Est. 1841 | YORK ST JOHN UNIVERSITY

Siddalingappa, Rashmi and Kanagaraj, S. (2021) Anomaly Detection on Medical Images using Autoencoder and Convolutional Neural Network. International Journal of Advanced Computer Science and Applications, 12 (7). pp. 148-156.

Downloaded from: https://ray.yorksj.ac.uk/id/eprint/12880/

The version presented here may differ from the published version or version of record. If you intend to cite from the work you are advised to consult the publisher's version: http://dx.doi.org/10.14569/IJACSA.2021.0120717

Research at York St John (RaY) is an institutional repository. It supports the principles of open access by making the research outputs of the University available in digital form. Copyright of the items stored in RaY reside with the authors and/or other copyright owners. Users may access full text items free of charge, and may download a copy for private study or non-commercial research. For further reuse terms, see licence terms governing individual outputs. Institutional Repositories Policy Statement

RaY

Research at the University of York St John

For more information please contact RaY at ray@yorksj.ac.uk

Anomaly Detection on Medical Images using Autoencoder and Convolutional Neural Network

Rashmi Siddalingappa[1], Sekar Kanagaraj[2]
Department of Computational and Data Science
Indian Institute of Science, C V Raman Road, Bangalore 560012, India

***Abstract*—Detection of anomalies from the medical image dataset improves prognosis by discovering new facts hidden in the data. The present study aims to discuss anomaly detection using autoencoders and convolutional neural networks. The autoencoder identifies the imbalance between normal and abnormal samples. They create learning models flexible and accurate on training data. The problem is addressed in four stages: 1) training: an autoencoder is initialized with the hyper-parameters and trained on the lung cancer CT scan images, 2) test: the autoencoder reconstructs the input from the latent space representation with a slight variation from the original data, indicated by a reconstruction error as Mean Squared Error (MSE), 3) evaluate: the MSE value of the training and test dataset are compared. The MSE values of anomalous data are higher than a base threshold, detecting those as anomalies, 4) validate: the efficiency metrics such as accuracy and MSE scores are used at both training and validation phases. The dataset was further classified as benign and malignant. The accuracy reported for outlier detection and the classification task is 98% and 97.2%. Thus, the proposed autoencoder-based anomaly detection could positively isolate anomalies from the CT scan images of lung cancer.**

***Keywords*—Anomalies; autoencoder; convolutional neural networks (CNN) (ConvNets); deep neural network architecture; regularization**

I. Introduction

Outliers are the data that are not normal when compared to the rest of the information in any dataset. They indicate extreme values which usually diverge from the general model [1]. The occurrence of outliers in the dataset is possible for many reasons, such as a fault in the system, manual errors, fraudulent errors, equipment errors, and the data may vary for inexplicable reasons camouflaging a few unseen motifs. At times, these unusual patterns indicate hidden knowledge about the data. For instance, irregular Electrocardiography (ECG) data may suggest heart-related problems because it will be dissimilar from the ECG report of a healthy person. Thus, identifying outliers is an essential part of the knowledge discovery process [2]. Because of this reason, outlier detection has always been an exciting factor for researchers, scientists, and data analysts. Outlier detection is widely employed in nearly all subject areas such as medical, fraud detection, credit card analysis, financial sectors, social network analysis, and weather forecast analysis. Outliers are of different types: univariate, multivariate, point/global, context, and collective outliers [3]. The outlier detection approaches [4] are broadly classified into three categories; 1) Statistical method: this approach is used in a typical univariate environment where the data distribution is normal/ Gaussian-like. Here, approximately 68% of the data fall with the normal distribution anchored to the 1st standard deviation measure. About 98% of data distribution fall in the 2nd standard deviation and 99.7% of data value belong to the 3rd standard deviation. The approach yields faster results. The compact representation of the model facilitates anomaly detection even on large datasets. However, the statistical methods often fail in a multidimensional dataset environment, and also, they require prior knowledge about the anomaly pattern [5], 2) supervised method: The model is trained on the labeled features that differentiate between a normal and an abnormal data class. The unseen data is fed to the system, i.e., test data, and the model determines to which category the data point belongs. Interestingly, they do not rely on any prior knowledge of the anomaly pattern and it is easy to train the model. Again, this model fails in a high-dimensional space, further attributed with the local neighborhood problem [6], 3) unsupervised method: the anomalies are detected through a heuristic approach with certain assumptions of segregating the regular instances versus other data points that deviate from the cluster. K-means and DBSCAN are the prominent techniques here [7]. These methods are highly dependent on users' perception making the outlier detection task quite spontaneous. The main drawback of this approach is the binary nature of data separation, which is used for data grouping. Several algorithms have been proposed in the realm of anomaly detection however, they focus on arbitrary labels in the classification of datasets to distinguish between previously observed outlier samples. The protocols for feature selection are not indicated, making the model detect only the previously known anomalies. Moreover, the statistical methods will lead to high false negatives that may skip identifying the actual anomalies, and the rule-based models are highly dependent on user-set parameters, whereby changing these features will negatively impact the performance of the model [8]. Therefore, to fill the research gap of the existing methods, the present study uses a deep learning approach – autoencoder and convolution neural network (CNN). These methods have been doing miracles on a diverse range of datasets amidst any complexities in the structure. Thus, the main objective of the present study is to use an autoencoder with encoder and decoder arrangement to detect and eliminate outliers on lung cancer computerized tomography (CT) scan images [9]. During the training, the encoder will learn the latent representation of the normal data at the core layer. Thereon, the decoder will use the information present in the core layer

to reconstruct the data. The normal and abnormal data's behavior is separated by using a Mean Squared Error (MSE) score. The MSE calculates the difference between the original input data and the data the model constructed at the output side. For a good model, the MSE scores should be small. In further steps, the images are subsequently classified into benign and malignant. The significant contributions of the proposed work are as follows:

- Image datasets are highly sparse with a complex structure. Thus, the study empirically demonstrates a deep neural architecture to detect the medical image outliers.
- The input data distribution is transformed into output distribution space with the least amount of feature loss (distortion).
- A reconstruction error is calculated for the training and test data for understanding the gap between normal and abnormal data samples. A base threshold is pivotal for this mapping function [10].
- The proposed method works on an unsupervised dataset without any labels, making the framework efficient enough to ascertain the unusual patterns in the underlying data.

