



Variational Autoencoder Based Imbalanced COVID-19 Detection Using Chest X-Ray Images

Sankhadeep Chatterjee¹ · Soumyajit Maity² · Mayukh Bhattacharjee² · Soumen Banerjee³ · Asit Kumar Das¹ · Weiping Ding⁴

Received: 3 December 2021 / Accepted: 16 October 2022 / Published online: 19 November 2022
© Ohmsha, Ltd. and Springer Japan KK, part of Springer Nature 2022

Abstract

Early and fast detection of disease is essential for the fight against COVID-19 pandemic. Researchers have focused on developing robust and cost-effective detection methods using Deep learning based chest X-Ray image processing. However, such prediction models are often not well suited to address the challenge of highly imbalanced datasets. The current work is an attempt to address the issue by utilizing unsupervised Variational Auto Encoders (VAEs). Firstly, chest X-Ray images are converted to a latent space by learning the most important features using VAEs. Secondly, a wide range of well established data resampling techniques are used to balance the preexisting imbalanced classes in the latent vector form of the dataset. Finally, the modified dataset in the new feature space is used to train well known classification models to classify chest X-Ray images into three different classes viz., "COVID-19", "Pneumonia", and "Normal". In order to capture the quality of resampling methods, 10-folds cross validation technique is applied on the dataset. Extensive experimental analysis have been carried out and results so obtained indicate significant improvement in COVID-19 detection using the proposed VAE based method. Furthermore, the ingenuity of the results have been established by performing Wilcoxon rank test with 95% level of significance.

Keywords COVID-19 · Class imbalance · Variational autoencoder · Oversampling · Undersampling

1 Introduction

The novel Coronavirus disease (COVID-19) was first reported in late 2019, from the city of Wuhan situated in the Hubei province of China [1]. The disease is the outcome of the SARS-COV-2 (Severe Acute Respiratory Syndrome Coronavirus-2)

✉ Sankhadeep Chatterjee
chatterjeesankhadeep.cu@gmail.com

Extended author information available on the last page of the article

virus. The fatality of COVID-19 was highlighted in the early months of 2020 when the highly transmissible nature of the virus was observed across various nations [2, 3]. As of August 12, 2021, the total number of COVID-19 cases reported across the world sums up to over 200 million with a total of 4 million deaths [4]. The common manifestations of the disease primarily constitute cough, fever, headache, and also dyspnea for severe cases. Experts suggest that the primary modes of virus transmission are respiratory secretions of a COVID-19 infected individual. The manually operated Reverse Transcription-Polymerase Chain Reaction (RT-PCR) test for detecting the disease is performed on respiratory specimens and requires a considerable amount of time and money. Moreover, the test is further complex owing to the limited availability of test kits. Apart from that, medical experts rely on X-rays for COVID-19 detection due to their effectiveness in determining the disease at an early stage. In addition to that, the tools required to carry out X-Rays are sufficiently available in most clinics [5]. In contrast to bacterial pneumonia, COVID-19 can produce lung air-space in multiple lobes of the lung [6]. Chest X-Rays can provide a clear view of these opacities and help experts in diagnosis of COVID-19.

1.1 Objective

Analysis using Chest X-rays is time-efficient but requires a medical expert to interpret the results. The surge in the number of COVID-19 cases requires certain improvements in the current medical infrastructure concerning the outbreak. The daily increase in the number of cases demands a good strength of medical experts to attend each patient in a more convenient way. Moreover, the current situation demands a much faster way to perform the diagnosis. To overcome this challenge, automatic intelligent systems are required owing to their flexibility and time efficiency.

Biomedical Images [7, 8] have been well-studied in the field of Digital Image Processing and Machine Learning. Although X-Ray images are known for their attainability and simplicity, biomedical image analysis has proven to be a challenging problem due to the images' blurred boundary contours, different sizes, various shapes and uneven density [9]. Another important factor that leads to wrongful diagnosis is that in the premature stage, COVID-19 symptoms can look similar to those of other viral infections, for instance RSV pneumonia. Therefore it is incontrovertible that the significance of correct diagnosis is very crucial in COVID-19.

1.2 Motivation

Data-driven feature learning through deep learning outperforms traditional approaches such as manual human-intervention. The success of convolutional neural network in medical images is mainly due to its ability to learn features automatically from domain-specific images [10]. However, CNNs are naturally suited for labeled data. Unsupervised learning approaches are mainly used in auto-encoders which utilize unlabeled data. At the same time, unlabeled data is relatively easier to acquire in biomedical images [11]. Convolutional Autoencoders, therefore have

been widely employed for biomedical image analysis. In [12] Post-denoising autoencoders (DAE) is proposed, which improves erroneous and noisy segmentation masks to a feasible space after experimenting in binary and multi-label segmentation of chest X-ray and cardiac magnetic resonance images [12]. An experiment over a colon cancer dataset to propose a Joint Triplet Autoencoder Network (JTANet) by facilitating triplet learning in autoencoder framework is reported in [13]. In [14], the authors prevent the model from learning an identity mapping by introducing skip-connections to Autoencoder-based approaches for Unsupervised Anomaly Detection (UAD) in brain MRI.

Using autoencoders for biomedical images would be ideal except for a few data distribution problems like the class imbalance one [15]. The class imbalance problem results in performance degradation of standard machine learning models due to the imbalance in number of samples in the classes, training the models. In other words, the majority class has significantly larger number of samples than that of the minority class, leading to an aberration in training of standard models. Imbalanced data has been a major problem while using data driven machine learning models in numerous domains including biomedical datasets. Authors in [16] incorporate methods for dimensionality reduction and alleviating the class imbalance problem in genomic datasets. Liu et al. review data resampling and feature selection methods for imbalanced biomedical datasets [17]. In order to overcome the class imbalance problem, many approaches have been proposed. DBSCAN clustering algorithm is used to filter out the noisy majority class samples, and therefore, reduce the imbalance ratio [18]. In [19], the authors propose a boosting aided adaptive cluster-based undersampling technique to revoke learning from insignificant majority class noise.

A Tomek link exists between a pair of samples if they belong to different classes as well as are each other's nearest neighbors. In order to balance the dataset, the majority class samples that are a part of a Tomek link are removed. The Edited Nearest Neighbor (ENN) takes on an approach where the majority class sample gets deleted if k -nearest neighbors have different classes. Neighborhood Cleaning Rule (NCL) adopts a similar approach to the Edited Nearest Neighbor where the incorrectly classified samples belonging to the majority class are removed if they consists of three nearest neighbors of a minority class sample [20]. The class imbalance problem in medical images have been tackled by using the Neighborhood Cleaning Rule in [21]. In [22] the authors proposed a K-means clustering-based undersampling algorithm and tested it utilizing two small-scale breast cancer datasets.

Although undersampling might increase the accuracy of predictions, it also leaves the risk of unavailing beneficial data samples, which might be costly otherwise. Therefore, other alternatives for solving the class imbalance problem are synthetic sampling methods. A sophisticated oversampling algorithm called Synthetic Minority Over-sampling Technique (SMOTE) has been very popular since its publication [23]. It proposes an approach where minority samples are synthetically added randomly along a line drawn from minority samples to its k nearest neighbors in the dataset. In [24], the authors in order to boost the accuracy of cardiovascular patient's survivor prediction and also to handle class imbalance problem, employed SMOTE. This algorithm has been extensively used to balance imbalanced medical datasets and thus improve the predictability of the classifiers in various applications

[25–27]. In [27], the authors predict lung cancer from X-ray images while mitigating the problem of class imbalance by utilizing oversampling algorithms such as ADASYN. Borderline SMOTE, an enhancement of SMOTE, is an auxiliary oversampling approach also applied in major biomedical image analysis. In [28], the authors used Borderline-SMOTE on imbalanced medical data of lung cancer metastasis to the brain, collected from National Health Insurance Research Database, Taiwan. The imbalance of the medical datasets has been ensured to have a very discernible consistency. Another such disproportion is managed by the SVM SMOTE for the prediction of thyroid cartilage invasion in [29]. The authors in [30] offer to predict fetal health rate after subjecting the dataset through several oversampling techniques including SMOTE, Borderline SMOTE, ADASYN and SVM SMOTE. In the current article, we explore the autoencoders to create synthetic instances of chest X-Ray images for predicting symptoms of coronavirus.

1.3 Contribution

Owing to successful applications of VAEs in various medical image analysis, it is used in the current study to learn the most important features in order to represent chest X-Ray images into a latent space. To understand the optimal size of latent space vectors, experiments have been carried out by varying the size of latent vectors and observing the performance of classifiers for all such datasets. Besides, the contribution factor of reconstruction loss and latent loss in total loss calculation during VAE training is also analysed. Two different datasets are merged together and used in the current study. The final merged dataset is highly imbalanced with respect to the “COVID-19” class thereby making it extremely hard to be used in training of classifiers. After fixing the parameters of VAE model, the entire dataset has been transformed to latent vector form. In the current work, a balanced testing has been performed to measure the performance of classifiers where 80% of data instances are kept in training set and rest is used for testing. It is to be noted that here resampling techniques are applied to modify the training set only. Therefore, the test set is not modified at any stage. Six different well known classifiers of various types have been used to classify data instances into three categories viz., “COVID-19”, “Pneumonia”, and “Normal” which indicates chest X-Rays having COVID-19, Pneumonia and normal condition respectively. Test phase performance of classifiers are measured in terms of accuracy, precision, recall, and Area Under Curve (AUC) scores. Experimental results have indicated towards significant improvement of classifiers after balancing the dataset using the proposed method. The contributions of the current article can thus be summarized as follows:

1. The imbalanced image classification based COVID-19 detection using chest X-Ray images have been studied.
2. Variational Autoencoder based model has been proposed to extract the most important features from the input raw images. The resampling strategies are then applied on latent space to mitigate the class imbalance problem.

3. Oversampling, Undersampling and Hybrid methods of class imbalance algorithms have been studied rigorously to overcome problem of class biased latent vectors in detection of COVID-19 using chest X-Ray images.

The rest of the manuscript is broadly arranged as follows: Sect. 2 illustrates the background of variational autoencoders, synthetic resampling methods and latent space resampling techniques in detail. Subsequently, in Sect. 3, a detail experimental results have been reported including the performance analysis of the classifiers on different metrics. The parametric study, visualization of generated samples using t-SNE [31] have been reported as well. Finally, Sect. 4 consists of the concluding remarks and future works for the current proposed study.

