International Journal of Innovative Research in Engineering
Volume 3, Issue 3 (May-June 2022), PP: 461-468.
www.theijire.com
ISSN No: 2582-8746

Anomaly Detection Using Auto encoders

Deepak Kumar C R [1], Guhanesvar M [2], Tilak Vijayaraghavan S [3], Jona .J.B [4],Gunasekaran S.A [5]
[1,2,3] M.Sc. (integrated) Decision and computing Science, Coimbatore Institute of Technology, Coimbatore, India.
[4] Associate, Professor, Department of Computer Applications, Coimbatore Institute of Technology, Coimbatore, India.
[5]Assistant Professor, Department of Computer Applications, Coimbatore Institute of Technology, Coimbatore, India.

How to cite this paper: Deepak Kumar C R [1], Guhanesvar M [2], Tilak Vijayaraghavan S [3], Jona .J.B [4],Gunasekaran S.A [5], "Anomaly Detection Using Autoencoders", IJIRE-V3I03,461-468

Abstract: *The ECG (Electrocardiogram) which records the electrical signal from the heart to check for different heart conditions. The ECG shows how a person's health condition is using its signals, we can determine the person's health. The Autoencoders is an unsupervised neural network that learns. Autoencoders learn the data input patterns and reconstruct the output resembling the input. Using this, anomalies in the heart can be detected.*

Key Word: *ECG; Autoencoder; Unsupervised; impulses; neural network*

I.NTRODUCTION

Electrocardiography is the process of producing an electrocardiogram (ECG or EKG), a recording of the heart's electrical activity. It is an electrogram of the heart which is a graph of voltage versus time of the electrical activity of the heart using electrodes placed on the ski-n. These readings are used to monitor a person's heartbeat. Any major anomaly can be easily identified while others may not be. Autoencoders help in going the extra mile when it comes to finding such anomalies. AutoEncoder is a generative unsupervised deep learning algorithm used for reconstructing high-dimensional input data using a neural network with a narrow bottleneck layer in the middle which contains the latent representation of the input data.

II. DATA

The dataset is ecg 5000 data set which is taken from time series classification.com.ECG is electrocardiogram. The first electrocardiogram checks how the heart is functioning by measuring electrical activity of the heart. In each heartbeat an electrical impulse travels through the heart that creates muscles to squeeze and pump blood to the heart that's what it does.Typically in our body this electrocardiogram captures the electrical impulse that can help doctors determine if the heart is functioning normally or irregularly. The dataset contains information of both normal activity and irregular activity of the heart. This dataset contains about 5000 rows. Each row corresponds to a single complete ECG of a patient. Every single ECG is composed of 140 data points(readings). The naming of the columns starts with zero,one,two,three,four and so on as of the default column naming given by pandas.

Data structure: Columns 0-139 contain the ECG data point for a particular patient. These are floating point numbers.

Data label: The label which shows whether the ECG is normal or abnormal. It is a categorical variable with value either 0 or 1.The label at the last is the target column.

III.PREPROCESSING

First inorder to fit the data into the model you need to process the data. Here, since the data is already in the numerical format there is no need for data encoding.Data encoding is the process of converting categorical values into numercial.By this conversion we can easily fit the data into our model. The missing values were check and for all columns it is less than 5% so there is no need for imputation.Imputation is the method of replacing the missing value by a substitutevalue inorder to retain the information that the numerical value posses.

Target Imbalance

Imbalanced data refers to those types of datasets where the target class has an uneven distribution of observations, i.e one class label has a very high number of observations and the other has a very low number of observations. Here the target classes 1 and 0 are normal and abnormal heart impulses. If the target balancing is not checked or dealt with then it may result in

low predictive accuracy for the infrequent class. Cost-sensitive learning is a common approach to solve this problem. There were about 2919 data points which had a normal ecg rate and 2079 data points which had abnormal ecg rate. So,the dataset is balanced.