The remainder of the paper is organized in the following sections. Section 2 discusses various works related to the present study. Section 3 introduces the autoencoder. The implementation details are shown in Section 4. Section 5 delivers results and analysis of the proposed model. Lastly, the paper culminates with Sections 6 and 7, highlighting the discussions, scope for future research, and conclusion.

II. Literature Study

The problems associated with anomaly detection are found abundantly in the literature. Various researchers have proposed various models and methods globally in the past two decades [11] [12]. In [13], LUNA16 dataset, CT scan images with label nodules are used by the authors to detect cancer using 3D-CNN. Initially, the raw images are preprocessed using a threshold approach, and later vanilla 3D NN architecture is used to classify the images into cancerous and non-cancerous. The model achieved 80% accuracy with 120-sec computational time. Though the results of this research work are better than the previous results, it uses a relatively small amount of dataset (~100 CT images). The same LUNA16 (lung nodule analysis 2016) datasets have been used by the authors Gritli, et al. in [14]. The aim was to classify the datasets into benign and malignant using 3D AlexNet architecture. Through 10-fold cross-validation, the proposed approach resulted in 97%, proving to be efficient than the existing methods even at low-dose CT scan images. However, the layers at the semantic network are tiny and light, making the class activation function not perform well. There was a significant amount of data lost in the process of maintaining the class equivalence. The lung cancer detection in CT scan images using CNN is proposed by Sharma, et al. in [15]. The researchers have performed preprocessing and segmentation. Later U-net model is used to classify the patients' nodules into cancerous or non-cancerous. The authors claim to obtain 77% accuracy but the proposed model suffers from data-imbalance problems, due to which the accuracy is dropped. Rasha, et al. [16] have worked on anomaly detection in lung cancer image datasets. The features have been selected through techniques such as local binary pattern (LBP), discrete wavelet transform (DWT), and histogram of oriented gradients (HOG). The firefly algorithm is used to optimize the selected features and later on support vector machine (SVM) is applied to classify the normal instance of the image. The authors have not shown the real-time datasets taken from Moulana hospital. The details of the preprocessing of the dataset are not discussed. When the training set contains a small fraction of outliers, it becomes extremely challenging to identify anomalies in the given image dataset. Thus Laura Beggel, et al. in [17] have proposed a unique anomaly detection using adversarial autoencoders that places anomaly patterns in low likelihood regions. The proposed model is performed on the MNIST image dataset. The model resulted in some overlap with reconstruction images making the task rely on a supervised training mode. The performance is not studied for a high-dimensional dataset. The 3D-National lung screening trial (NLST) datasets have been used to study anomaly detection using deep generative models in [18]. The model works on the fact that positive samples are available in scarce; thus, the likelihood of the unseen data is estimated without the implications of the negative samples, thereby identifying the samples as low likelihood datapoints. However, the applicability is not suited when the complexity of the data increases. The results of the 0.62 score under ROC results are still not good enough for determining anomalies at the nodule level. Mehdi, et al. [19] have proposed lung cancer detection using an autoencoder that is semi-automatically trained on datasets from the Lung Image Database Consortium image collection (LIDC-IDRI) database. The dataset of healthy patients is used for training, later the output was fed to a segmentation process, and the variation in a pattern other than healthy patients was removed. However, the segmentation network could fail while training on abnormalities of the diseased images.

III. Architecture Recurrent Outlier Detection using Deep Neural Architecture – Autoencoders

Autoencoders (AE), a multi-layered feed-forward neural network, is an unsupervised machine learning approach [20] used for dimensionality reduction in a multivariate data environment. However, on a univariate dataset, the autoencoders are similar to linear regression or a typical principal component analysis (PCA) problem [21]. Though PCA and other clustering algorithms perform reasonably well on multidimensional data, the autoencoder does a better job because of hyper-parameters [22]. A significant difference between a PCA and an AE is that the latter perform analysis on the data with a non-linear activation function on the hidden layers. Architecturally, an AE is a simple feed-forward network because the information is fed to the input layer, passed through a set of hidden layers. Each has a varied number of nodes/neurons to transform the input and arrives at the output. The nodes are extrapolated into different layers, each connected to all the nodes on the previous layers. The

input and the output layers have the same number of nodes, 'n,' due to the symmetric arrangement of the autoencoder that intends to reconstruct input at the output side. The values predicted at each node through activation functions are passed into consecutive layers ahead. The general representation of AE is shown in Fig. 1. An AE consists of two main stages, an encoder and a decoder [23]. An encoder maps the given input into a compressed representation, and a decoder transforms the compressed data back into the original input. Alongside, an encoder wraps the original data by hidden layers into a squeezed vector representation.

$$x_n = \sum_{i=1}^{n} e_n(w_{en}x_0 + b_{en}) \quad (1)$$

Where, e_n is an encoding function of the hidden layer ranging between 1 and n, w_{en} and b_{en} are the weight and bias parameters at layer 'n' and x_0 is the original input vector from the input layer. Similarly, at the decoder side, the output will be the same as the input that the system received initially but with a difference that the output at encoder represents the input (x) as a reconstruction error for x_0.

$$x' = \sum_{i=1}^{n} d_n(w_{dn}x_n + b_{dn}) \quad (2)$$

Where, x' is a decoding function at nth decoding hidden layer with the weights and bias being represented for the corresponding nth decoding layer as w_{dn} and b_{dn}.