2 Proposed Method

2.1 Autoencoders

An autoencoder, also termed as autoassociator [32] are basic learning circuits that aims to convert the inputs into outputs with as little distortion as possible. The concept of autoencoders were first introduced by Hinton and Rumelhart et al. to resolve the issue of “backpropagation without a teacher”, by utilizing the input data as the teacher. The main goal of an autoencoder is to learn in an unsupervised manner an “informative” representation [33] of the data that can be utilized for various purposes such as clustering. Autoencoders, along with Hebbian learning principles [34], provide one of the ideal fundamental models for unsupervised learning and addresses the secret of how synaptic changes prompted by neighborhood biochemical occasions can be facilitated in a self coordinated way to produce intelligent behavior. The representative capacity of a shallow neural network with only one hidden layer is extremely limited [35]. To minimize the additional time consumption and enhance the representation of the features, the greedy layer-wise training of the deep neural networks proves to be a better choice than a single-layered shallow neural network [36]. A deep neural architecture can be initiated by assembling multiple autoencoders to enhance the representative ability of learned instances from the input data. A single autoencoder comprises of three parts, the input layer, the encoding layer, and the decoding layer. For every instance (y^i) in the training dataset $D = \{(y)^1, (y)^2, \dots, (y)^N\}$ where (y^i) and N represents the input vector of the i th sample and the number of instances respectively. The encoding layer can be represented as:

$$f(y) = s_e(W_e y + b_e), \quad (1)$$

where $s_e(\cdot)$, W_e , and b_e represents the activation function, the weight matrix, and the bias vector of the encoding layer, respectively. In the same manner, the decoding layer can be defined as:

$$g(y) = s_d(W_d y + b_d), \quad (2)$$

where $s_d(\cdot)$, W_d , and b_d denotes the activation function, the weight matrix, and the bias vector of the decoding layer, respectively. Hence, the output of the autoencoder for the instances can be defined as:

$$z = g(f(y)). \quad (3)$$

2.2 Variational Autoencoders

The Variational Autoencoders (VAE) have proved to be a major improvement while dealing with the feature representation capability [37]. The VAE are generative models that is based on the Variational Bayes Inference and combine deep neural networks which aim to regulate the encoding pattern during training so that the latent space has good properties to enable the process of the instance generation using a probabilistic distribution. The VAE has had many applications in the domain of image synthesis [38], video synthesis [39], and unsupervised feature extraction [40] respectively. As described in [41], numerous data points with similar characteristics to the input can be created by sampling different points from the latent space and decoding them for use in downstream tasks. However, a constraint is imposed on learning the latent space to store the latent attribute as a probability distribution in order to generate new high-quality data points.

in the VAE model, the input data is as follows:

$$p_\theta(x|z) = f(x; z, \theta) \quad p(z) = N(z|0, I), \quad (4)$$

where, f is a posterior probability function that uses deep neural network to perform a non-linear transformation with z parameters. The exact computation of the posterior $p_\theta(z|x)$ in this model is not mathematically feasible. Instead, a distribution $q_\phi(z|x)$ [37] is used to approximate the true posterior probability. This inference network $q_\phi(z|x)$ is parameterized as a multivariate normal distribution as shown below:

$$q_\phi(z|x) = N\left(z | \mu_\phi(x), \text{diag}\left(\sigma_\phi^2(x)\right)\right), \quad (5)$$

where, both $\sigma_\phi^2(x)$ and $\mu_\phi(x)$ represent the vectors variance and means respectively.

As discussed previously, the VAE helps to accomplish two things. Firstly, it allows to encode an image x to a latent vector, represented by, $z = \text{Encoder}(x) \sim q(z|x)$. Subsequent to which a decoder network is used to decode the latent vector z back to the image which is equivalent to the original fed image represented as, $\bar{x} = \text{Decoder}(z) \sim p(x|z)$. Because the marginal likelihood is intractable, a VAE's objective function is the variational lowerbound of the marginal likelihood of data. The marginal likelihood of the singular data points can be represented as:

$$\log p_\theta(x^{(i)}) = D_{\text{KL}}(q_\phi(z|x) || p_\theta(z)) + \mathcal{L}(\theta, \phi, x^{(i)}). \quad (6)$$

The first term on the right hand side (RHS) of Eq. (6) is the *Kullback–Leibler* (KL) divergence of approximate posterior and anterior. The second term on the RHS of Eq. (6) is the lower bound variance for the marginal probability of data point i . As the KL divergence is always greater than 0, Eq. (6) can be re-written as:

$$\log p_{\theta}(x^{(i)}) \geq \mathcal{L}(\theta, \phi, x^{(i)}) \tag{7}$$

$$= E_{q_{\phi}(z|x^{(i)})}[-\log q_{\phi}(z|x) + \log p_{\theta}(x|z)] \tag{8}$$

$$= -D_{KL}(q_{\phi}(z|x^{(i)})||p_{\theta}(z)) + E_{q_{\phi}(z|x^{(i)})}[\log p_{\theta}(x|z)]. \tag{9}$$

In the equations above, $p_{\theta}(x|z)$ is the probability of the data x given the latent variable z . The KL divergence between the estimated posterior and the prior of the latent variable z is represented in Eq. (9). As a regularisation term, this forces the posterior distribution to be identical to the previous distribution. The latter part of Eq. (9) can be described in terms of reconstruction of x using the posterior distribution $q_{\phi}(z|x)$ and the probability $p_{\theta}(x|z)$. The primary difference between an Autoencoder and a VAE is that [42] an Autoencoder is a deterministic discriminative model that does not possess a probabilistic approach, while VAE is a stochastic generative representation that provides calibrated probabilities.

The VAE is trained through Backpropagation method. The second term of Eq. (9) is calculated by using Monte Carlo methods. However, the traditional Monte Carlo gradients used to optimize the variational lower bounds are said to have quite high variance and are unsuitable for use [43]. The VAE mitigates this by employing a reparameterization technique that employs a random variable from a normal distribution rather than from the original distribution. The random variable: $z \sim q_{\phi}(z|x)$ is reparameterized with the help of a deterministic transformation $h_{\phi}(\epsilon, x)$ where ϵ denotes a standard distribution.

$$\tilde{z} = h_{\phi}(\epsilon, x). \tag{10}$$

It is ensured by the reparameterization technique that \tilde{z} follows the distribution of $q_{\phi}(z|x)$. This technique is found to be more stable than using the Monte Carlo gradient method directly. The training algorithm for a VAE has been discussed in Algorithm 1.

Algorithm 1 Training a Variational Autoencoder (VAE)**Require:** Data x^1, \dots, x^N **Ensure:** probabilistic decoder P_ϕ , probabilistic encoder P_θ $\theta, \phi \leftarrow \text{Initialize}$ **repeat** **for** $k \leftarrow 1$ to N **do** Draw Samples S from $\epsilon \sim \mathcal{N}(0, 1)$ $z^{(k,s)} = h_\phi(\epsilon^{(k)}, x^{(k)})$ **end for** $E = \sum_{k=1}^N -D_{KL}(q_\phi(z|x^{(k)})||p_\theta(z)) + \frac{1}{S} \sum_{s=1}^S (\log p_\theta(x^{(k)}|z^{(k,s)}))$ $\phi, \theta \quad \triangleright$ Update the parameters using Stochastic Gradient Descent**until** The parameters ϕ, θ converges

2.3 Synthetic Resampling

General Machine Learning models presume that the number of instances are roughly similar for all classes taken into consideration. But, in different real-life domains, such as analyzing cancer malignancy grading [44], sentiment analysis [45], crime prediction [46], and social media sarcasm detection [47] the distribution of classes are skewed as the instances considered in those classes appear more often. This develops a biasness of the learning algorithms for the frequently occurring classes also known as the majority class, leading to performance degradation of the learning algorithm [48]. Simultaneously, the rarely occurring class also known as the minority class is highly important from the outlook of data mining approach [49], as it may consist of important data.

Therefore, the need to balance class distribution by using different resampling technique arises. Synthetic Resampling refers to the use of oversampling, undersampling, and combining both the oversampling-undersampling techniques respectively to mitigate the problem of skewed distribution of the target instances. As discussed in [50], the approach to tackle the problem of imbalanced data can be distinguished into three different categories:

- *Data-Level methods* that primarily focus on improving the performance of the learning algorithms by adding or removing the data instances
- *Algorithm-Level methods* that aims to modify the prevalent learning algorithms to weaken the affinity towards the majority class.
- *Hybrid methods* which combine both the previously mentioned types.

However, for the current study, only the *Data-Level methods* are considered and the different techniques are discussed extensively as follow:

2.3.1 Oversampling Techniques

The most well known oversampling approach was proposed in [23] called Synthetic Minority Oversampling TEchnique (SMOTE). The SMOTE algorithm produces new instances by interpolation of the minority class instances. The process involves selecting instances in the feature space that are close together, drawing a line in the feature space between the examples, and creating a new instance at a point along the same line. Several drawbacks of the SMOTE algorithm has been discussed in [51]. An Adaptive synthetic sampling approach for imbalanced learning (ADASYN) [52] utilizes a modified version of the SMOTE algorithm. It primarily improves the learning in two separate ways: (1) Shifting the categorization decision border toward the difficult samples in an adaptive manner, and (2) Eliminating the bias caused by the disparity in class. For the present article, we also have used two other algorithms viz., Borderline-SMOTE and SVM-SMOTE are also used. The Borderline-SMOTE algorithm [53] generates the synthetic instances between the boundary region of the two different classes. The SVM-SMOTE algorithm [54], which is another variation of the Borderline-SMOTE algorithm approximates the class boundary area with the help of support vector machines (SVM), following the training of SVM classifiers on the official training dataset. The synthetic instances are then randomly generated along the boundary line connecting every support vector of the minority class along and the number of nearest neighbors.

2.3.2 Undersampling Techniques

The two most common undersampling method, the Random Undersampling technique and Condensed Nearest Neighbors (CNN) focus on balancing the class distribution by randomly deleting the instances of the majority class. While this lessens the training time for the models, it also may lead to the removal of highly significant data points. [55]. The algorithms that consist of data cleansing procedures are used in more informed undersampling strategies. The Edited Nearest Neighbor (ENN) and Tomek Link removals are two such algorithms. The Tomek Link removal technique proposed by Tomek aims to enhance the CNN technique by putting forward the rule which locates border point pairs that are involved in the development of (piecewise-linear) borders. These pairs could be used to construct increasingly simpler descriptions of approximations of the original fully stated boundaries. The ENN technique proposed by Wilson states that for every data point x in the training set, its $k = 3$ nearest neighbors are calculated. If x belongs to a majority class and is misapprehended by its three nearest neighbors, then x is deleted from the train set. On the other hand, if x belongs to minority class and is misapprehended by its three nearest neighbors, then the majority instances among the neighbors of that particular instance x are removed. For the present work two other algorithms viz., Neighborhood Cleaning Rule (NHC), and ALL-KNN are considered as well. The NHC Rule technique [20] combines both CNN and ENN techniques. The CNN rule helps to delete the redundant synthetic instances and the ENN aids to remove the noise from the training data. The aim of this algorithm is to focus more on the quality

of data instances that are retained in majority class than that of balancing the class distribution.

2.3.3 Combination of Over- and Under-Sampling Techniques

The combination of over- and under-sampling techniques incorporates the use of oversampling of the minority instances and also undersampling the majority instances at the same time. Batista et al. studied the combination of the oversampling algorithm SMOTE with the two different undersampling algorithms. In the case of *SMOTE – Tomek*, Batista et al. [56] only the instances of the majority class were removed, since, the minority class samples occurred very rarely to be deleted. This resulted in the reduction of false-negative predictions for a binary classification task. In final method, that is *SMOTE-ENN*, [57] the authors state that the ENN in this case, is more intrusive at undersampling the majority class than Tomek-Link, that incorporates a more thorough removal of the instances from both the majority and minority classes.

2.4 Latent Space Resampling

In the current work latent space vectors are resampled to mitigate imbalanced class problem in the dataset. The proposed approach is depicted in Fig. 1. Firstly, a VAE is trained to reconstruct the input images. During this phase the labels of all training samples are ignored. This unsupervised phase of training enables the encoder to produce efficient latent space representation of input images. The proposed VAE architecture contains convolution layers, max pooling layers, batch normalization and flatten layers. Input layer accepts 96×96 images. Input layer is followed by 2D-Convolution layer with 12 filters of size 3×3 , max pooling layer with filter size 2×2 , and batch normalization. This structure is repeated two times with filter sizes of 3×3 and 2×2 respectively. This is followed by a flatten layer which converts the final activation to a vector of size 12. For all intermediate layers LeakyRELU activation function has been used. The decoder network is an exact opposite replica of encoder. The architecture has been depicted in Fig. 2.