Data Splitting

The train-test split is used to estimate the performance of machine learning algorithms that are applicable for prediction-based Algorithms/Applications. This method is a fast and easy procedure to perform such that we can compare our own machine learning model results to machine results. We perform a 80-20 train-test split. All the columns are taken for training from 0 to 141. Only the first column is taken as the target column. Basically passing x and y values , with x as the feature columns and y as the label column.

Data Normalization

Data normalization is the part of machine learning to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. First we calculate the minimum and maximum of the data .We normalize the data by the Z-score method. Each data point is subtracted from the minimum value and divide by minimum value and maximum value. This makes the data into a normal distribution which makes it easy for model fitting.

Data Scaling

Data scaling is done to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values. We use the Min-Max scaler to perform data scaling. It scales the range of the features between 0 and 1. This doesn't work well for data that is not normally distributed. So, before data scaling we perform data normalization. It does not change the shape of the distribution of the features.

$$x' = x - x_{min} / x_{max} - x_{min}$$

Data Visualization:

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.The charts show how data differs from the ones obtained using auto encoders. In Anomaly detection the data distribution has to be separate than the normal patterns. Since auto encoder is an unsupervised learning technique. The labels will not be fed into the model it is going to learn on its own.

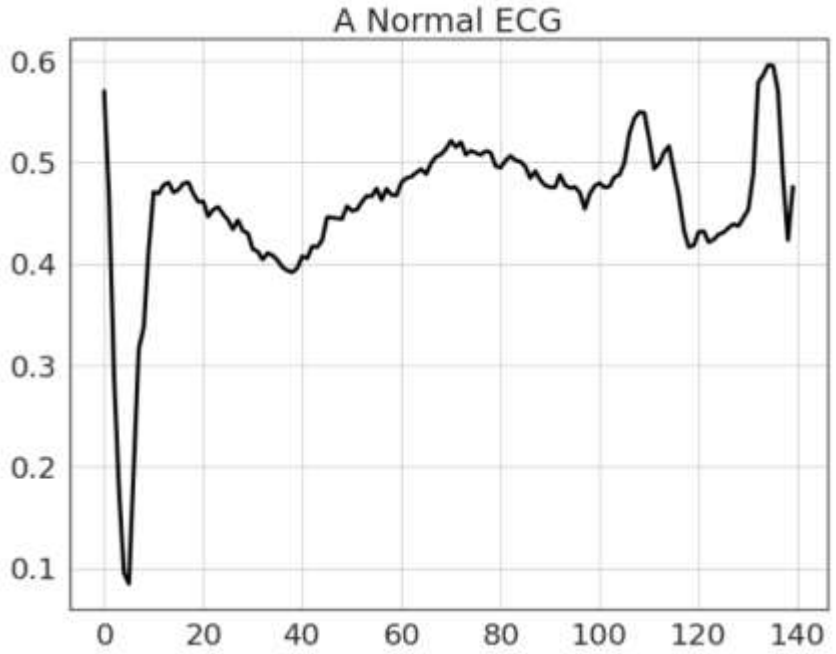

Figure 3.1 Normal ECG readings

Figure 3.1 is the normal ECG data pattern with all the data points of the normal readings plotted. The readings are for all the 140 observations. The pattern is quite normal with up and down trends,initially there is a drop because that is the start of the ECG measurement. Avoiding that otherwise it's a normal cyclic circle.