The AE extracts the crucial features and stuff in a latent space representation between an encoder and a decoder. Besides, this representation contains a low-dimensional version of the original input. Thereby, at the decoder, the AE reconstructs the input data as the output from the latent space features. This reconstruction is dependent on the training data, i.e., an AE cannot build a new representation of the input but only specific to what has been trained. Furthermore, the autoencoder calculates the reconstruction error through MSE. For a normal data sample, the reconstruction error is small. However, these numbers are usually large and above a certain base threshold for the anomalous data, typically set by the user.

The encoding section takes the input image; the autoencoder captures only the spatial features and converts them to a low dimensional image. Further, in the decoding section, the image is reconstructed.

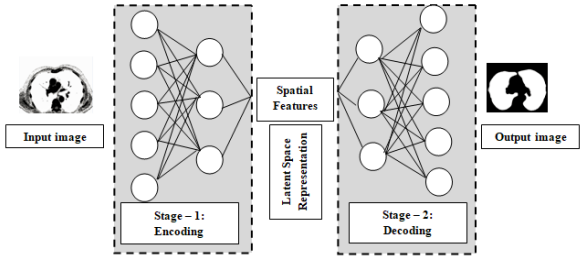

Fig. 1. A Diagrammatic view of an Autoencoder Network.

IV. Autoencoders and its Components in Anomaly Detection

The fundamental role of an AE in anomaly detection is to determine how much the output data (reconstructed data) deviates from the input data. Thus, the AE is essentially trained on the theory of minimizing the reconstruction error. The following parameters are considered during the training process:

- The number of hidden layers – The decision boundary is observed to split the input data into several classes, and later these classes are expressed as a straight line [24]. The joining curve of these lines indicates the number of hidden layers, and the number of consecutive lines decides the number of neurons in these hidden layers. In an AE, the number of neurons in the input and the output layers are the same.
- Regularization – The main objective of using any machine learning approach is to make the model fit for both training and test data to avoid overfitting and underfitting. In both cases, the model will not generalize well. The regularization techniques are adopted to minimize the error rate on the test data at the cost of boosting the training error. Lasso regression (L1) and Ridge regression (L2) are the two popular regularization methods [25]. Here, L1 regularization [26] is used since this is particularly useful for the feature selection process on a wide range of input values. The loss function is given by;

Loss Function

$$= \sum_{i=1}^{n}(Z_i - \sum_{j=1}^{p} y_{ij}\alpha_j)^2 \quad (3)$$

Here, Zi is the input variable at some neuron layer 'i' (i ∈ 1 to 'n' inputs), and yij is the output layer obtained at some neuron layer 'j' corresponding to the input 'i'. The output has 'j' layers, the same as the input layer such that j ∈ 1 to 'p' outputs. αj is the reconstruction error. The entire component is squared to eliminate any negative value. The L1 regression defines an absolute value of the magnitude for a penalty term along with loss function [27], and it is given by;

Regularization Function (L1) =

$$\text{Loss Function} + \rho\sum_{j=1}^{p}|\alpha_j| \quad (4)$$

- Learning rate: Indicates the number of weights updated at every epoch. It tunes the algorithm to achieve minimum reconstruction error.
- Batch size: This refers to the number of training samples used at different iterations from which the model learns.
- Optimizer: An optimizer is used to combat the time complexity of the algorithm. Adam Optimization algorithm [28] is a replacement for a traditional stochastic gradient descent method to update the

training network's weights. The learning rate is calculated for various parameters and frequently preserved for individual network weights. These values are finally adopted as a learning process unfolds.

A. Training a Deep Neural Network through ConvNets

When an input image passed through a standard neural network, many of the temporal [29] (time-related: pictures that were taken at different time intervals) and spatial [30] (space-related: properties related to a single image such as coordinates, gradients, resolution and so on) features are lost. Convolutional Neural Network – ConvNet – CNN [31] model is used to overcome this problem. Spatial elements are essential to reconstruct the images as they describe each image's characteristics. An AE retains only spatial features, eliminating the images' temporal aspects. The encoder comprises three ConvNet layers with different dimensions. At the core, there is a hidden layer that is dense and fully connected autoencoder with neurons. Once the image is resized, a low-dimensional version of the input is stored in the latent space. The decoder comprising three deConvNets reconstructs the input image with limited features. Each image is 512x512 pixels. The first layer of ConvNet is a convolutional layer with 32 filters such that each filter is of size 5x5. Only one feature out of 32 will be considered at each evaluation step, indicated by 512x512x1. The second layer is pooling with a 3x3 pool size. The output size is 509x509 since pooling prunes 3x3 pixels from each side. Here, the image would be reduced to 169x169 (509/3 ~ 169) with 32 filters (a similar process is repeated for the 2nd and 3rd ConvNets). The flattening process induces the product of these numbers. The pooled features of the input image are mapped onto columnar representation. The fully connected layer in the core is then turned on with batch size = 128. The spatial features are juxtaposed to form many attributes sufficient to create the original input image. At the decoding side of DeConvNets, the same process is reversed by retaining the dimensions constant.

The architecture of a CNN model is shown in three stages, Fig. 2.

- Convolutional Layer: The feature space is created for an input image and preserves the relationship between the pixels through filtering. The filters' values are usually; 1, -1, and 0 – a positive value for feature brightness, a negative value for darkness, and 0 for a grey image. These values are placed indefinitely at different locations in the filters. When an original image passes through the filters, the filtered image features produce two types of high and low scores for a match and low for a no-match/mismatch. The filters here represent the number of features that the model can extract. However, with a more significant number of filters, the training process is prolonged. The filter values are 32, 64, 128, and so on.
- Activation Function: The activation function helps the model map the resulting feature values into a normalized value between 0 to 1 and -1 to +1. In the proposed system, two activation functions are Sigmoid and ReLu. The sigmoid function squashes the feature values between 0 and 1. The ReLu – Rectified Linear Unit – substitutes a negative value to zero [32].
- Pooling: This is used to reduce the size of filter vectors. For instance, in max-pooling, if the filter is 3X3, the highest value is chosen at every 3X3 matrix. Once the pooling is completed, the filtered images are stacked up to form a list.