After training is complete, the encoder is used to convert all training set images of the dataset to their latent vector form. This process has been discussed in Algorithm 2. However, this conversion inherits the data imbalance problem which is

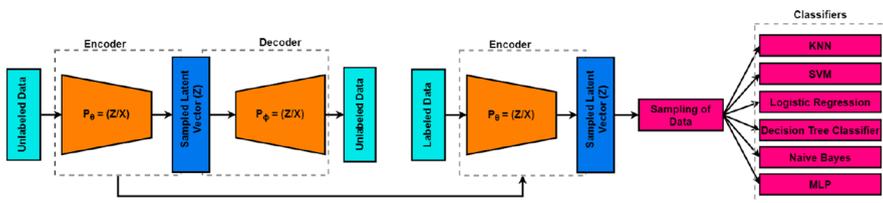


Fig. 1 Proposed framework with Variational Autoencoder Training and classification of latent vectors through different classifiers

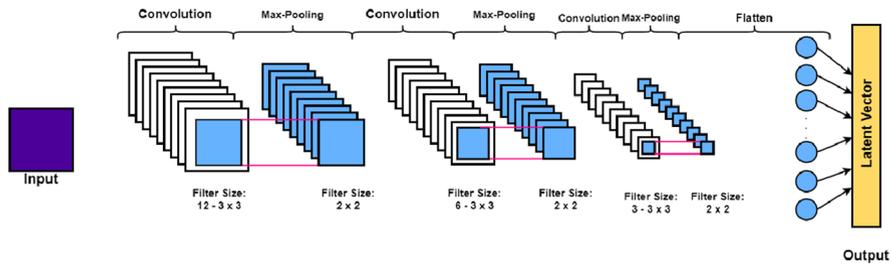


Fig. 2 Proposed encoder architecture

already present in original dataset. Hence, the modified dataset of labelled latent vectors are still imbalanced. Therefore, in the next phase, resampling techniques are used to resample majority/minority latent vectors in the modified dataset. It can be noted that splitting the dataset into training and testing sets after oversampling may lead to data leakage [58]. Thus, a set of randomly selected balanced data samples are kept separately before resampling the modified dataset for testing phase. Rest of the dataset is balanced using resampling techniques. The latent vector representations of test images are obtained by feeding them to the already trained encoder part of the VAE. Next, the latent vectors are classified by using trained classifiers. In the current study, well established oversampling algorithms viz., SMOTE, ADASYN, Borderline-SMOTE, SVM-SMOTE, undersampling techniques viz., Cluster Centroid, Tomek Links, Neighborhood Cleaning Rule, ENN, All-KNN, along with hybrid methods like SMOTE-ENN and SMOTE-Tomek are used to resample the data instances.

Algorithm 2 Proposed Architecture

Require: Image Dataset I_0, \dots, I_N

Ensure: Performance of the Classifiers

Train the VAE \triangleright The training algorithm for the VAE has been discussed above

Encoder $\xrightarrow{\text{Output}}$ Latent Vector (L)

Resample (L) $\xrightarrow{\text{Output}}$ L_R

Train Classifiers $\leftarrow L_R$

3 Results and Discussion

In this section, the different performance scores of various validation classifiers are evaluated on original imbalance data and synthetic data obtained by applying sampling methods on original imbalance data latent form. A description of dataset is provided followed by the implementation of the experiment. The models are simulated with NVIDIA GeForce GTX 1650, Ryzen 5-3550H, 8 GB RAM, Windows 10 Home 21H1, and TensorFlow 2.5.0.

3.1 Experimental Setup

In the current study, two datasets discussed in [59] are merged to form a single dataset which contains three types of images viz., “COVID-19”, “Pneumonia” and “Normal” standing for COVID-19 affected, Pneumonia affected and Normal chest X-ray images respectively. It has a total of 137 images of COVID-19 infected chest X-Ray, 4,343 Pneumonia affected chest X-Ray and many Normal chest X-Ray where each image is of 96×96 in size. Most of the images are 1-channel grayscale image with few images being 3-channel (RGB). Figure 3 depicts few examples of the dataset. Evidently, the dataset is highly imbalanced which affects the classifier performance.

In order to balance the dataset, firstly, the VAE is trained by feeding training samples obtained from the dataset. During unsupervised training of VAE, RELU activation function and ADAM optimizer are used to take leverage of nonlinear and non-saturating characteristics. The value of learning rate is set to 0.001. All input images are normalized to $[0, 1]$ before using them to train VAE. The trained VAE is then

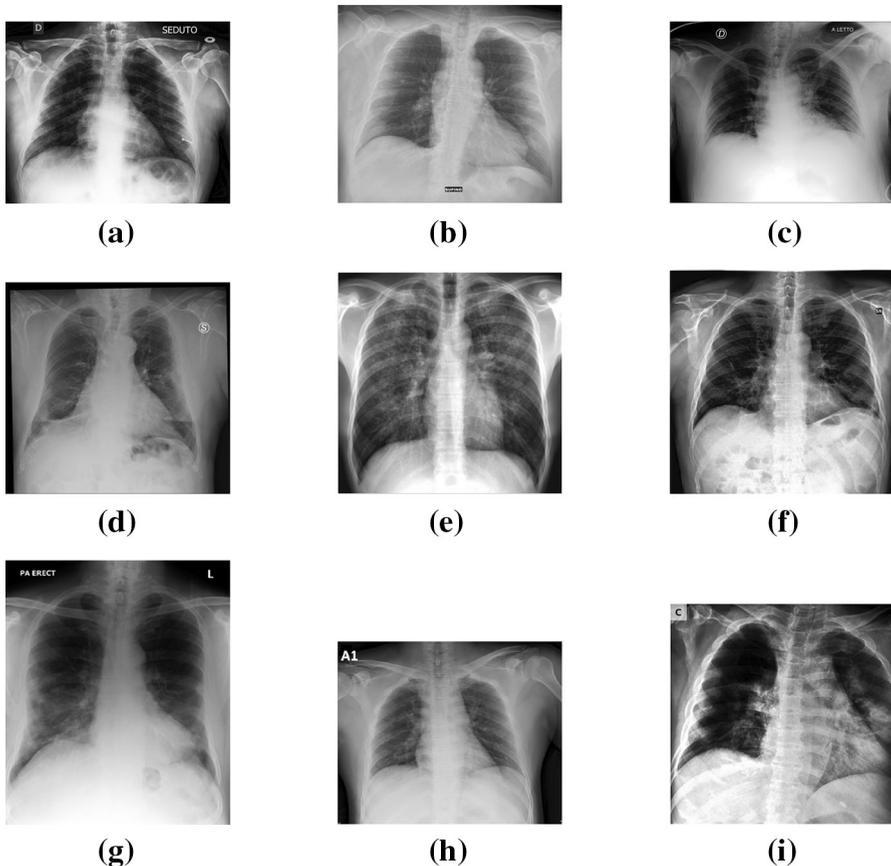


Fig. 3 Chest X-Ray images of COVID-19, Pneumonia and Normal patients

used to bring the input images down to a latent vector of size 12. However, even after transforming the images into latent vectors it remains imbalanced.

In order to cope with the class imbalance problem, few well known data level algorithms are used. Resampling methods, including Over-sampling (i.e., SMOTE, ADASYN, Borderline SMOTE, SVM SMOTE), Under-sampling (i.e., Cluster Centroid, Tomek Links, Edited Nearest Neighbor, Neighborhood Cleaning Rule, All KNN) and the combination of over-sampling and under-sampling (i.e., SMOTE ENN, SMOTE Tomek) are utilized in the current study. The class imbalance algorithms are used to resample the latent vectors which are obtained from previously trained VAE.

To validate the performance improvement of classifiers, 6 different classifiers are applied viz., K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree Classifier (DT), Logistic Regression (LR), Naïve Bayes (NB) and Multilayer Perceptron (MLP). Model hyperparameters are decided by applying Grid Search [60]. The classifiers are implemented using *scikit-learn* package [61]. Optimal hyperparameter values obtained from Grid search is summarised as follows: KNN Classifier is used by setting number of neighbors to 10. SVM is used with the Radial Basis Function (RBF) as kernel. Logistic Regression is implemented by using the 'lbfgs' optimizer with maximum iteration of 100. Decision Tree classifier is applied with 'entropy' splitting criterion. The variance smoothing attribute is set at $1e-09$ for Naïve Bayes classifier. In case of MLP classifier, RELU activation function with ADAM optimizer is used where learning rate is set to 0.001 and maximum iteration is set to 200.

An extensive comparative analysis has been carried out using the set of well known classifiers mentioned earlier. Firstly, the initial dataset is converted to latent vector form by applying the encoder part of the trained VAE. The dataset thus obtained in latent vector form is then split into training and test sets. 80% data is used for training purpose and 20% is kept for testing. In order to avoid data leakage, only training set is resampled by applying data imbalance algorithms. The modified resampled training set is then used to train the classifiers. As no synthetic data is added to the test set or any is removed from the same, the testing phase performance reflects the true performance of classifiers along with the quality of synthetic data generated by resampling techniques. Test phase performance of classifiers have been measured by calculating performance metrics viz., accuracy, precision, recall, and AUC.

3.2 Visualization of Generated Samples

In order to understand the quality of synthetically generated data by using VAE, the synthetically generated data instances obtained after applying the best performing resampling technique SMOTE-ENN, are visualized by using well known t-SNE [31] algorithm. Latent vectors of original and synthetic data instances obtained from the VAE are projected to a two dimensional space using t-SNE. The projected two dimensional vectors are plotted in Fig. 4. This plot reveals that the synthetically generated data instances are close to original data instances thereby indicating towards

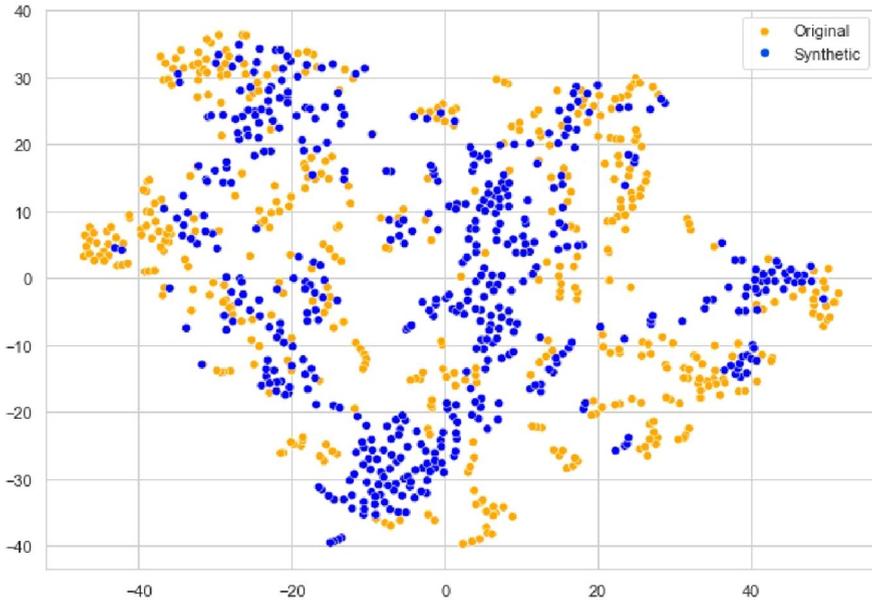


Fig. 4 Visualization of synthetically generated data samples using t-SNE

a good resemblance with original data. Thus, data generated by the VAE to balance the dataset are of good quality and the underlying distribution has been properly approximated.