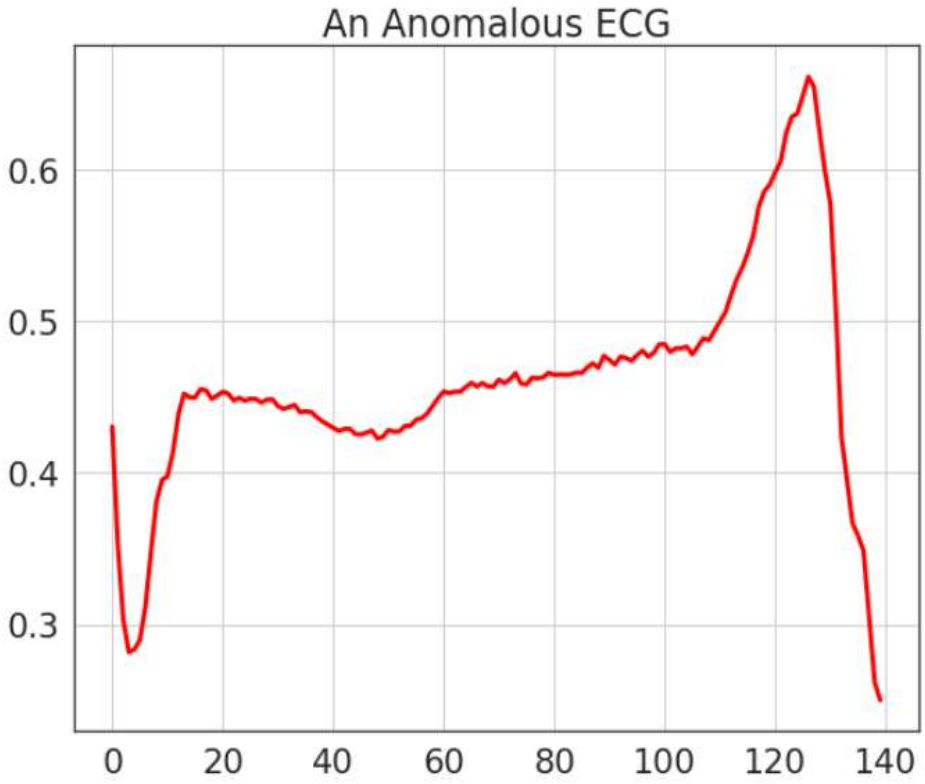

Figure 3.2 Anomalous ECG readings

Figure 3.2 is the abnormal ECG data pattern with all the data points of the anomaly readings plotted. The readings are for all the 140 observations. The anomaly data starts off normally but there is a sudden drop towards the last observation. Inferring from the two graphs there is a clear separation between the normal and the anomaly trends.

IV.AUTOENCODERS

Data Comprehension is to convert our input into smaller representations that we recreate to a degree of quality. The smaller representation is what would be passed around and when anyone needed the original they would reconstruct it from the smaller representation. Autoencoders are unsupervised neural networks that use machine learning to do this comprehension for us. The aim of an autoencoder is to learn a compressed distributed representation for the given data typically for the purpose of dimensionality reduction. Now that we have principal component analysis, then why is there a need for Autoencoders? An Autoencoder can learn non-linear transformations unlike PCA with a non-linear animation function and multiple layers now it doesn't have to learn dense layers so it can use Convolutional layers to learn which could be better for video image and series data now it may be more efficient in terms of model parameters to learn several layers with an autoencoder rather than learn one huge transformation with PCA an autoenncoder also gives a representation as the output of each layer and having multiple representations of different dimensions is always useful so an autoencoder could let you make use of pe-trained layers from another model to applytransfer learning to prime the encoder or the decoder despite the fact that the practical applications of autoencoders were pretty rare sometime back today data denoising and dimensionality reduction for data visualisation are considered as two main interesting practical applications of autoencoders now with appropriate dimensionality an sparsity constraints autoencoders can learn data projections that are more interesting than PCA or other basic techniques it also provides a more accurate output when compared to PCA now autoencoders are simple learning networks that aim to transform inputs into outputs with the minimum possible error this means we want the output to be as close to input as possible an autoencoder neural network is basically an unsupervised machine learning algorithm that applies back propagation setting the target values to be equal to the inputs so let's have a look at some of the key features about autoencoders so it is an unsupervised machine learning algorithm that is similar to PCA but minimizes the same objective function an autoencoder is a neural network whose target output is its input autoencoder although is quite similar to PCA but they are flexible when compared to the other autoencoders can represent both linear and non linear transformation in encoding but PCA can only perform linear transformation so now let's have look at the components of encoders so there are basically there are basically three main layers:

Encoder

It is the art of the network that compresses the input into a latent space representation; the encoder layer encodes the input dimensions as a compressed representation in a reduced dimension now the compressed image typically looks garbled nothing like the original image.