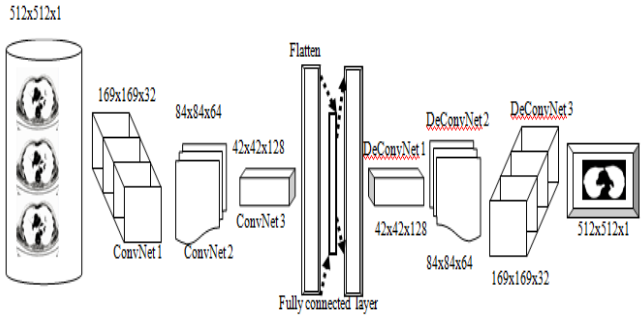

Fig. 2. The Encoding and Decoding Processes of an Autoencoder using ConvNets and DeConvNets, Respectively. The Encoding and Decoding Processing are Symmetric and have the Same Layers in each Section.

B. Training Algorithm for ConvNets-Autoencoders using Adam Optimization Technique

Adam optimizer is used to train a deep neural network using ConvNets. Here, the learning curve is estimated based on the lower-order moments with fewer memory requirements. Algorithm 1 illustrates the training process adopted for this research study.

Algorithm 1: Training algorithm for ConvNets using Adam Optimization Technique

Input: Training data split (d); the input vector (x_0); Adam's learning rate parameters β = (e_n, w_{en}, b_{en}); the number of hidden layers; the number of epochs; batch size; ρ is the regularization parameter;

Output: Trained model with decoding function ($\dot{x}$) returns α_j as reconstruction error; fw,d(x)~ x

1: start
2: arrange the data required for the training model with the dataset (d)
3: initialize the parameters β = (e_n, w_{en}, b_{en});
4: **for** P ∈ (1, 2, 3, 4… epochs)
5: **for** q∈ (1, 2, 3, 4… batch size)
6: **for** fw,d(x) ∀ x in d
7: transform the input layer vectors into their corresponding hidden layers in a series of encoder layers and compute output layer with decoding function [eq. 1 and eq. 2]
8: calculate the reconstruction error α_j by using eq. 4
9: update Adam's learning rate parameters β = (e_n, w_{en}, b_{en}); for each iteration
10: **end for**
12: **end for**
13: **end for**
14: train the model with the results of the above steps and return
15: stop

V. IMPLEMENTATION

A. Dataset Description

CT scan images of lung cancer are used as a dataset[1]. The dataset is a subset of the LUNA16 Grand Challenge[2]. The dataset is efficient enough to analyze the model because it contains the images exposed to a two-phase annotation process by four different radiologists. Thus, it makes the dataset suited for testing with an emphasis on identifying anomalies. Further, the images are adequately compressed, due to which no additional image compression techniques are used in the present study. A total of 297 images are separately marked for training and testing purposes. Convolutional autoencoders are implemented on the Spyder platform version 4.1.5 [3]by adopting a high-level neural network application package – Keras 2.3.0 [4] ,which runs on Tensorflow v2.4.1 [5] at the background. The code is written in python 3.8.8 [6]

B. Parameter Setting and Preprocessing

The details of the hyper-parameters used for the implementation are as follows: learning rate: 0.01, epochs: 40, batch size: 30, Adam optimizer parameters: alpha (learning rate) = 0.001, beta1 (exponential decay rate for the first estimate) = 0.9, beta2 (exponential decay rate for the second estimate) and epsilon (to overrule divide by zero error) = 10E-8, input images: 297, corresponding to 297 neurons in each hidden layer, sequential CNN model with kernel size = (3,3) at convolution layer and pool size = (2,2) at MaxPooling layer.

The images were preprocessed before the model is executed on the input. Those are; a function was called to load images from the folder onto an array variable. Further, images in the dataset had varying sizes. Thus, the height and width were rescaled to 512 pixels each to maintain uniformity throughout. The pixel values of the image (0 –black to 255 – white) are scaled between the ranges of 0 and 1 in the process called normalization (the ImageDataGenerator divides the pixel value by 255, for instance, 1/255 = 0.0039). This is performed because a neural network usually works with small weights used to update the neurons. If a large value is used, the network consumes a great deal of time, slowing down the learning process. With 40 epochs, the model attained an accuracy of 98% and an MSE value as low as 0.011. With every epoch, the model learns the features better with extra latent manifolds. The relevant features are then retained, and the characteristics that are not scalable for latent space representation are pruned.

C. Results

Out of 297 images, the dataset was split into three categories as training: 70% (207 images), validation: 10% (29 images), and test: 20% (61 images). The efficacy of the proposed system is measured at both the times – training and validation. The terms used are:

- Overall accuracy – accuracy is calculated at every epoch as, accuracy= images the system constructed correctly / the total number of images in each epoch (batch size).
- MSE – MSE defines an average square of the difference between the original input image and the image constructed by the model.

$$\text{MSE} = \frac{1}{B}\sum_{i=1}^{B}(X_i - \hat{X}_i)^2 \quad (5)$$

Here, 'B' indicates the batch size since the parameters are considered for individual batches. The error score of the original input image at the 'i' instance is given by X_i, and the error score of the reconstructed image at 'i' is provided by $\hat{X}_i$. MSE score of the anomalous data tends to be above the normal data threshold. The MSE scores for all samples are calculated to set the base threshold. The distribution of these MSE scores determines the threshold; 92% of the data was in the range of 0.011 to 0.6. The remaining 8% of the data had many variations in their MSE scores, such as 17.5, 2.5, 9.2, and 11.3, so on. Therefore, by looking at this distribution, the base threshold for anomaly detection was set as 0.7. The MSE score of the reconstructed images of the normal samples will be less than or equal to 0.7, and for abnormal images, the score will be greater than 0.7.

