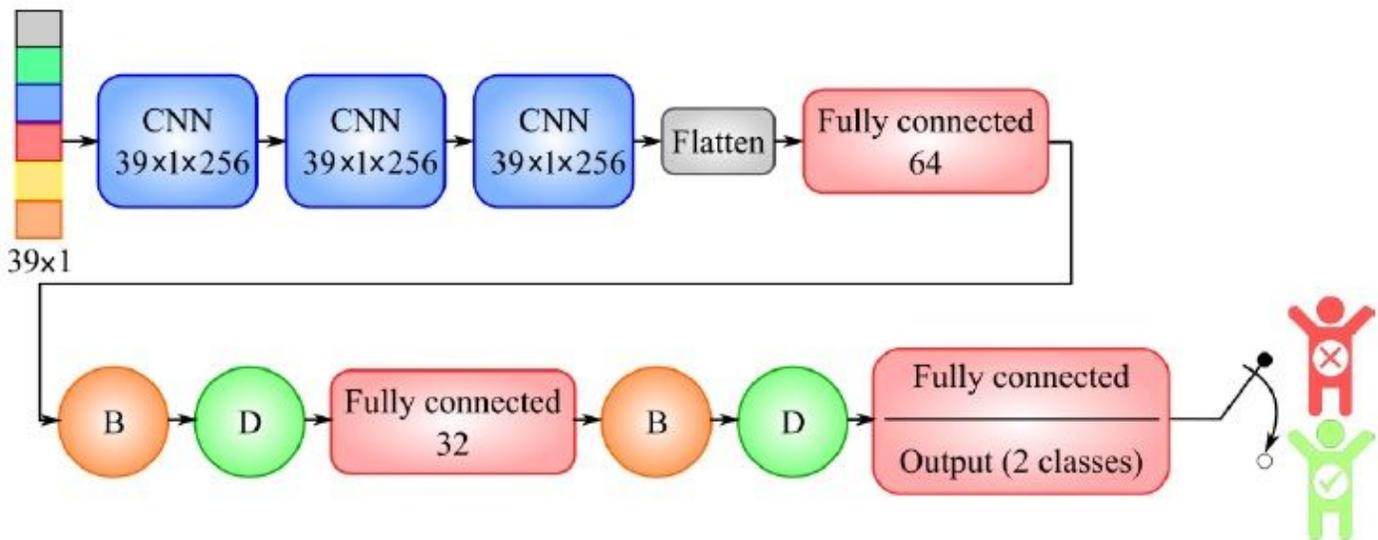


Figure 6

Implementation details of CNN: circles with "B" letter represent batch normalization layers and circles with "D" letter represent dropout layers with probability 0.5.