Code:

It represents the latent space now code is the part of the network that represents the compressed input fed to the decoder.

Decoder:

This layer basically decodes the encoded input back to the original dimension. The decoded input is a lossy reconstruction of the original input now it reconstructs the input from the latent space representation.

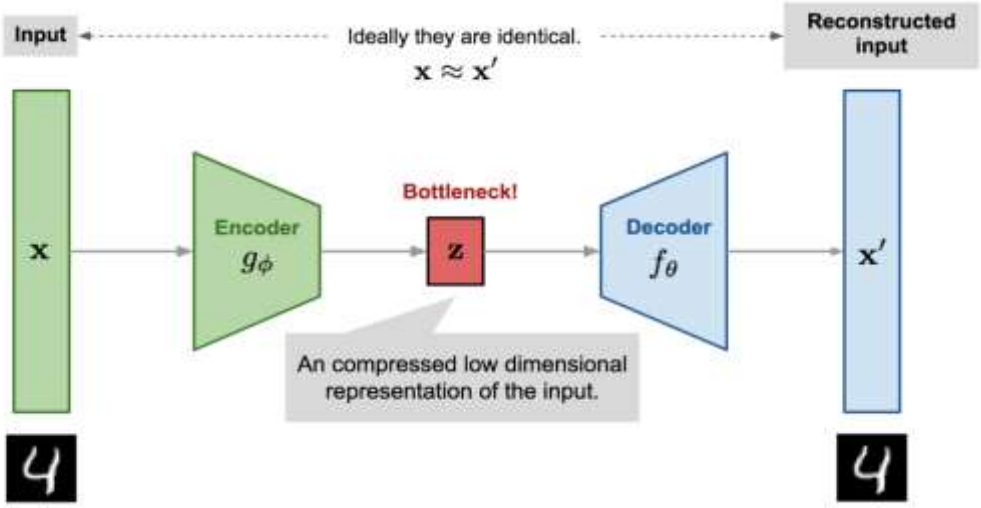

Figure 4.1 Autoencoder architecture

The auto encoder learns to produce the same output as the input as much as possible. Through this learning process, the representation of the input can be effectively compressed in the latent space. In other words, it compresses the dimensions and stores the representation of the input in the latent space.

V. MODEL

Model that learns from the data and we will check whether there is any Anomaly present in the data. The data has a separation of Normal and Anomaly data. The Reconstruction error will determine whether the data is an Anomaly or not. We have created the Autoencoder using the Sequential model approach. The Encoder layer is constructed with four Dense layers with the units 64,32,24,16 respectively. The Bottleneck layer has a Dense layer which has 8 units. The Decoder layer has four Dense layers with the units 16,24,32,64 respectively. The output layer which is the classifier layer uses one Dense layer with 140 units as the data has 140 inputs. The activation function used for all the layers except the output or the classifier layer is Rectified Linear unit, this activation function is a piecewise linear function that will output the input directly if it is positive otherwise it will output zero. The reasoning is that ReLU will clip a portion of the data before applying the activation function, the data vector is multiplied by a weight matrix. This matrix may contain negative values, so the negative inputs may yield positive values before the ReLU is applied, and conversely, positive inputs may yield negative values. Therefore, not applying the activation function on the input values directly. The Sigmoid activation function is used in the output layer or the classifier layer to get output from the decoder as an Anomaly or normal. In the model compiler layer we have given the batch size as 512 data points so that is a large batch size to train their model as it allows computational speedups from the parallelism of GPUs.

VI. IMPLEMENTATION

Python is one of the high-level, interpreted, general-purpose programming languages. Its design philosophy emphasizes code readability with the employment of serious indentation. Python is dynamically-typed and garbage-collected. Python supports a variety of programming paradigms which include structured, object-oriented and functional programming. It's often described as a "batteries included" language thanks to its comprehensive standard library.