Of the 297 images, 23 images are identified as anomalies, with an MSE score greater than 0.7, and the 274 images are identified as normal samples, as demonstrated in Fig. 3(a). Initially, the accuracy was low even for a low MSE score; however, it is evident that, as the epoch progressed, the accuracy increased for normal data; however, the accuracy dropped as low as 11%, indicating a very high MSE score (17.5) for some data. Nevertheless, it is observed that the samples with high MSE scores have low accuracy values indicating the presence of the outliers. The accuracy achieved with low MSE scores was excelled, nearing 98%. The data with high MSE and low accuracy indicate the presence of the outliers, which were identified through the MSE scores.

- Val_loss: This is applied to the test data. val_loss is a good sign of how the model performs on the unseen data. Smaller val_loss indicates that there is no problem with overfitting. Consequently, if the model is trained heavily on the data, the val_loss increases as evidence of overfitting.
- Val_accuracy: The overall accuracy is an indicator of the classification performed on the training data. But for the test data, val_acc is crucial as it tests the accuracy of the unseen data. A neural network model is considered good when the val_loss starts decreasing, and the val_acc starts increasing [33], as shown in Fig. 3(b). Here, the number of examples used to calculate the loss/ error gradient is called a batch size or simply a batch. However, the training epoch indicates that the model has made learning for a randomly selected batch. As the validation loss is calculated in terms of samples, the term batch is used.

[1] https://www.kaggle.com/kmader/finding-lungs-in-ct-data
[2] https://luna16.grand-challenge.org/Data/
[3] https://www.spyder-ide.org/
[4] https://keras.io/
[5] https://www.tensorflow.org/
[6] https://www.python.org/

- Val_mse: The MSE score for the validation/test data

The overall evaluation of the proposed model is plotted in a line chart for the key terms explained so far. This is shown in Fig. 4. It must be noted that, as the epochs progress, the accuracy metrics increases, and the MSE values decreases. Additionally, val_loss is also reduced, indicating that the model is trained appropriately. Fig. 5 shows a set of images identified as anomalies and normal data. Once the outliers are removed, the image dataset is classified into either benign or malignant with simple neural network architecture [34].

The predicted output is put forward in the form of a confusion matrix in Fig. 6. Out of 297 input images, 259 images were correctly classified as benign (TP), and 22 out of 24 (actual number of malignant) images were classified as outliers (TN), 5 images that are non-benign (actual malignant) but are identified incorrectly as benign (FP) and 11 images were obtained incorrectly as malignant (FN). The ROC (Receiver-Operating-Curve) is plotted to determine the model performance based on predicting the probabilities of outcome (whether an image is an outlier or not) as illustrated in Fig. 7. The ROC is plotted against True Positive Rate (TPR) and False Positive Rate (FPR) for a wide range of threshold values. TPR – Recall – Sensitivity is given by, TPR = (TP) / (TP + FN) and FPR is given by, FPR = (FP) / (FP + TN). Area-Under-Curve (AUC) measures the degree of separation, which tells how capable the system is at distinguishing between the classes.

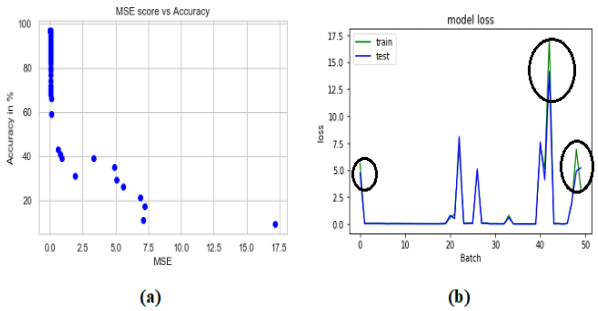

Fig. 3. (a) A Graphical Representation of Variation in the Accuracy and the MSE Scores. The Accuracy Increased, and the MSE Value is Dropped to a Minimum towards the End of 40 Epochs, (b) A Graphical Representation of Variation in Training and Test v_loss. As Observed, the val_ loss of Test Data is Slightly Reduced at Encircled Points.

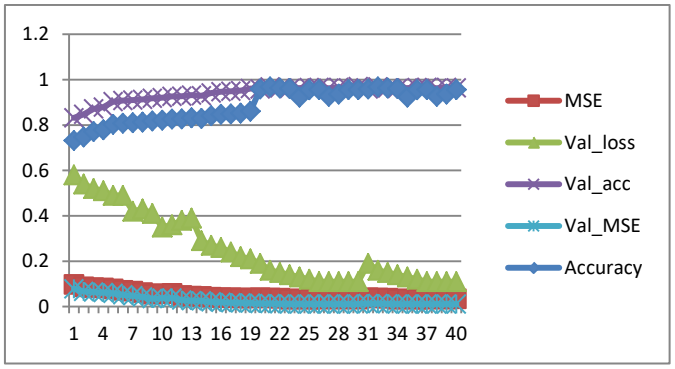

Fig. 4. The Evaluation Metrics Such as MSE, Val_loss, Val_acc, val_MSE, and Overall Accuracy Plotted across 40 Epochs.