3.3 Parametric Analysis

Hyperparameters of classifiers, used in the current study, are decided by employing Grid search [60]. In addition to the parameters of classifiers, two parameters of VAE i.e. latent vector size and reconstruction loss parameter (α) are also analyzed. Experiments have been carried out by varying the latent vector size between range [2, 32]. For each latent vector size, all data instances are converted to latent form using the encoder part of the VAE. The classifiers are then trained with resampled latent vectors. For resampling, SMOTE-ENN hybrid model has been used as it reflects best performance among all resampling techniques. Individual classifiers are run 10 times and AUC scores are visualized by box plots. Figure 5 depicts a box plot of AUC score by varying latent vector shape. The plot shows that for filter size 2 and 12, average performance of classifiers in terms of AUC achieves the maximum average as well as maximum highest values. In order to embed reasonable information in latent vectors, the size 12 has been considered in the current study. On the other hand, Fig. 6 depicts a boxplot by varying the α parameter which decides the contribution of reconstruction loss and latent loss in the total loss calculation as follows:

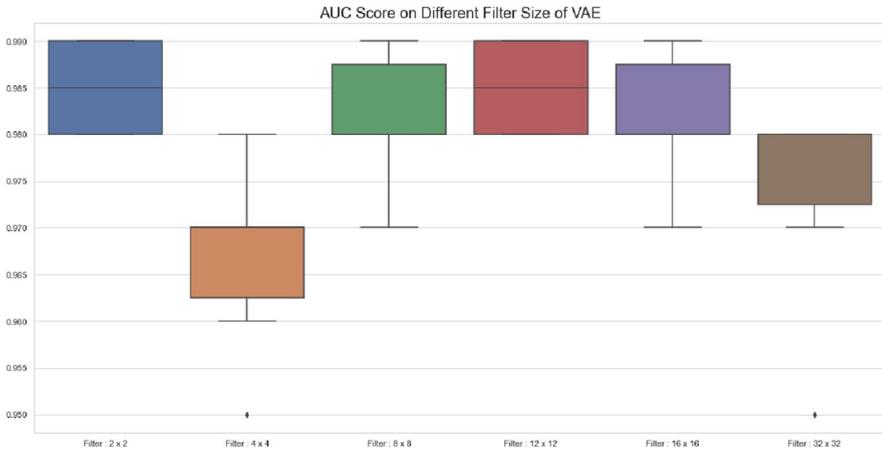


Fig. 5 Performance analysis in terms of Varying Latent vector shape

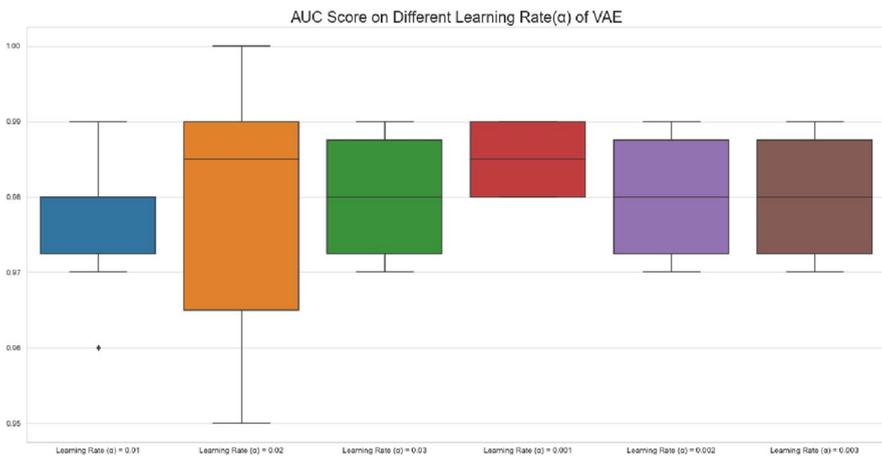


Fig. 6 Performance analysis in terms of Varying α parameter

$$L_t = \alpha * L_r + (1 - \alpha) * L_l, \tag{11}$$

where, L_t denotes total loss, L_r and L_l denotes reconstruction loss and latent loss respectively. Average AUC scores for all classifiers have been recorded and plotted in Figure 6 for α values 0.1, 0.2, 0.3, 0.001, 0.002, and 0.003 respectively. The box plot reveals that highest average and maximum performance are achieved for α value 0.001. Hence, in the current study α is set to 0.001 in all experiments. In addition, it is observed that overall performance of classifiers is improving with decreasing value of α . It indicates that latent vector representation becomes better when priority is given to latent loss component of total loss as described in Eq. (11).

3.4 Comparison of Classifiers

Table 1 reports accuracy of different classifiers. The ‘No Resampling’ column shows accuracy of classifiers while they are trained with original imbalanced dataset in latent vector form. Rest of the table reports the performance of various classifiers which are considered in the current study for other undersampling and oversampling techniques. Performance of all classifiers are found to be extremely poor for original Imbalanced dataset. KNN, SVM, LR, DT, NB, and MLP achieved accuracy of 0.04, 0.04, 0.05, 0.11, 0.06, and 0.07 respectively. After oversampling latent training vectors using SMOTE, performance of classifiers improved significantly and KNN, SVM, LR, DT, NB, and MLP achieved accuracy of 0.84, 0.87, 0.84, 0.83, 0.84, and 0.86 respectively. Other resampling techniques viz., ADASYN, Borderline SMOTE, SVM SMOTE, Cluster Centroid, Tomek Links, NHC Rules, ALL KNN and SMOTE Tomek have reported similar trends of improvement. However, ENN and SMOTE ENN have performed even better. Oversampling latent training vectors with ENN improved accuracy of classifiers above 0.90. Whereas, KNN, SVM, LR, DT, NB, and MLP achieved accuracy of 0.98, 0.98, 0.97, 0.98, 0.96, and 0.98 while minority latent vectors are oversampled using hybrid oversampling technique SMOTE ENN. A 10-fold cross validation is performed in the present experiments and the results so obtained are reported in Table 1 for each validation classifier. Table 1 reveals that accuracy score of SMOTE ENN sampling method is better for all validation classifiers and at least 4% better than most of the other resampling techniques considered in the current study.

As per the present experimental setup, Precision score of each resampling technique is computed for every validation classifier. In Table 2 the ‘Imbalanced’ column reports the precision of classifiers when trained with imbalanced dataset and the rest of the columns represent the precision score which are obtained by oversampling and undersampling the original imbalance data. Performance of all classifiers over Imbalance dataset seems to be unsatisfactory. KNN, SVM, LR, DT, NB and MLP attains Precision score of 0.01, 0.01, 0.04, 0.11, 0.07 and 0.08 respectively. After performing oversampling using SMOTE on original imbalanced dataset, the Performance of validation classifiers improved drastically and KNN, SVM, LR, DT, NB and MLP achieved Precision scores of 0.78, 0.8, 0.8, 0.8, 0.78 and 0.8 respectively. The other resampling techniques viz., ADASYN, Borderline SMOTE, SVM SMOTE, Cluster Centroid, Tomek links, ENN, NHC rule, ALL KNN, SMOTE Tomek also improved the performance to a greater extent. Further, Table 2 reports that ENN and SMOTE ENN resampling techniques have obtained precision score over 0.90 for every validation classifier wherein SMOTE ENN has achieved the best precision score compared to all other sampling techniques. Experimental results have revealed that after resampling the imbalanced latent vectors with SMOTE ENN, classifiers KNN, SVM, LR, DT, NB and MLP have obtained a precision score of 1, 0.98, 0.98, 0.97, 0.98 and 0.98 respectively.

Table 3 presents the Recall score obtained by every validation classifier for each resampling technique. It is evident from Table 3 that the performance of all classifiers over imbalance data is found to be unsatisfactory as revealed by previous performance metrics as well. KNN, SVM, LR, DT, NB and MLP have

Table 1 Performance comparison in terms of accuracy

	No resam- pling	SMOTE	ADASYN	Bor- derline SMOTE	SVM SMOTE	Cluster cen- troid	Tomek Links	ENN	NHC Rule	ALL KNN	SMOTE ENN	SMOTE Tomek
KNN	0.04	0.84	0.84	0.84	0.83	0.84	0.86	0.93	0.91	0.9	0.98	0.84
SVM	0.04	0.87	0.86	0.87	0.87	0.87	0.88	0.94	0.91	0.92	0.98	0.88
LR	0.05	0.84	0.83	0.84	0.83	0.84	0.85	0.93	0.9	0.9	0.97	0.85
DT	0.11	0.83	0.82	0.84	0.82	0.82	0.83	0.93	0.9	0.9	0.98	0.85
NB	0.06	0.84	0.83	0.83	0.84	0.84	0.84	0.92	0.89	0.89	0.96	0.84
MLP	0.07	0.86	0.85	0.85	0.86	0.85	0.87	0.94	0.9	0.92	0.98	0.86

Table 2 Performance comparison in terms of precision

	No resam- pling	SMOTE	ADASYN	Bor- derline SMOTE	SVM SMOTE	Cluster cent- roid	Tomek links	ENN	NHC Rule	ALL KNN	SMOTE ENN	SMOTE Tomek
KNN	0.01	0.78	0.84	0.85	0.84	0.78	0.87	0.93	0.96	0.9	1	0.88
SVM	0.01	0.8	0.89	0.87	0.87	0.8	0.87	0.94	0.95	0.89	0.98	0.87
LR	0.04	0.8	0.86	0.84	0.85	0.79	0.86	0.93	0.92	0.88	0.98	0.86
DT	0.11	0.8	0.84	0.83	0.88	0.78	0.88	0.92	0.9	0.89	0.97	0.86
NB	0.07	0.78	0.83	0.84	0.84	0.76	0.85	0.93	0.9	0.85	0.98	0.82
MLP	0.08	0.8	0.85	0.84	0.84	0.78	0.88	0.94	0.91	0.9	0.98	0.85

Table 3 Performance comparison in terms of recall

	No resam- pling	SMOTE	ADASYN	Bor- derline SMOTE	SVM SMOTE	Cluster cen- troid	Tomek links	ENN	NHC rule	ALL KNN	SMOTE ENN	SMOTE Tomek
KNN	0.03	0.86	0.85	0.8	0.87	0.82	0.83	0.95	0.88	0.9	0.98	0.9
SVM	0.04	0.86	0.86	0.84	0.86	0.89	0.85	0.94	0.92	0.91	1	0.93
LR	0.05	0.84	0.82	0.84	0.86	0.85	0.85	0.92	0.89	0.91	0.98	0.9
DT	0.11	0.86	0.78	0.84	0.82	0.83	0.8	0.94	0.89	0.87	0.98	0.84
NB	0.07	0.87	0.85	0.86	0.86	0.83	0.86	0.92	0.89	0.86	1	0.91
MLP	0.07	0.86	0.83	0.81	0.84	0.84	0.89	0.95	0.9	0.91	0.97	0.9

attained a recall score of 0.03, 0.04, 0.05, 0.11, 0.07 and 0.07 respectively. After oversampling the imbalanced data using SMOTE, the recall score of every classifier has been found to have improved significantly and KNN, SVM, LR, DT, NB and MLP have attained recall scores of 0.86, 0.86, 0.84, 0.86, 0.87 and 0.86 respectively. The other sampling techniques viz., ADASYN, Borderline SMOTE, SVM SMOTE, Cluster Centroid, Tomek Links, ENN, NHC Rule, ALL KNN, SMOTE Tomek have also improved classifier performance in the similar fashion. ENN and SMOTE ENN are two such methods which have always secured the performance score over 0.91 in this case. SMOTE ENN has obtained best recall performance among all resampling methods. KNN, SVM, LR, DT, NB and MLP achieved a recall score of 0.98, 1, 0.98, 0.98, 1 and 0.97 respectively on SMOTE ENN method.