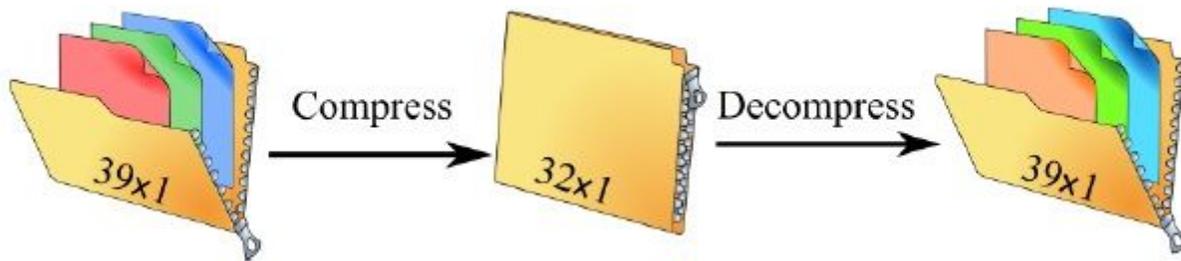


Figure 7

Implementation details of Autoencoder

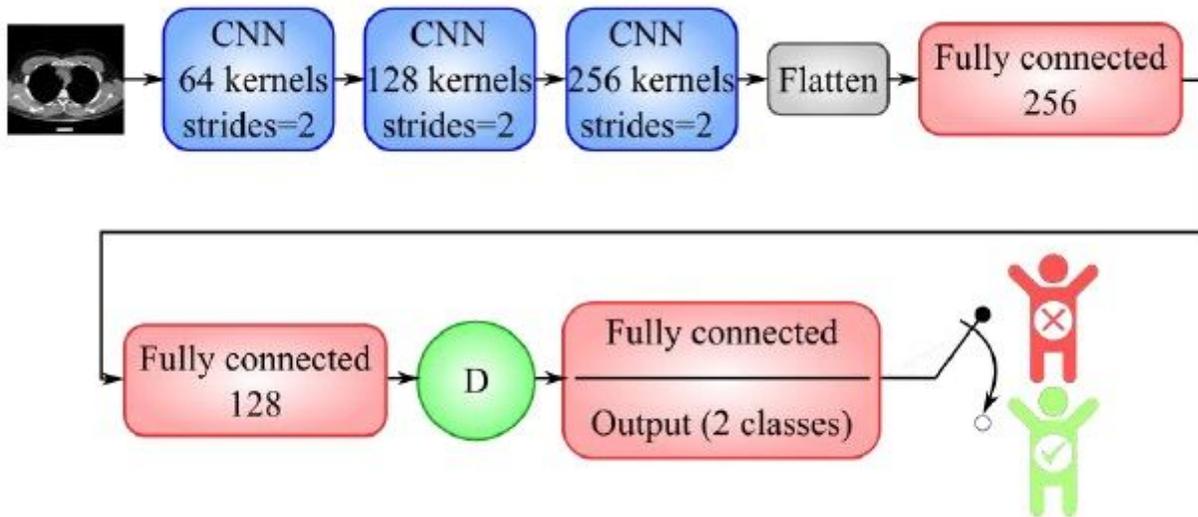


Figure 8

Implementation details of the CNN which is trained on CT images.