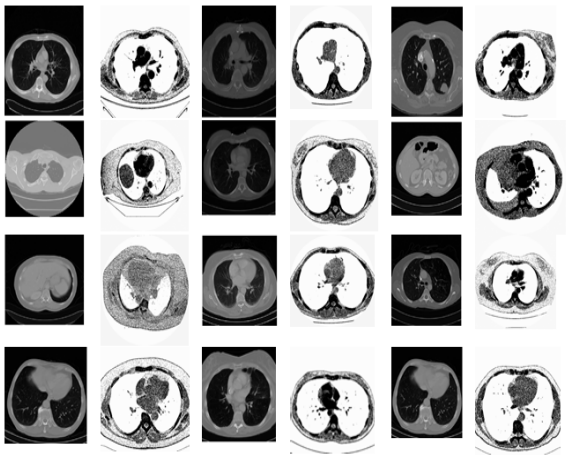

Fig. 5. A Series of Data for both Anomalous (Black Background) and Normal (White Background) as Identified by the Proposed Model.

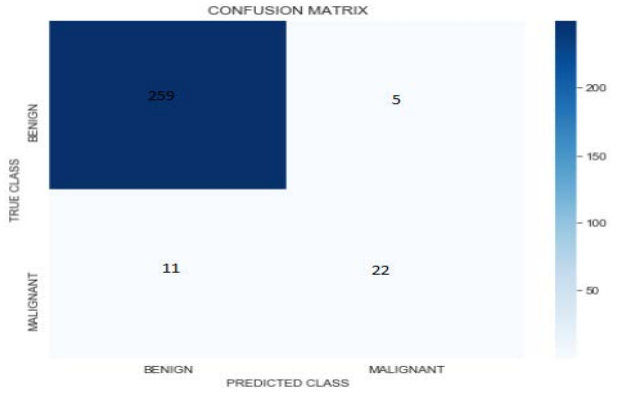

Fig. 6. A Confusion Matrix for the Two Classes – benign and Malignant Plotted against the True and Predicted Classes. Here, 259 Indicates TP, FP = 5, FN = 11 and TN = 22. The Ranking is shown for all the 297 Input Samples.

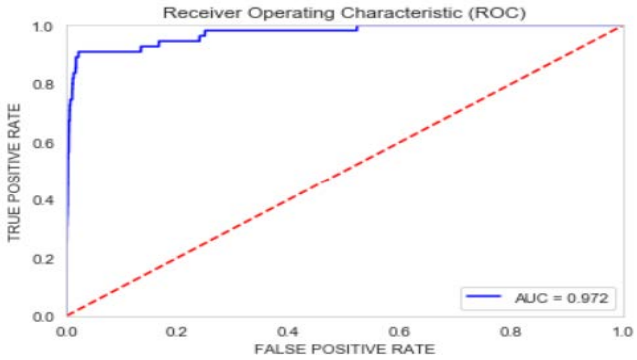

Fig. 7. ROC Curve for Cumulative Results of the Classification Task. The Value of AUC = 0.972 (97%) Reveals that the Model is Excellent in Distinguishing between the benign and Malignant Classes.

D. Comparison of Various Outlier Detection Methods with the Proposed Model

In this subsection, the proposed model is compared with the classical state-of-art systems. The result of this comparison is described in Table I. The proposed model outperformed the other conventional methods by achieving 98% accuracy. This indicates that the model can be well adapted even for distinct datasets with complex structures.

TABLE I. COMPARISON OF THE PROPOSED MODEL WITH THE OTHER STATE-OF-ART TRADITIONAL SYSTEMS IN TERMS OF ACCURACY

	Accuracy
Gritli. et.al in [14]. 3D AlexNet architecture.	97%
Sharma. et.al in [15]. CNN-based architecture	77%
Rasha. et.al [16]. Firefly algorithm with SVM	78%
Laura Beggel. et al. in [17]. Adversarial autoencoder-based	0.62 [Under 0.62]
AnoGAN deep convolution using adversarial network [35]	84%
Our proposed method	**98% - outlier detection** **97.2% - classification task**

VI. DISCUSSIONS

Like any other expert system, the proposed model also deals with some limitations. The model is highly dependent on the training data. As a result, when unseen data – a typical healthy heart image – was fed as an input, the system calls it an anomaly. This could be a potential problem mainly when the corpus is more generic than domain-specific. The proposed model considers only the spatial features, thereby removing the temporal characteristics of the image in the cleaning step. Thus, the edges and variations in the local binary pattern of the images are skipped, leading to misrepresentation of features sometimes. Interestingly, the global minimum MSE score is 0.01, and it cannot be reduced even with further training. This hypothesis helped to shorten the input size and exemplify the latent space representation. Additionally, a sparse hierarchical model is witnessed in most activations, mainly when the spatial features are selected. Further, the complex representation is brought down to lower dimensions in the encoder and later decoded into an original image. During this transformation, the model may memorize the data during the training process leading to overfitting. Therefore, the proposed method restricts the number of neurons in the core layer, usually half of the number of input variables in the network. This will ensure that the model is learning the key patterns, rules, and essential features from the input data. It is imperative to note that no training labels are used in the model, making it completely unsupervised. However, each neuron at the hidden layers is driven by the data on hand that makes the system data-reliant. Thus, when the input features change, the activation function triggers different neurons and results in a different output through the network. While the latent space representation stress enough on the encoding and decoding process, the regularization used in the network minimizes the error rate through the L1 regularization technique. Though the proposed model performs reasonably well, there is still room for improvement. For instance, the gradient-weighted activation mapping technique could be used to obtain visual explanations of the predictions made by the system, and using a larger dataset could further improve the performance.

The future direction of this research study is to identify the nodule location and size measurement using Deep NN techniques and later categorize it into different cancer stages. The present work could be implemented on different types of autoencoder for a complex dataset and study the performance. The hyper-parameters may be tweaked to refine the CNN model and check if the accuracy is improved. The outliers can be grouped into different clusters and analyze their behavior in each set. Alongside, the feature rules can be generated to highlight the anomaly score of each group to understand the depth of anomalies present in the data. The accuracy could be improved further by choosing a giant database such as LUNA16 or LIDC/IDRI. The results obtained will help clinicians detect cancer more accurately with an anomaly-free dataset.