In Table 4, performance of classifiers have been reported in terms of AUC score for each resampling method. The arrangement of Table 4 is identical to previously reported tables. It is obvious from Table 4 that the performance score of the classifiers for imbalance data is again not satisfactory. KNN, SVM, LR, DT, NB and MLP have secured AUC scores of 0.18, 0.08, 0.07, 0.07, 0.03 and 0.03 respectively. Applying oversampling using SMOTE on original imbalance data, the AUC performance score of all the classifier improved significantly and KNN, SVM, LR, DT, NB and MLP acquired the AUC performance scores of 0.83, 0.84, 0.84, 0.84, 0.83 and 0.84 respectively. A similar trend of improving performance is observed in case of the other sampling techniques viz., ADASYN, Borderline SMOTE, SVM SMOTE, Cluster Centroid, Tomek Links, ENN, NHC Rule, ALL KNN, and SMOTE Tomek. Table 4 reflects that SMOTE ENN attains best AUC score when compared to all other sampling methods with an AUC score above 0.97. Following the experimental setup, KNN, SVM, LR, DT, NB and MLP have obtained AUC performance scores of 0.99, 0.99, 0.98, 0.98, 0.99 and 0.98 respectively on SMOTE ENN. The confusion matrix of best performing classifier i.e Decision Tree for imbalanced dataset has been reported in Fig. 11a which shows that majority of the samples are incorrectly classified thereby increasing both type-I and type-II errors. On the other hand the confusion matrix of the same after applying best performing resampling technique i.e SMOTE-ENN is reported in Fig. 11b which reflects significant improvement.

From Figs. 7, 8, 9 and 10 it is observed that the performance of every classifier used in the current study are extremely poor when they are trained with the original imbalanced latent vectors in terms of all performance metrics. However, after resampling the data by using various over-sampling, under-sampling and the combination of over-sampling and under-sampling methods in latent space, performance of the classifiers have improved significantly. It has been revealed that the performance obtained by the classifiers are approximately 70% better with the proposed latent space resampling framework. The figures further reveal that classifier performance have improved remarkably when SMOTE ENN, ENN and Neighborhood Cleaning Rule are used as resampling methods. In other words, the latent space resampling methods used in the current study on COVID-19 Chest X-ray images have efficiently mitigated the issue of imbalanced class problem hence the performance improvement is evident.

Table 4 Performance comparison in terms of AUC

	No resam- pling	SMOTE	ADASYN	Borderline SMOTE	SVM SMOTE	Cluster cen- troid	Tomek links	ENN	NHC rule	ALL KNN	SMOTE ENN	SMOTE Tomek
KNN	0.18	0.83	0.85	0.84	0.86	0.82	0.83	0.91	0.91	0.89	0.99	0.9
SVM	0.08	0.84	0.88	0.86	0.87	0.86	0.84	0.92	0.92	0.88	0.99	0.91
LR	0.07	0.84	0.85	0.84	0.86	0.84	0.83	0.9	0.89	0.88	0.98	0.89
DT	0.07	0.84	0.82	0.84	0.86	0.82	0.83	0.89	0.87	0.87	0.98	0.86
NB	0.03	0.83	0.84	0.86	0.86	0.82	0.83	0.89	0.87	0.84	0.99	0.87
MLP	0.03	0.84	0.85	0.83	0.85	0.83	0.86	0.92	0.89	0.9	0.98	0.88