Libraries used:

Python as the scripting tool to implement auto encoders. A series of python libraries like tensorflow, pandas, numpy etc. Jupyter notebooks, which is one of the most popular open sourceintegrated development, was used for creating the autoencoders.

Pandas:

Pandas is one of the software libraries written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and statistics. Pandas supports a wide variety of data/file formats ranging from text/csv file to sql databases. Pandas also provides the necessary tools/APi to enable data preprocessing amongst other notable features essential for analysis.

Numpy:

NumPy is another library for Python, adding support for giant, multi-dimensional arrays and matrices, together with an outsized collection of high-level mathematical functions to control on these arrays. These arrays have proven to be way faster than conventional python lists/tuples. Numpy also provides typecasting between various data types, also conversion of image to array and vice-versa.

TensorFlow:

TensorFlow is a free and open-source software library by Google mainly developed for machine learning and huge data computing. It is often used across a variety of tasks but features a particular specialization in training and inference of deep neural networks. This particular library has provided the foundation for various ML and AI projects and also provides means to deploy such apps with other corresponding libraries of its own.

Matplotlib:

Matplotlib is one of the plotting libraries for Python data visualization and provides various numerical and mathematics extensions for NumPy and other libraries. This library goes hand-in-hand with many data types and supports various visualizations like scatter plot, line plot amongst other visuals.

Keras:

Keras is another open-source software library that is used which provides an interface for artificial neural networks implementation. Keras goes hand-in-hand with the TensorFlow library. Keras provides the necessary tools, metrics for creating the ML model. Keras also provides the necessary tools to measure the performance of a ML model.

Sequential model:

A sequential model built using keras aims at serial processing of data through a series of dense or sparse layers.

ReLu Activation*:*

Rectified Linear activation function is one of the most used activations function in deep learning owing to its ability to give a positive output for any given input.

$$f(x) = x^+ = \max(0, x)$$

Sigmoid Activation*:*

Sigmoid activation function is one of the activation functions which gives binary output for a given input.

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x).$$

Sklearn:

Scikit-learn (formerly scikits.learn and also referred to as sklearn) is one of the free software machine learning libraries for the Python artificial language. Scikit-learn features various classification, regression and clustering algorithms including svm, random forests, gradient boosting, k-means, and is intended to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

The model as every auto encoder consists of a series of encoders and decoders along with a bottle neck layer. Using the keras 'layers' module, a series of rectified linear layers were added and funnel into a single bottleneck layer which was then decompressed to the output layer which then gives the signal as an anomaly or not. The results of the model were summarized with the scikit learn library and visualized using the matplotlib library. A clear idea is achieved when viewing the output chart of the model which shows the distinction between an anomaly and a normal signal.

VII. RESULTS

The Scikit learn library along with keras provides the necessary tools to analyze the model to generate interpretable results. These include accuracy metrics like accuracy and Precision.Accuracy and precision are two measures of observational error. Accuracy is how close or far off a given set of measurements (observations or readings) are to their true value, while precision is how close or dispersed the measurements are to each other.Accuracy is a description of statistical bias of a given measure of central tendency; low accuracy causes a difference between a result and a true value.Precision on the other hand is

how close the values are with themselves. A model can have high precision while having low accuracy meaning that the model as a whole was targeting one aspect which is not the target.Recall is quite similar to precision, it corresponds to the number of true positives retrieved among the total retrieved.

```
Accuracy = 0.944
Precision = 0.9921875
Recall = 0.9071428571428571
```

The model created here has high accuracy and high precision meaning that the model aims the target and doesn't miss the target. There were also a series of graphs which were formulated to see the difference between the normal ecg and an anomalyecg.