VII. CONCLUSION

A study on outliers in medical data has been one of the leading research concerns over the past few years. By and large, the anomalies in the medical data are inevitable but impose complications if left unnoticed. Previously known anomaly detection approaches using PCA are equally efficient; however, PCA attempts to uncover the lower-level features of the input data, but autoencoders learn features from the data having higher dimensions with any complex and non-linear structures. With the help of an encoder and a decoder, clustered in multiple convolutional layers, the autoencoders efficiently remove the outliers without any training labels in the dataset. The encoder absorbs significant features of the images. The original image is reconstructed at the decoder side. Of the 297 images, 23 images are identified as anomalies, with an MSE score greater than 0.7, and the 274 images are identified as normal samples. With 40 epochs, the model attained an accuracy of 98% and an MSE value as low as 0.011. With every epoch, the model learns the features better with extra latent manifolds. The outputs are further classified into benign and malignant. The confusion matrix indicates a good classification of the two classes. Out of 297 input images, 259 images were correctly classified as benign, and 22 out of 24 images were classified as outliers, 5 images that are non-benign but are identified incorrectly as benign, and 11 images were obtained wrongly as malignant. The ROC-AUC curve showed 97.2% efficiency on the classification task. Thus, autoencoder could be a one-stop destination to remove the outliers from complex multivariate data.

ACKNOWLEDGMENTS

One of the authors (RS) acknowledges the Department of Science and Technology – Science and Engineering Research Board (DST-SERB), New Delhi, India, for providing a research grant and postdoctoral fellowship (NPDF, sanction order no PDF/2019/000254). The authors would like to thank the Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India, for providing complete support to execute this work.

AVAILABILITY OF DATA AND MATERIALS

For ease of use of the proposed methodologies for other researchers and academicians, the entire code and datasets with relevant results have been deposited in the GitHub repository, available (https://github.com/RashmiSKarthik/Outlier-Detection-Anomaly-Detection). The code can be used

for other cancer datasets, and the model works efficiently on any Python platform. The researchers may reproduce any copyrighted material with appropriate citations of this work. The repository is for restricted use and in private access mode. However, one of the authors (RS) wishes to provide access to those who need the implementation model adopted in this research paper. Kindly write a mail to drrashmis64@gmail.com to obtain access to this repository.

DECLARATION OF CONFLICTING INTERESTS

The author(s) declare that there are no potential conflicts of interest concerning this current research, authorship, and/or publication of this work.

REFERENCES

[1] Liu, H., Li, J., Wu, Y., & Fu, Y. (2018, January 5). Clustering with outlier removal. *ArXiv*. arXiv. https://doi.org/10.1109/tkde.2019.2954317.

[2] Yu, W., Ding, Z., Hu, C., & Liu, H. (2019). Knowledge Reused Outlier Detection. *IEEE Access*, *7*, 43763–43772. https://doi.org/10.1109/ACCESS.2019.2906644.

[3] Lodhia, Z., Rasool, A., & Hajela, G. (2017). A survey on machine learning and outlier detection techniques. *International Journal of Computer Science and Network Security*, *17*(5), 271–276.

[4] Almardeny, Y., Boujnah, N., & Cleary, F. (2020). A Novel Outlier Detection Method for Multivariate Data. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2020.3036524.

[5] Gribl, A., & Petrinovic, D. (2021). A Robust Method for Gaussian Profile Estimation in the Case of Overlapping Objects. *IEEE Access*, *9*, 21071–21084. https://doi.org/10.1109/ACCESS.2021.3055282.

[6] Bawono, A. H., & Bachtiar, F. A. (2019). Outlier Detection with Supervised Learning Method. In *Proceedings of 2019 4th International Conference on Sustainable Information Engineering and Technology, SIET 2019* (pp. 306–309). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/SIET48054.2019.8986101.

[7] Wang, H., Bah, M. J., & Hammad, M. (2019). Progress in Outlier Detection Techniques: A Survey. *IEEE Access*, *7*, 107964–108000. https://doi.org/10.1109/ACCESS.2019.2932769.

[8] Ramchandran, A., & Sangaia, A. K. (2018). Unsupervised anomaly detection for high dimensional data-An exploratory analysis. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications* (pp. 233–251). Elsevier. https://doi.org/10.1016/B978-0-12-813314-9.00011-6.

[9] Legrand, A., Niepceron, B., Cournier, A., & Trannois, H. (2019). Study of Autoencoder Neural Networks for Anomaly Detection in Connected Buildings. In *2018 IEEE Global Conference on Internet of Things, GCIoT 2018*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/GCIoT.2018.8620158.

[10] Chang, S., Du, B., & Zhang, L. (2019). A Sparse Autoencoder Based Hyperspectral Anomaly Detection Algorihtm Using Residual of Reconstruction Error. In *International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 5488–5491). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IGARSS.2019.8898697.

[11] Cook, A. A., Misirli, G., & Fan, Z. (2020, July 1). Anomaly Detection for IoT Time-Series Data: A Survey. *IEEE Internet of Things Journal*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/JIOT.2019.2958185.

[12] Yao, D. (Daphne), Shu, X., Cheng, L., & Stolfo, S. J. (2017). Anomaly Detection as a Service: Challenges, Advances, and Opportunities. *Synthesis Lectures on Information Security, Privacy, and Trust*, *9*(3), 1–173. https://doi.org/10.2200/s00800ed1v01y201709spt022.

[13] Ahmed, T., Parvin, Mst. S., Haque, M. R., & Uddin, M. S. (2020). Lung Cancer Detection Using CT Image Based on 3D Convolutional Neural Network. *Journal of Computer and Communications*, *08*(03), 35–42. https://doi.org/10.4236/jcc.2020.83004.