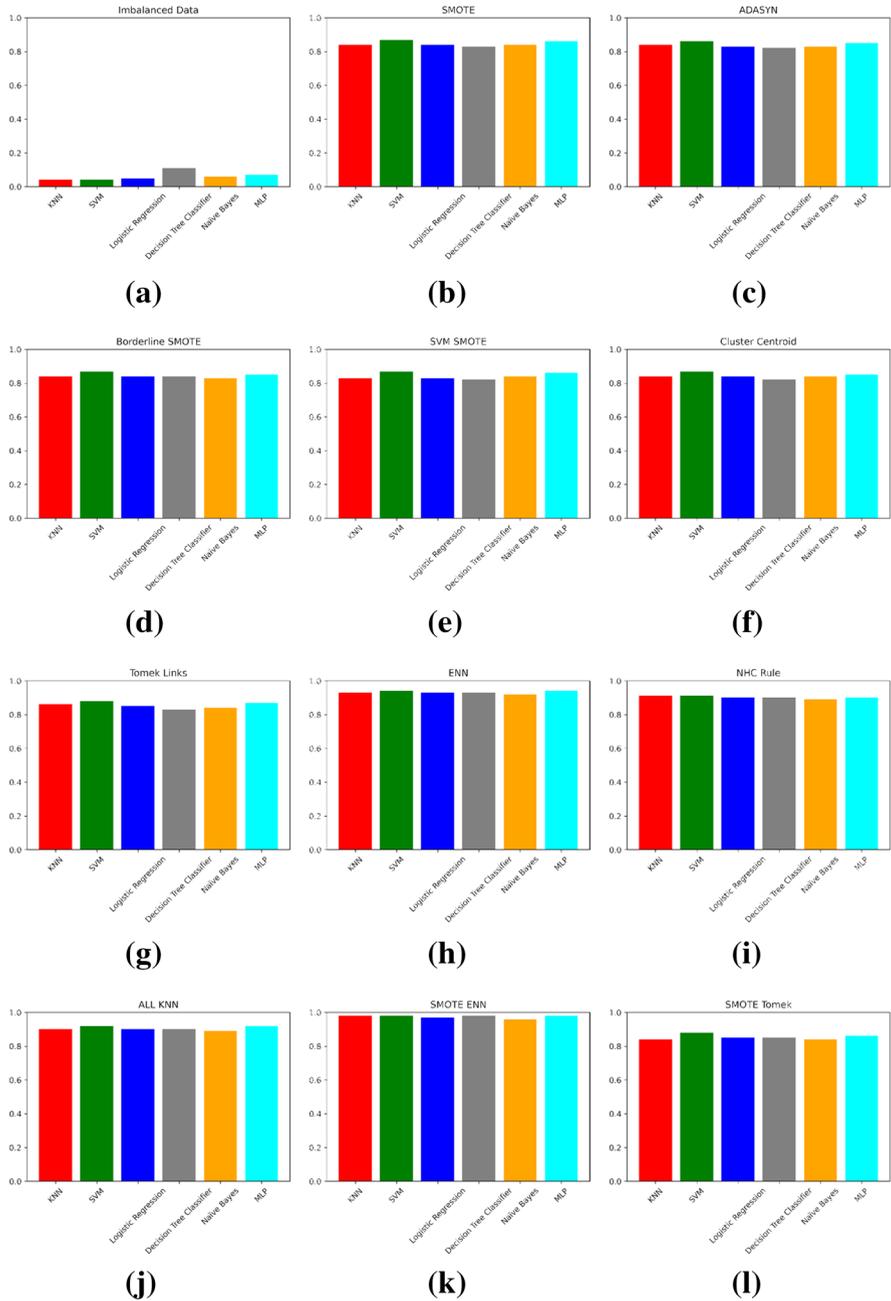


Fig. 7 Performance analysis in terms of Accuracy score

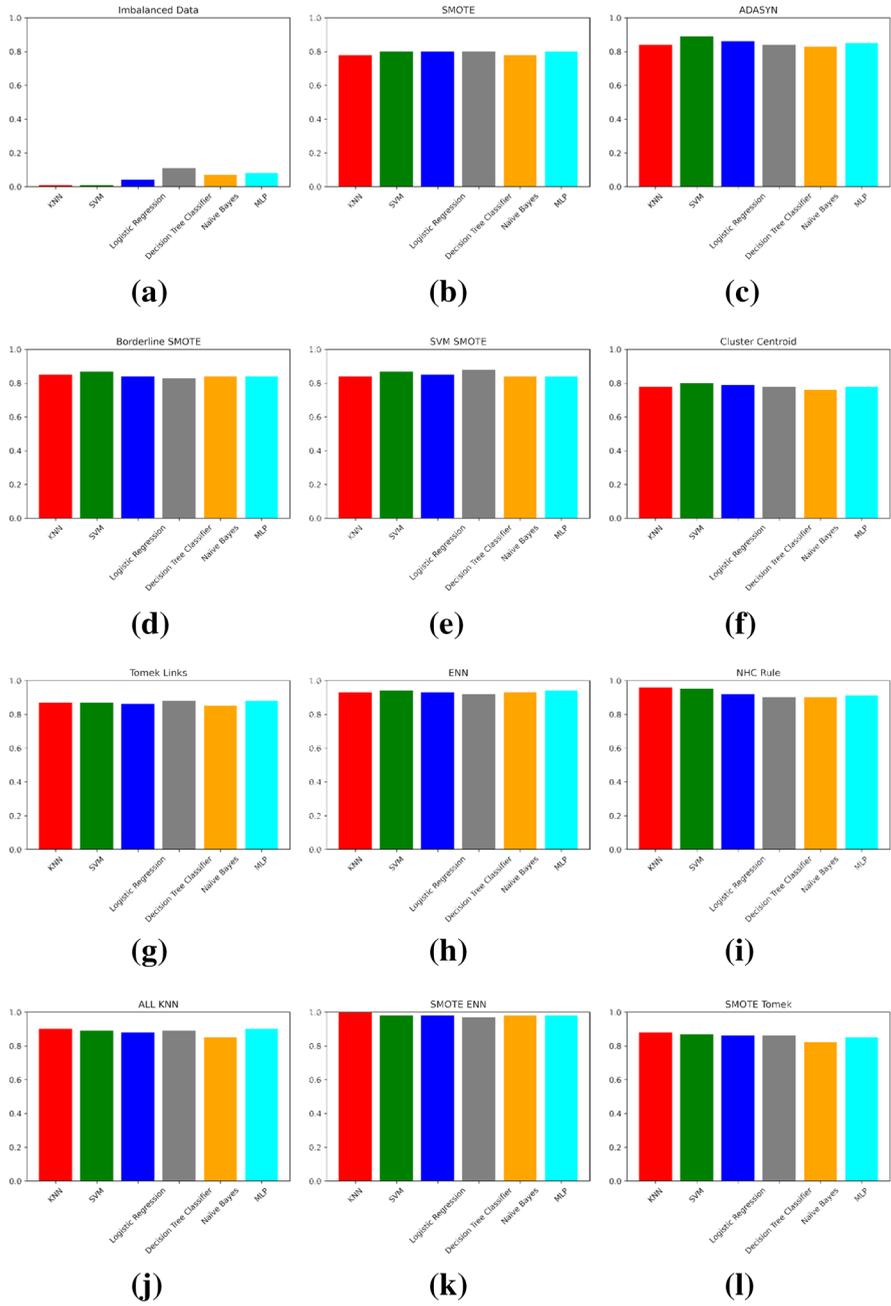


Fig. 8 Performance analysis in terms of Precision score

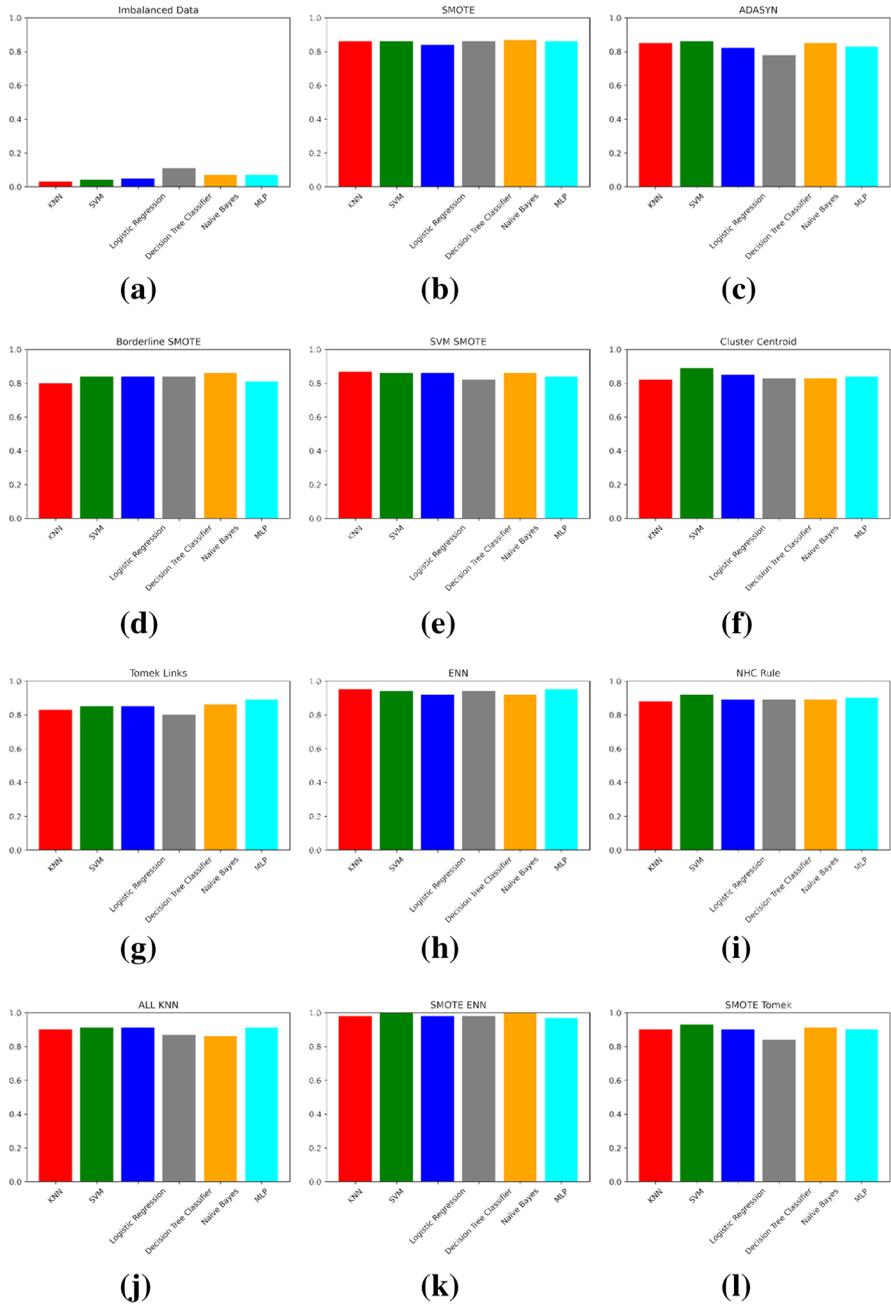


Fig. 9 Performance analysis in terms of Recall score

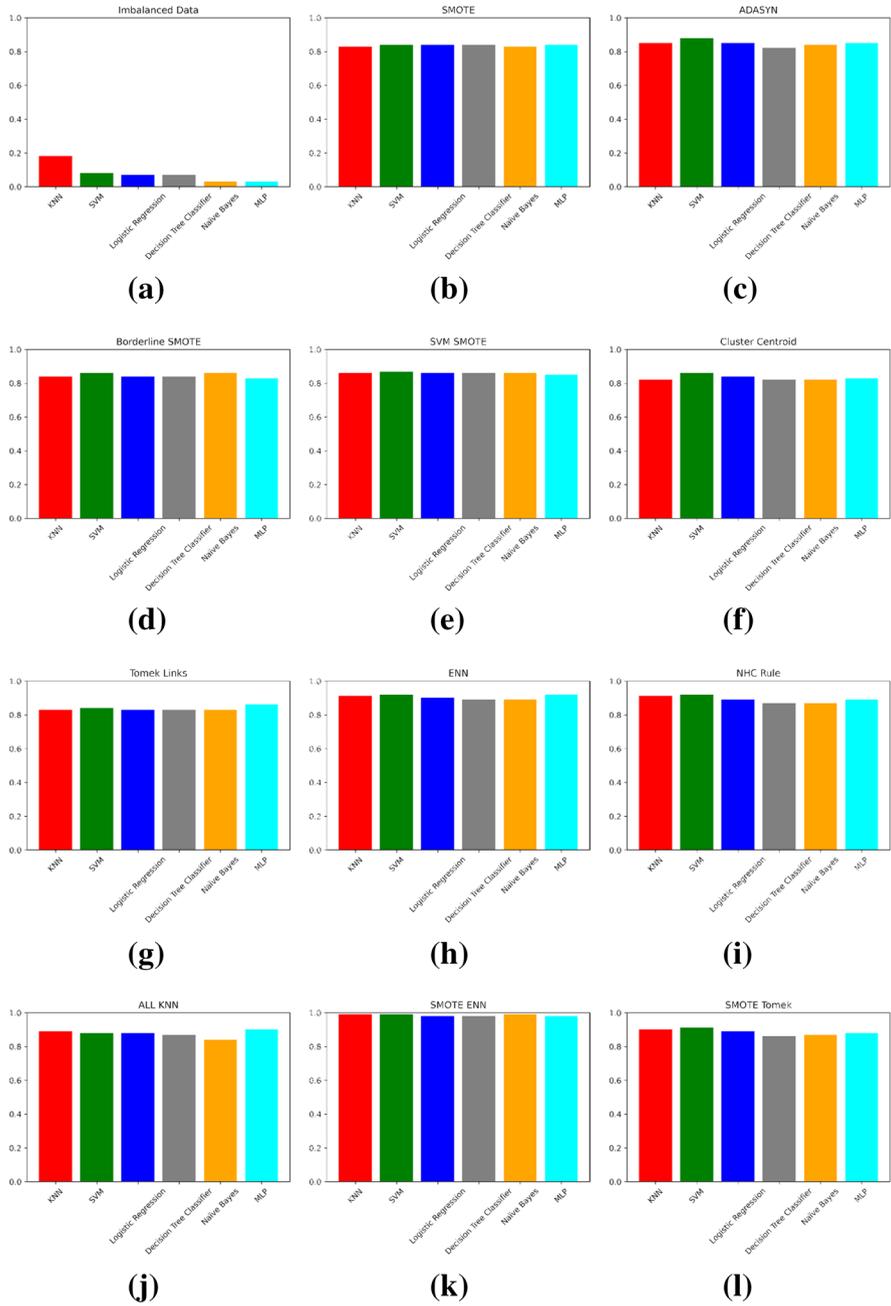


Fig. 10 Performance analysis in terms of AUC

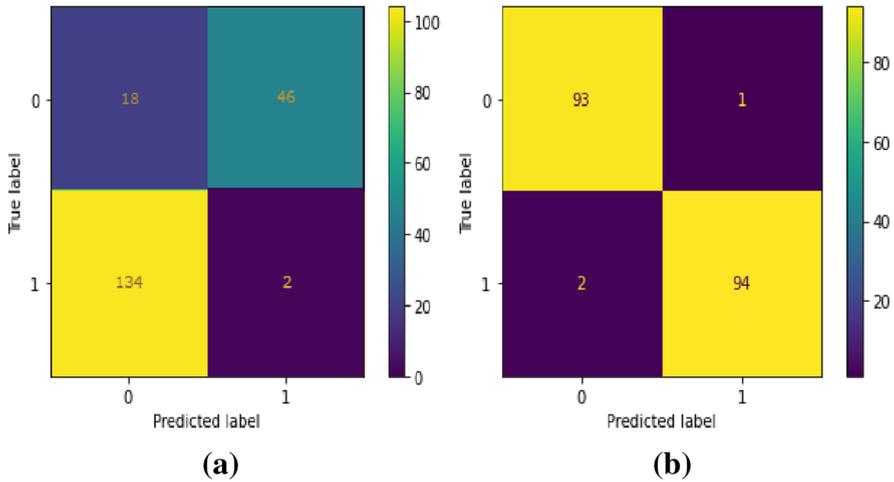


Fig. 11 **a** Confusion Matrix of best performing classifier while trained with imbalanced latent vectors. **b** Confusion Matrix of best performing classifier while trained with resampled latent vectors

3.5 Comparison with State-of-the-Art

The proposed VAE based latent space resampling method to mitigate the imbalanced class problem in detection of COVID-19 using chest X-Ray images has been thoroughly analysed in previous section. In the current section, state-of-the-art methods, already reported in literature, have been considered. Broadly two categories of research works have been reported. In the first category, various methods have been adopted to address the imbalanced classes in detecting COVID-19 using chest X-Ray images. In [62] the authors have reported a semi-supervised approach to solve the imbalance class problem. In [63] a modified version of SMOTE have been considered to mitigate the issue whereas stacking ensemble approach using multi layer perceptron has been explored in [64]. Besides, several other research works

Table 5 Comparison with state-of-the-art methods

Method	Accuracy	Precision	Recall	AUC	F1-score
Ramirez et al. [62]	0.964	0.951	0.883	0.89	0.916
Turlapati et al. [63]	0.92	0.914	0.88	0.88	0.897
Autee et al. [64]	0.95	0.96	0.98	0.98	0.97
Ahsan et al. [65]	0.98	0.95	0.93	0.94	0.94
Qiao et al. [67]	0.88	0.78	0.86	0.87	0.818
Narayanan et al. [66]	0.98	0.91	0.94	0.95	0.925
Nayak et al. [68]	0.98	0.96	0.98	0.98	0.97
Wang et al. [69]	0.92	0.88	0.92	0.93	0.9
Ozturk et al. [70]	0.98	0.98	0.95	0.95	0.965
Proposed method	0.98	1	0.98	0.99	0.99

have considered the possibilities of feature fusion [65], transfer learning [66] and specially tailored deep networks [67–70] for the same. Majority of these research works have considered accuracy, precision and recall to measure the performance of respective proposed methods. However, performance needs to be reported in terms of precision and recall to better understand whether the classifier has been able to overcome the effects of imbalanced classes or not.