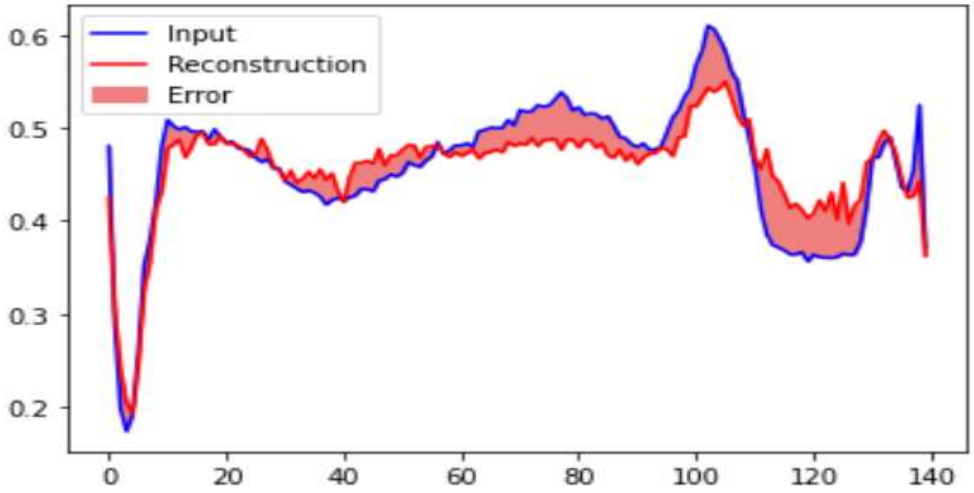

Figure 7.1 Reconstruction of a normal ecg

Figure 7.1 is an example of reconstruction of an ecg graph. Data and the decoder output the normal data is the actual data that is first passing. The decoder data is the reconstructed data which is but not exactly the same there are some errors to it. The blue color line is the normal data and the red line is the reconstructed one. The gap between the red and blue line are the reconstructed errors.

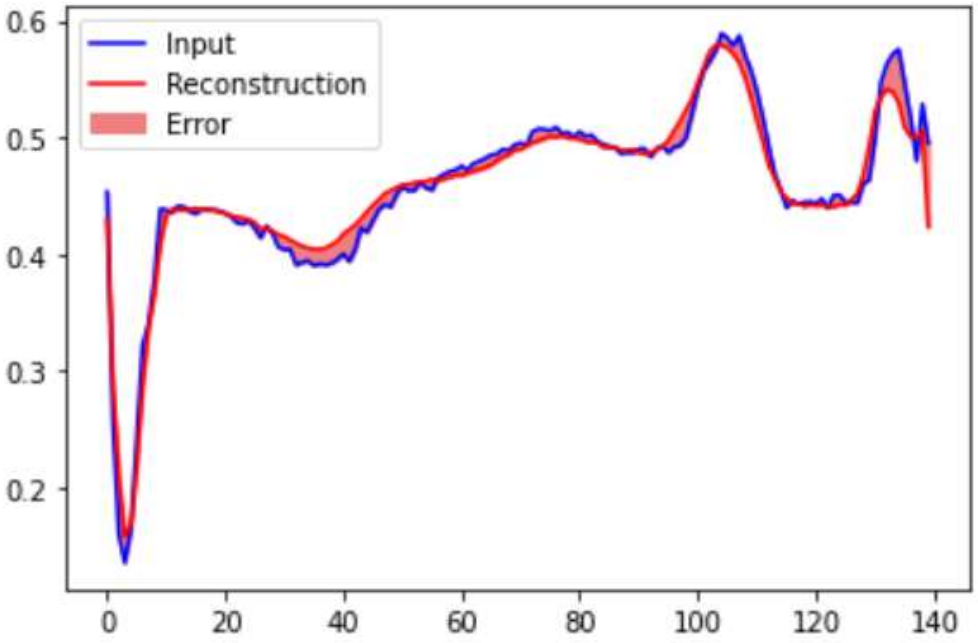

Figure 7.2 Reconstruction of a normal ecg

Figure 7.2 is reconstruction of a normal ecg with less error than the previous reconstruction. The input data fits well with the reconstructed data.