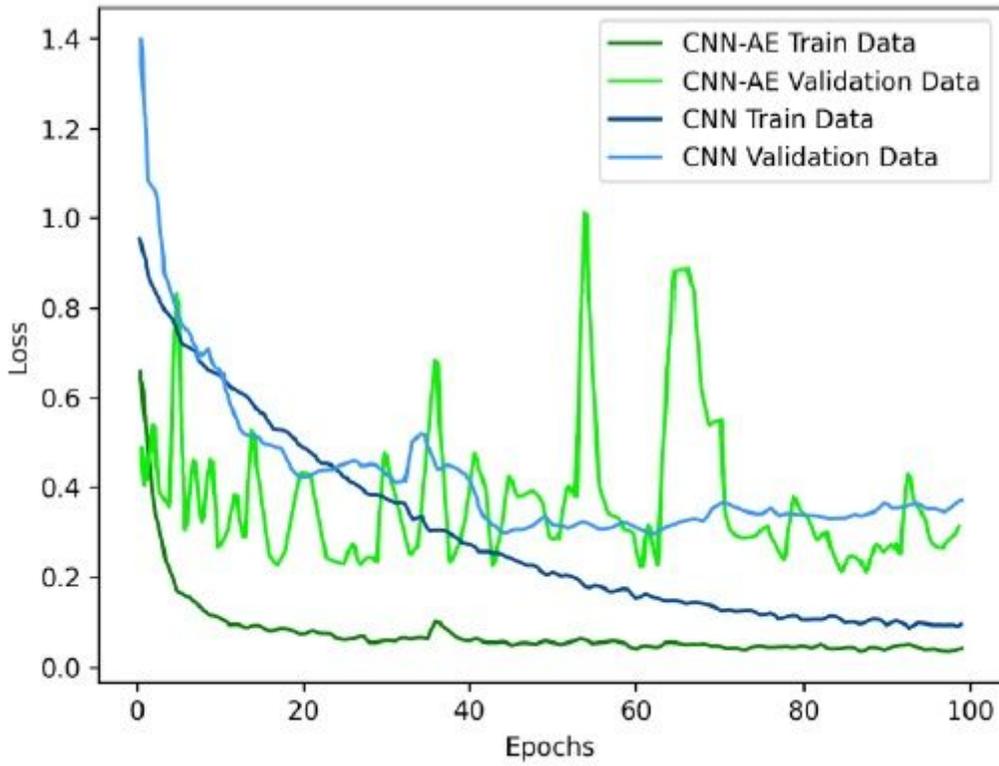


Figure 9

Loss plots of CNN and CNN-AE methods during training on our dataset

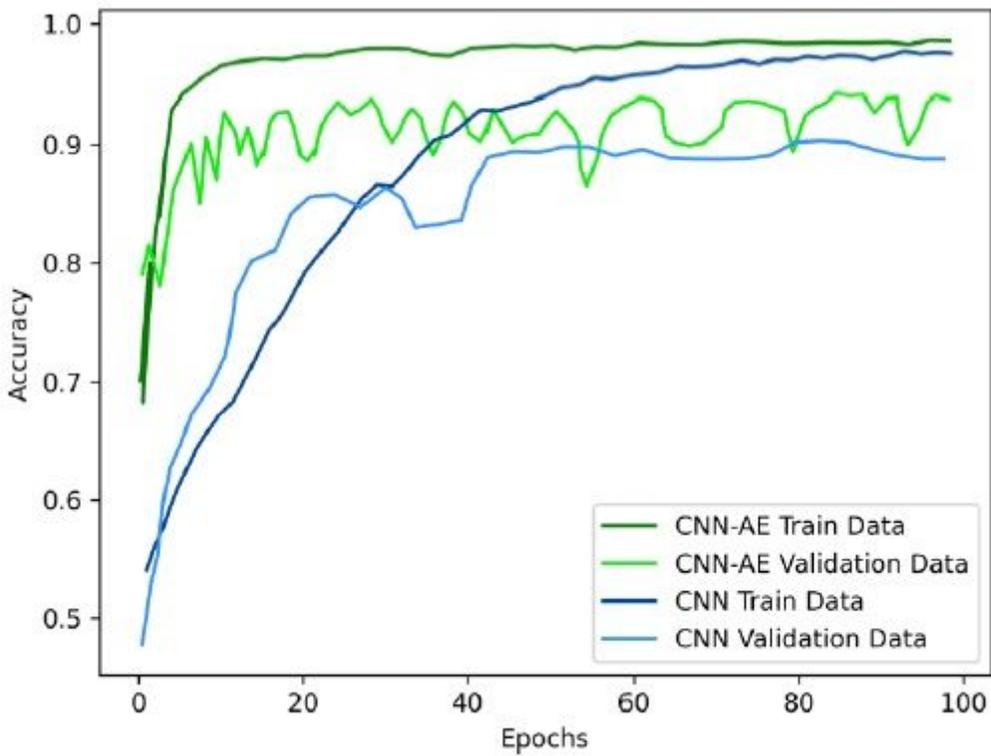


Figure 10

Accuracy plots of CNN and CNN-AE methods during training on our dataset

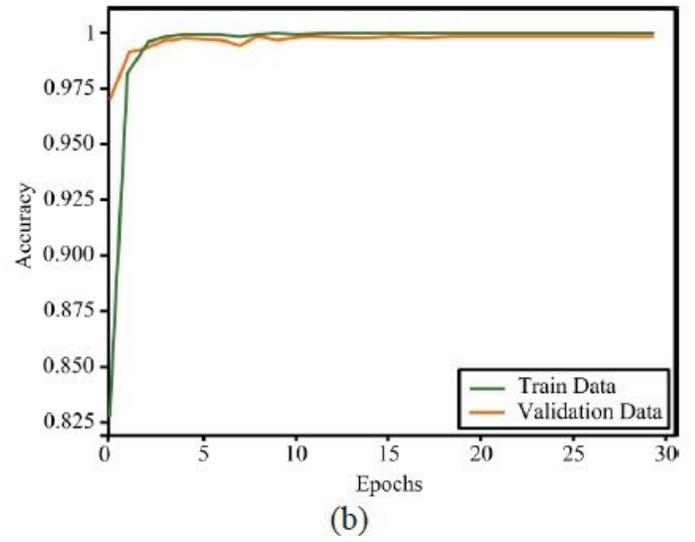
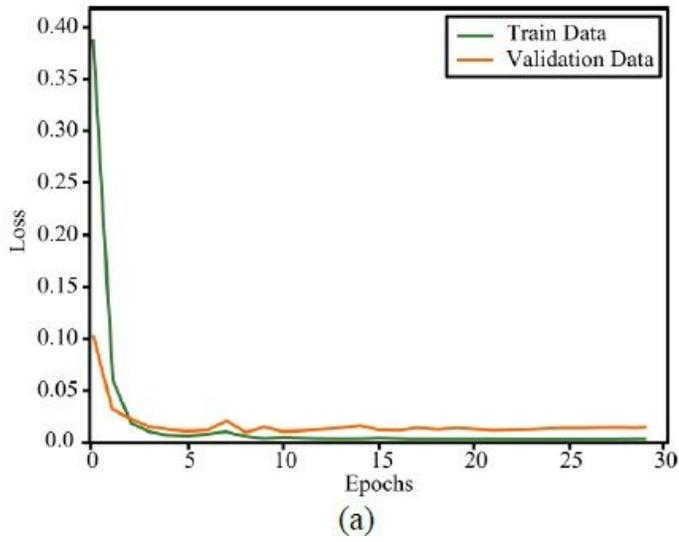


Figure 11

CNN trained on CT images: (a) loss plot, (b) accuracy plot