Table 5 reports a comparative study where the proposed method is compared with nine other methods reported earlier. The bold values in Table 5 indicate the best performance achieved in terms of a particular performance metric. The parametric setup of all proposed method are kept as reported in respective studies. For the proposed method performance of best model setup which is SMOTE-ENN with KNN has been reported. The comparison reveals that the studies focusing specifically on imbalanced class problem have performed moderately in terms of accuracy, precision, recall and AUC scores. Whereas, the performance of proposed method is significantly better than all such methods. In addition, other studies dealing with similar detection tasks have performed well in few cases. However, in terms of Precision and AUC the proposed method has been found to be best with scores 1 and 0.99 respectively. This also suggests that the proposed VAE based latent space resampling method is better equipped to tackle imbalanced class problem in detecting COVID-19 using chest X-Ray images.

3.6 Statistical Significance

To establish the ingenuity of the proposed methods, a statistical significance test is conducted. Wilcoxon Rank test with 95% level of significance has been considered for the same. After obtaining the latent vectors from the VAE, various resampling techniques have been applied. For each resampling technique, the resampled dataset is then used to train all six classifiers. Performance of the classifiers have been measured in terms of accuracy, precision, recall and AUC score. The average of the performance metrics have been calculated to compare the resampling techniques with each other. The null hypothesis for the rank test has been taken as there is no significant difference between the average of the performance indicator of two models. On the other hand, in the alternative hypothesis there is significant difference. Each resampling technique has been run for 30 times and average performance indicator values for individual resampling technique has been calculated.

The results of the statistical significance test for accuracy has been tabulated in Table 6. The table contains the p -values obtained from the rank test for all pair of resampling techniques used in the current study. It reveals that in majority of the cases the null hypothesis has been rejected thereby indicating towards statistically significant results. The p -values reported in Tables 5, 6, 7, and 8 have been tabulated in boldface for the cases where the null hypothesis has been rejected. The rank test results for other performance indicators viz., precision, recall and AUC have been tabulated in Tables 7, 8 and 9 respectively. The p -values indicate that in majority of the cases the experimental results obtained have been found to be statistically sound.

Table 6 Wilcoxon rank test for accuracy

	SMOTE	ADASYN	BOR- DERLINE SMOTE	SVM SMOTE	CLUSTER CENTROID	TOMEK LINKS	ENN	NHC RULE	ALL KNN	SMOTE ENN
ADASYN	0.060838648	-	-	-	-	-	-	-	-	-
BORDER- LINE SMOTE	0.092662415	0.03983371	-	-	-	-	-	-	-	-
SVM SMOTE	0.013214557	0.034423034	0.026605939	-	-	-	-	-	-	-
CLUSTER CENTROID	0.086799403	0.063986434	0.032558199	0.068854722	-	-	-	-	-	-
TOMEK LINKS	0.02256007	0.001458376	0.269456128	0.005775086	0.038142077	-	-	-	-	-
ENN	0.093484671	0.007374301	0.035620171	0.0532229647	0.018519598	0.01509136	-	-	-	-
NHC RULE	0.037730319	0.03199027	0.055780721	0.061858792	0.005770234	0.041129065	0.043378686	-	-	-
ALL KNN	0.257640269	0.079595534	0.058087355	0.020625528	0.076947221	0.030474678	0.070721783	0.021209832	-	-
SMOTE ENN	0.059688231	0.020278605	0.054596992	0.037863855	0.082487753	0.148878164	0.0248971	0.006019575	0.064256847	-
SMOTE TOMEK	0.059525463	0.067023755	0.044549914	0.092892338	0.083549674	0.099913749	0.067079604	0.0019555434	0.086210657	0.025082589

Table 7 Wilcoxon Rank Test for Precision

	SMOTE	ADASYN	BOR- DERLINE SMOTE	SVM SMOTE	CLUSTER CENTROID	TOMEK LINKS	ENN	NHC RULE	ALL KNN	SMOTE ENN
ADASYN	0.02765614	-	-	-	-	-	-	-	-	-
BORDER- LINE SMOTE	0.077238377	0.003060634	-	-	-	-	-	-	-	-
SVM SMOTE	0.003037996	0.097509581	0.043939979	-	-	-	-	-	-	-
CLUSTER CENTROID	0.063731769	0.060547683	0.009158315	0.057477985	-	-	-	-	-	-
TOMEK LINKS	0.026844691	0.055379014	0.091333314	0.188583033	0.086111233	-	-	-	-	-
ENN	0.099766757	0.097969388	0.026587807	0.048840087	0.03712372	0.022291302	-	-	-	-
NHC RULE	0.048336449	0.071496204	0.052023011	0.083260835	0.053725561	0.078051844	0.051071683	-	-	-
ALL KNN	0.143710324	0.051065693	0.006298819	0.033202986	0.050315281	0.058935153	0.062610109	0.087884409	-	-
SMOTE ENN	0.009716608	0.030091405	0.07676487	0.046487994	0.04312352	0.063114588	0.007432681	0.085697828	0.016134939	-
SMOTE TOMEK	0.039462217	0.058611842	0.056650401	0.057726207	0.06827774	0.024755505	0.016069008	0.060224679	0.095166453	0.024212822

Table 8 Wilcoxon rank test for recall

	SMOTE	ADASYN	BOR- DERLINE SMOTE	SVM SMOTE	CLUSTER CENTROID	TOMEK LINKS	ENN	NHC RULE	ALL KNN	SMOTE ENN
ADASYN	0.031007242	-	-	-	-	-	-	-	-	-
BORDER- LINE SMOTE	0.095975966	0.099955116	-	-	-	-	-	-	-	-
SVM SMOTE	0.072257405	0.035270022	0.08916941	-	-	-	-	-	-	-
CLUSTER CENTROID	0.027426173	0.014461838	0.068914946	0.025461599	-	-	-	-	-	-
TOMEK LINKS	0.000164385	0.090484366	0.089122133	0.138200194	0.037126647	-	-	-	-	-
ENN	0.010942135	0.047553193	0.087721146	0.082910862	0.01768336	0.066664772	-	-	-	-
NHC RULE	0.083595649	0.018412799	0.066470637	0.007162469	0.096795869	0.004368974	0.097751077	-	-	-
ALL KNN	0.146642891	0.030263396	0.090813526	0.083167598	0.035387499	0.047337605	0.021895647	0.075581278	-	-
SMOTE ENN	0.082018709	0.044374271	0.009580631	0.050617525	0.078217696	0.040125679	0.0732012	0.02995118	0.044780416	-
SMOTE TOMEK	0.004501229	0.042873936	0.060844923	0.001964142	0.064841636	0.094334036	0.049735318	0.003338294	0.038122578	0.05076502

Table 9 Wilcoxon rank test for AUC

	SMOTE	ADASYN	BOR- DERLINE SMOTE	SVM SMOTE	CLUSTER CENTROID	TOMEK LINKS	ENN	NHC RULE	ALL KNN	SMOTE ENN
ADASYN	0.028709003	-	-	-	-	-	-	-	-	-
BORDER- LINE SMOTE	0.010285098	0.060028168	-	-	-	-	-	-	-	-
SVM SMOTE	0.095820374	0.036846147	0.039653246	-	-	-	-	-	-	-
CLUSTER CENTROID	0.079753399	0.030202484	0.093026508	0.043054336	-	-	-	-	-	-
TOMEK LINKS	0.005884766	0.081974189	0.136211693	0.008339633	0.066554507	-	-	-	-	-
ENN	0.020353253	0.0712274	0.015559541	0.021403019	0.075144805	0.028651878	-	-	-	-
NHC RULE	0.070981638	0.012347449	0.05249582	0.022179146	0.087805596	0.084819737	0.062112221	-	-	-
ALL KNN	0.080957162	0.001761758	0.07063358	0.061007095	0.015076009	0.086228817	0.052488632	0.071282633	-	-
SMOTE ENN	0.061334386	0.095069116	0.051583921	0.005182214	0.033943266	0.127854155	0.098593477	0.002605542	0.015721872	-
SMOTE TOMEK	0.09149816	0.076017617	0.05161972	0.067206289	0.050509715	0.082154663	0.024824692	0.085766773	0.055594779	0.074451545

This study proves that the performance obtained by various classifiers after applying the resampling techniques are not random, they are statistically significant.

4 Conclusion

In the current article, imbalanced COVID-19 detection using chest X-Ray images has been addressed. Initial raw images are used to train a variational autoencoder to extract most effective latent representations of the images. Next, the class biased latent vectors are resampled using three categories of resampling techniques viz., Undersampling, Oversampling, and Hybrid to mitigate the effect of imbalanced classes. The balanced dataset is then used to train and subsequently test different types of classifiers to establish the ingenuity of the proposed method. Experimental results have revealed that the proposed resampling strategy of incorporating VAE has significantly improved classifier performance. In addition, it has been observed that SMOTE ENN hybrid method has overcome the problem of imbalanced classes better than any other resampling technique considered in the current study. Experiments have revealed that SMOTE ENN based resampling method supported COVID-19 detection is at least 4% better when compared with all other resampling method of similar category. The current work considered latent space resampling method to address the imbalanced class problem, however; as a future scope of the work, generative adversarial models can be explored to generate synthetic images in order to mitigate imbalanced class problem.

References

1. Ouchicha, C., Ammor, O., Mekkassi, M.: Cvdnet: a novel deep learning architecture for detection of coronavirus (covid-19) from chest x-ray images. *Chaos, Solitons Fractals* **140**, 110245 (2020)
2. Khan, S.H., Sohail, A., Zafar, M.M., Khan, A.: Coronavirus disease analysis using chest x-ray images and a novel deep convolutional neural network. *Photodiagn. Photodyn. Ther.* **35**, 102473 (2021)
3. Shibly, K.H., Dey, S.K., Islam, M.T.-U., Rahman, M.M.: Covid faster r-cnn: a novel framework to diagnose novel coronavirus disease (covid-19) in x-ray images. *Inf. Med. Unlocked* **20**, 100405 (2020)
4. Worldometer. Covid-19 coronavirus pandemic, 2021. <https://www.worldometers.info/coronavirus/>. Accessed 18 Nov 2021
5. Ahmad, F., Farooq, A., Ghani, M.U.: Deep ensemble model for classification of novel coronavirus in chest x-ray images. *Comput. Intell. Neurosci.* **2021** (2021)
6. Jacobi, A., Chung, M., Bernheim, A., Eber, C.: Portable chest x-ray in coronavirus disease-19 (covid-19): a pictorial review. *Clin. Imaging* **64**, 35–42 (2020)
7. Roy, M., Chakraborty, S., Mali, K., Banerjee, A., Ghosh, K., Chatterjee, S.: Biomedical image security using matrix manipulation and dna encryption. In: *International Ethical Hacking Conference*, pp. 49–60. Springer (2019)
8. Ding, W., Chakraborty, S., Mali, K., Chatterjee, S., Nayak, J., Das, A.K., Banerjee, S.: An unsupervised fuzzy clustering approach for early screening of covid-19 from radiological images. *IEEE Trans. Fuzzy Syst.* **30**(8) (2021)
9. Sallay, H., Bourouis, S., Bouguila, N.: Online learning of finite and infinite gamma mixture models for covid-19 detection in medical images. *Computers* **10**(1), 6 (2021)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)