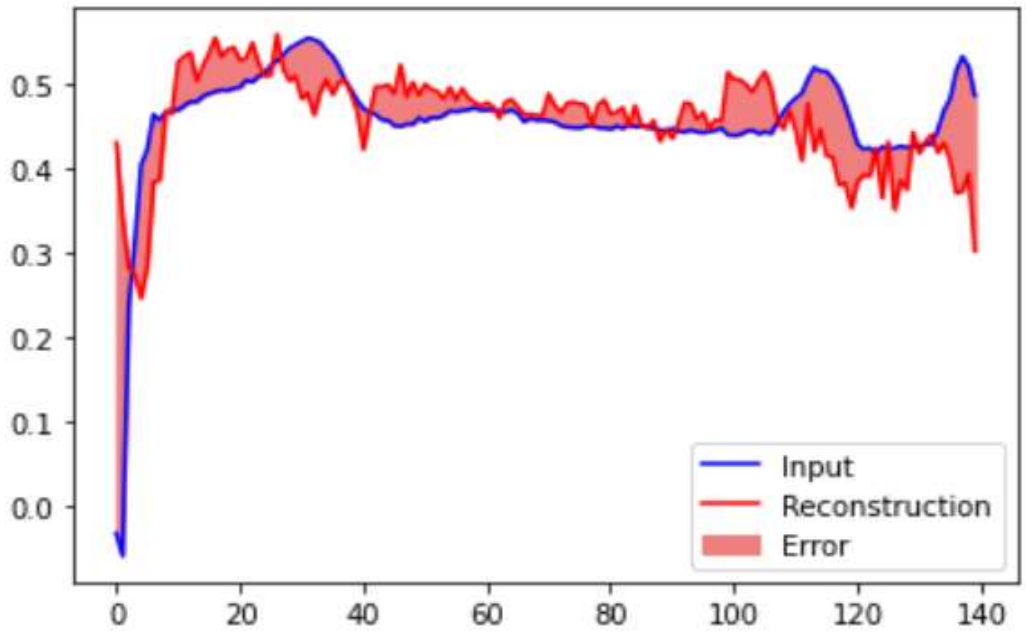

Figure 7.3 Reconstruction of an anomaly ecg

Figure 7.3 is a reconstruction of an anomaly ecg with high error indicating that it's an anomaly. Figure 7.1,Figure 7.2 and Figure7.3 are different samples taken for testing.