[14] Neal Joshua, E. S., Bhattacharyya, D., Chakkravarthy, M., & Byun, Y. C. (2021). 3D CNN with Visual Insights for Early Detection of Lung Cancer Using Gradient-Weighted Class Activation. *Journal of Healthcare Engineering*, *2021*. https://doi.org/10.1155/2021/6695518.

[15] Sharma, S., Kaur, M., & Saini, D. (2019). Lung cancer detection using convolutional neural network. *International Journal of Engineering and Advanced Technology*, *8*(6), 3256–3262. https://doi.org/10.35940/ijeat.F8836.088619.

[16] Lung Anomaly Detection System (LADS) Using SVM based on Firefly Algorithm. (2017). *International Journal of Science and Research (IJSR)*, *6*(7), 540–544. https://doi.org/10.21275/art20175294.

[17] Beggel, L., Pfeiffer, M., & Bischl, B. (2020). Robust Anomaly Detection in Images Using Adversarial Autoencoders. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 11906 LNAI, pp. 206–222). Springer. https://doi.org/10.1007/978-3-030-46150-8_13.

[18] Buitrago, N. R. S., Tonnaer, L., Menkovski, V., & Mavroeidis, D. (2018, November 2). Anomaly detection for imbalanced datasets with deep generative models. *ArXiv*. arXiv.

[19] Astaraki, M., Toma-Dasu, I., Smedby, Ö., & Wang, C. (2019). Normal Appearance Autoencoder for Lung Cancer Detection and Segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 11769 LNCS, pp. 249–256). Springer. https://doi.org/10.1007/978-3-030-32226-7_28.

[20] Albahar, M. A., & Binsawad, M. (2020). Deep Autoencoders and Feedforward Networks Based on a New Regularization for Anomaly Detection. *Security and Communication Networks*, *2020*. https://doi.org/10.1155/2020/7086367.

[21] Albahar, M. A., & Binsawad, M. (2020). Deep Autoencoders and Feedforward Networks Based on a New Regularization for Anomaly Detection. *Security and Communication Networks*, *2020*. https://doi.org/10.1155/2020/7086367.

[22] Meng, Q., Catchpoole, D., Skillicom, D., & Kennedy, P. J. (2017). Relational autoencoder for feature extraction. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 2017-May, pp. 364–371). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IJCNN.2017.7965877.

[23] Tan, Y., Jin, B., Nettekoven, A., Chen, Y., Yue, Y., Topcu, U., & Sangiovanni-Vincentelli, A. (2019). An encoder-decoder based approach for anomaly detection with application in additive manufacturing. In *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019* (pp. 1008–1015). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICMLA.2019.00171.

[24] Cao, J., Su, Z., Yu, L., Chang, D., Li, X., & Ma, Z. (2019). Softmax Cross Entropy Loss with Unbiased Decision Boundary for Image Classification. In *Proceedings 2018 Chinese Automation Congress, CAC 2018* (pp. 2028–2032). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CAC.2018.8623242.

[25] Muthukrishnan, R., & Rohini, R. (2017). LASSO: A feature selection technique in predictive modeling for machine learning. In *2016 IEEE International Conference on Advances in Computer Applications, ICACA 2016* (pp. 18–20). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICACA.2016.7887916.

[26] Sangari, A., & Sethares, W. (2016). Convergence Analysis of Two Loss Functions in Soft-Max Regression. *IEEE Transactions on Signal Processing*, *64*(5), 1280–1288. https://doi.org/10.1109/TSP.2015.2504348.

[27] Osman, H., Ghafari, M., & Nierstrasz, O. (2017). Automatic feature selection by regularization to improve bug prediction accuracy. In *MaLTeSQuE 2017 - IEEE International Workshop on Machine Learning Techniques for Software Quality Evaluation, co-located with SANER 2017* (pp. 27–32). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/MALTESQUE.2017.7882013.

[28] Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

[29] Anusha, A., Rao, I. S., & Student, M. T. (2018). A Study on Outlier Detection for Temporal Data. *International Journal of Engineering Science and Computing*, *8*(3), 16354–16356. Retrieved from http://ijesc.org/.

[30] Karadayi, Y., Aydin, M. N., & Ög˘renci, A. S. (2020). A hybrid deep learning framework for unsupervised anomaly detection in multivariate spatio-temporal data. *Applied Sciences (Switzerland)*, *10*(15). https://doi.org/10.3390/app10155191.

[31] Xin, M., & Wang, Y. (2019). Research on image classification model based on deep convolution neural network. *Eurasip Journal on Image and Video Processing*, *2019*(1). https://doi.org/10.1186/s13640-019-0417-8.

[32] Alrawashdeh, K., & Purdy, C. (2018). Fast Activation Function Approach for Deep Learning Based Online Anomaly Intrusion Detection. In *Proceedings - 4th IEEE International Conference on Big Data Security on Cloud, BigDataSecurity 2018, 4th IEEE International Conference on High Performance and Smart Computing, HPSC 2018 and 3rd IEEE International Conference on Intelligent Data and Security, IDS 2018* (pp. 5–13). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/BDS/HPSC/IDS18.2018.00016.

[33] Sadaf, K., & Sultana, J. (2020). Intrusion detection based on autoencoder and isolation forest in fog computing. *IEEE Access*, *8*, 167059–167068. https://doi.org/10.1109/ACCESS.2020.3022855.

[34] Xin, M., & Wang, Y. (2019). Research on image classification model based on deep convolution neural network. *Eurasip Journal on Image and Video Processing*, *2019*(1). https://doi.org/10.1186/s13640-019-0417-8.

[35] T. Schlegl, P Seeböck, S.M.Waldstein, U.Schmidt, G. Langs, “Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery”, (2017).