11. Sun, W., Tseng, T.-L.B., Zhang, J., Qian, W.: Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data. *Comput. Med. Imaging Graph.* **57**, 4–9 (2017)
12. Larrazabal, A.J., Martínez, C., Glocker, B., Ferrante, E.: Post-dae: anatomically plausible segmentation via post-processing with denoising autoencoders. *IEEE Trans. Med. Imaging* **39**(12), 3813–3820 (2020)
13. Singh, S.R., Dubey, S.R., Shruthi M.S., Ventrapragada, S., Dasharatha, S.S.: Joint triplet autoencoder for histopathological colon cancer nuclei retrieval. *arXiv preprint arXiv:2105.10262* (2021)
14. Baur, C., Wiestler, B., Albarqouni, S., Navab, N.: Bayesian skip-autoencoders for unsupervised hyperintense anomaly detection in high resolution brain mri. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), pages 1905–1909. IEEE, (2020)
15. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **21**(9), 1263–1284 (2009)
16. Pes, B.: Learning from high-dimensional biomedical datasets: the issue of class imbalance. *IEEE Access* **8**, 13527–13540 (2020)
17. Liu, S., Zhang, J., Xiang, Y., Zhou, W., Xiang, D.: A study of data pre-processing techniques for imbalanced biomedical data classification. *Int. J. Bioinform. Res. Appl.* **16**(3), 290–318 (2020)
18. Guzmán-Ponce, A., Sánchez, J.S., Valdovinos, R.M., Marcial-Romero, J.R.: Dbig-us: a two-stage under-sampling algorithm to face the class imbalance problem. *Expert Syst. Appl.* **168**, 114301 (2021)
19. Devi, D., Namasudra, S., Kadry, S.: A boosting-aided adaptive cluster-based undersampling approach for treatment of class imbalance problem. *Int. J. Data Warehous. Min. (IJDWM)* **16**(3), 60–86 (2020)
20. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: *Conference on Artificial Intelligence in Medicine in Europe*, pages 63–66. Springer (2001)
21. Junsomboon, N., Phienthrakul, T.: Combining over-sampling and under-sampling techniques for imbalance dataset. In: *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 243–247 (2017)
22. Zhang, J., Chen, L., Abid, A.: Prediction of breast cancer from imbalance respect using cluster-based undersampling method. *J. Healthcare Eng* **22** (2019)
23. Chawla, N.V., Bowyer, K.W., Hall, L.O., Philip Kegelmeyer, W.: Smote synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
24. Ishaq, A., Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., Nappi, M.: Improving the prediction of heart failure patients' survival using smote and effective data mining techniques. *IEEE Access* **9**, 39707–39716 (2021)
25. Venu, S.K.: Improving the generalization of deep learning classification models in medical imaging using transfer learning and generative adversarial networks. In: *International Conference on Agents and Artificial Intelligence*, pp. 218–235. Springer, Cham (2022)
26. Karabulut, E.M., Ibrikci, T.: Effective automated prediction of vertebral column pathologies based on logistic model tree with smote preprocessing. *J. Med. Syst.* **38**(5), 1–9 (2014)
27. Banik, D., Bhattacharjee, D.: Mitigating data imbalance issues in medical image analysis. In: Rana, D.P., Mehta, R.G. (eds.) *Data Preprocessing, Active Learning, and Cost Perceptive Approaches for Resolving Data Imbalance*, pp. 66–89. IGI Global (2021)
28. Wang, K.-J., Adrian, A.M., Chen, K.-H., Wang, K.-M.: A hybrid classifier combining borderline-smote with airs algorithm for estimating brain metastasis from lung cancer: A case study in taiwan. *Comput. Methods Progr. Biomed.* **119**(2), 63–76 (2015)
29. Guo, R., Guo, J., Zhang, L., Xiaoxia, Q., Dai, S., Peng, R., Chong, V.F.H., Xian, J.: Ct-based radiomics features in the prediction of thyroid cartilage invasion from laryngeal and hypopharyngeal squamous cell carcinoma. *Cancer Imaging* **20**(1), 1–11 (2020)
30. Shyamala Devi, M., Sridevi, S., Bonala, K.K., Dadi, R.H., Reddy, K.V.R.: Oversampling response stretch based fetal health prediction using cardiocographic data. *Ann. Rom. Soc. Cell Biol.* **25**(5), 1448–1464 (2021)
31. Wattenberg, M., Viégas, F., Johnson, I.: How to use t-sne effectively. *Distill* **1**(10), e2 (2016)
32. Bengio, Y.: *Learning Deep Architectures for AI*. Now Publishers Inc, Delft (2009)
33. Bank, D., Koenigstein, N., Giryas, R.: Autoencoders. *arXiv preprint arXiv:2003.05991* (2020)
34. Hebb, D.O.: *The Organization of Behavior: A Neuropsychological Theory*. Psychology Press, Hove (2005)
35. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1798–1828 (2013)

36. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: *Advances in Neural Information Processing Systems* **19** (2006)
37. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
38. Gregor, K., Danihelka, I., Graves, A., Rezende, D., Wierstra, D.: Draw: a recurrent neural network for image generation. In: *International Conference on Machine Learning*, pp. 1462–1471. PMLR (2015)
39. Babaieizadeh, M., Finn, C., Erhan, D., Campbell, R.H., Levine, S.: Stochastic variational video prediction. arXiv preprint [arXiv:1710.11252](https://arxiv.org/abs/1710.11252) (2017)
40. Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. *Adv. Neural Inf. Process. Syst.* **29**, 3738–3746 (2016)
41. Nguyen, T.-D., Nguyen, D.-K., Yu-Yen, O.: Addressing data imbalance problems in ligand-binding site prediction using a variational autoencoder and a convolutional neural network. *Brief. Bioinform.* **26**, 277 (2021)
42. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. *Sp. Lect. IE* **2**(1), 1–18 (2015)
43. Paisley, J., Blei, D., Jordan, M.: Variational Bayesian inference with stochastic search. arXiv preprint [arXiv:1206.6430](https://arxiv.org/abs/1206.6430) (2012)
44. Krawczyk, B., Galar, M., Jeleń, Ł., Herrera, F.: Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. *Appl. Soft Comput.* **38**, 714–726 (2016)
45. Bhattacharjee, M., Ghosh, K., Banerjee, A., Chatterjee S.: Multilabel sentiment prediction by addressing imbalanced class problem using oversampling. In: *Advances in Smart Communication Technology and Information Processing: OPTRONIX 2020*, pp. 239–249. Springer (2021)
46. Cavadas, B., Branco, P., Pereira, S.: Crime prediction using regression and resources optimization. In: *Portuguese Conference on Artificial Intelligence*, pp. 513–524. Springer (2015)
47. Banerjee, A., Bhattacharjee, M., Ghosh, K., Chatterjee, S.: Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media. *Multimed. Tools Appl.* **79**(47), 35995–36031 (2020)
48. Branco, P., Torgo, L., Ribeiro, R.P.: A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv. (CSUR)* **49**(2), 1–50 (2016)
49. de Moraes, R.F.A.B., Vasconcelos, G.C.: Boosting the performance of over-sampling algorithms through under-sampling the minority class. *Neurocomputing* **343**, 3–18 (2019)
50. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. *Progress Artif. Intell.* **5**(4), 221–232 (2016)
51. Sáez, J.A., Luengo, J., Stefanowski, J., Herrera, F.: Smote-ipf: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inf. Sci.* **291**, 184–203 (2015)
52. He, H., Bai, Y., Garcia, E.A., Li S.: Adasyn: adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE world congress on computational intelligence)*, pp. 1322–1328. IEEE (2008)
53. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: *International Conference on Intelligent Computing*, pp. 878–887. Springer (2005)
54. Nguyen, H.M., Cooper, E.W., Kamei, K.: Borderline over-sampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradig.* **3**(1), 4–21 (2011)
55. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Rusboost: improving classification performance when training data is skewed. In: *2008 19th International Conference on Pattern Recognition*, pp. 1–4. IEEE (2008)
56. Batista, G.E.A.P.A., Bazzan, A.L.C., Monard, M.C., et al.: Balancing training data for automated annotation of keywords: a case study. In: *WOB*, pp. 10–18 (2003)
57. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Expl. NewsL* **6**(1), 20–29 (2004)
58. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546 (2019)
59. Raikote, P.: Covid-19 image dataset, April 2020. <https://www.kaggle.com/pranavraikote/covid19-image-dataset>. Accessed 18 Nov 2021

60. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**(2), 281–305 (2012)
61. Hackeling, G.: *Mastering Machine Learning with Scikit-Learn*. Packt Publishing Ltd, Birmingham (2017)
62. Calderon-Ramirez, S., Yang, S., Moemeni, A., Elizondo, D., Colreavy-Donnelly, S., Chavarría-Estrada, L.F., Molina-Cabello, M.A.: Correcting data imbalance for semi-supervised covid-19 detection using x-ray chest images. *Appl. Soft Comput.* **111**, 107692 (2021)
63. Venkata Pavan Kumar Turlapati and Manas Ranjan Prusty: Outlier-smote: a refined oversampling technique for improved detection of covid-19. *Intell.-based Med.* **3**, 100023 (2020)
64. Autee, P., Bagwe, S., Shah, V., Srivastava, K.: Stacknet-dennis: a multi-layer perceptron stacked ensembling approach for covid-19 detection using x-ray images. *Phys. Eng. Sci. Med.* **43**(4), 1399–1414 (2020)
65. Mominal Ahsan, Md., Based, J.H., Kowalski, M., et al.: Covid-19 detection from chest x-ray images using feature fusion and deep learning. *Sensors* **21**(4), 1480 (2021)
66. Narayanan, B.N., Hardie, R.C., Krishnaraja, V., Karam, C., Davuluru, V.S.P.: Transfer-to-transfer learning approach for computer aided detection of covid-19 in chest radiographs. *AI* **1**(4), 539–557 (2020)
67. Qiao, Z., Bae, A., Glass, L.M., Xiao, C., Sun, J.: Flannel (focal loss based neural network ensemble) for covid-19 detection. *J. Am. Med. Inf. Assoc.* **28**(3), 444–452 (2021)
68. Nayak, S.R., Nayak, D.R., Sinha, U., Arora, V., Pachori, R.B.: Application of deep learning techniques for detection of covid-19 cases using chest x-ray images: A comprehensive study. *Biomed. Signal Process. Control* **64**, 102365 (2021)
69. Wang, L., Lin, Z.Q., Wong, A.: Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Sci. Rep.* **10**(1), 1–12 (2020)
70. Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O., Rajendra Acharya, U.: Automated detection of covid-19 cases using deep neural networks with x-ray images. *Comput. Biol. Med.* **121**, 103792 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Sankhadeep Chatterjee¹  · Soumyajit Maity² · Mayukh Bhattacharjee² · Soumen Banerjee³ · Asit Kumar Das¹ · Weiping Ding⁴

Soumyajit Maity
soumyajitmaity2709@gmail.com

Mayukh Bhattacharjee
mayukhofficial12@gmail.com

Soumen Banerjee
prof.sbanerjee@gmail.com

Asit Kumar Das
akdas@cs.iiests.ac.in

Weiping Ding
dwp9988@hotmail.com

- ¹ Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, West Bengal, India
- ² Department of Computer Science and Engineering, University of Engineering & Management, Kolkata, West Bengal, India
- ³ Department of Electronics and Communication Engineering, Budge Budge Institute of Technology, Budge Budge, Kolkata, West Bengal 700137, India
- ⁴ School of Information Science and Technology, Nantong University, 66479, Nantong 226019, Jiangsu, China