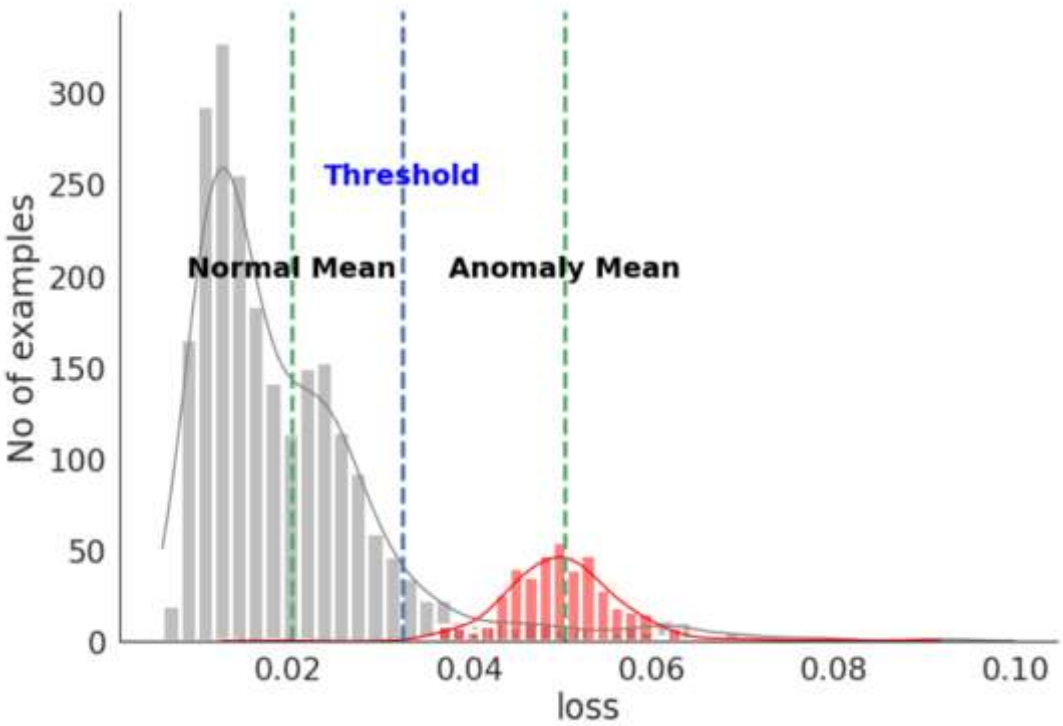

Figure 7.4 A Threshold graph between normal ecg and an anomaly

This graph shows the threshold for a normal and an anomaly ecg there by having a clear plane of separation between them. It seems that the threshold was set by considering normal and anomaly distributions. In particular, it seems that recall is more important than precision for anomaly detection. In actual application, it may be necessary to consider fine-tuning the threshold.

VIII. CONCLUSION

The normal classes are predicted better than anomaly classes right whenever you move the threshold the one class is going to benefit another class is going to impact it so this is how anomaly detectors using auto enoder are built. Adding many layers into the model has reduced the reconstruction loss and improved the model accuracry.You can use a lot of different architecture. This can be any architecture, it can be LSTM(long short term memory) architecture or it can be a 1D Convolution architecture. But this is the base for everything we need to build an anomaly detector using a n auto encoder..This can be used

in various heart rate monitoring devices to simultaneously check the person's heart impulses and warn them if there are any abnormalities in them. Since many devices nowadays come with a default application to measure the heart impulses this method can be used in them.

References

[1]. *JunhaiZhai, Sufang Zhang, Junfen Chen and Qiang He authored Autoencoder and its various variants published 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)2019*

[2]. *João Pereira and Margarida Silveira authouredUnsupervised representation learning and anomaly detection in ECG sequences published in 2019 International Journal of Data Mining andBioInformatics Vol.22, No. 4*

[3]. *C. Venkatesan; P. Karthigaikumar; Anand Paul; S. Satheeskumaran; R. Kumaranalyszed ECG Signal Preprocessing and SVM Classifier-Based Abnormality Detection in Remote Healthcare Applications published in IEEE Access(Volume:6)2018*

[4]. *J. A Uriguen and B. Garcia-Zapirain "EEG artifact removal-state-of-the-art and guidelines" J Neural Eng. vol. 12 no. 3 June 2015.*

[5]. *Jinwon An and Sungzoon Cho lectured "Variational Autoencoder based Anomaly Detection using Reconstruction Probability" at SNU Data mining center, 2015*

[6]. *D. Tax and R. Duin "Outliers and data descriptions" Proc. 7th Annual Conf. Advanced School for Computing and Imaging (ASCI) 2001*

[7]. *V. J. Hodge and J. Austin "A survey of outlier detection methodologies" Artificial Intelligence Review vol. 22 2004.*

[8]. *M. Allvlahamdy and H. Bryan Riley "Performance Study of Different Denoising Methods for ECG Signals" Proceedings The 4th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2014) pp. 325-332 2014.*

[9]. *Hu ZhengbingSu Jun V. Jotsov O. Kochan M. Mykyichuk R. Kochan et al. "'Data Science Applications to Improve Accuracy of Thermocouples" Proceedings of the IEEE 8th International Conference on Intelligent Systems (IS-2016) pp. 180-188 2016.*

[10]. *LT. Jolliffe Principal Component Analysis Series: Springer Series in Statistics NY:Springer vol. XXIX pp. 487 2002.*