IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 1, January 2023

DOI: 10.17148/IJARCCE.2023.12103

Anomaly Detection Using Machine Learning

Mr. Ali Karim Sayed [1], Dr. Ankush Pawar[2], Prof. Ankit Sanghavi[3]

ME Computer Engineering, Alamuri Ratnamala Institute of Engineering & Technology, Mumbai, India[1]

Associate Professor, AI&DS Dept., Konkan Gyanpeeth College of Engineering, Karjat, Mumbai, India[2]

Head of Dept. Computer Engineering, Alamuri Ratnamala Institute of Engineering & Technology, Mumbai, India[3]

Abstract: Comparative Machine Learning Analysis on Electrocardiogram (ECG) Anomaly Detection. Anomaly Detection on the ECG dataset of 4998 patients was done with each patient having 140 data points, around 7,00,00 data points. Data is divided as 4998 patients learning data. Machine Learning (ML) model is created using Algorithms like Logistic Regression, Decision Tree Classifier, Random Forest Classifier, and Support Vector Machines on which and remaining 1000 patient data is tested to get accuracy to check whether the model has learned correctly. The same is again analyzed by Deep Learning algorithms like RMSProp, Adam Optimization, and SGD.

Keywords: Anomaly detection, Electrocardiogram (ECG), Machine Learning (ML), Deep Learning (DL), Artificial Intelligence (AI), SVM (Support Vector Machines).

I. INTRODUCTION

This analysis shows the anomaly detection of ECG reports of the patients using Machine learning. As all Anomaly Detection is carried out over labeled data so Machine Learning Models were created using supervised Learning Algorithms and Deep Learning algorithms.

II. EASE OF USE

Label Data:

1: Good ECG Report

0: Bad ECG Report

III. DETAILS

The original dataset for "ECG5000" is a 20-hour-long ECG downloaded from Physionet. The name is BIDMC Congestive Heart Failure Database (chfdb) and it is recorded as "chf07". The data was pre-processed in two steps: (1) extract each heartbeat, (2) make each heartbeat equal in length using interpolation. After that, 5,000 heartbeats were randomly selected. The patient has severe congestive heart failure, and the class values were obtained by automated annotation. Best Algorithm: COTE, Best Accuracy: 94.61%. This result was obtained from previous research.

A. Lab Requirements

Online Google Research Lab, i.e., COLAB Python 3 Google Computer Engine Backend RAM: 12.68GB & Disk Space: 108GB.

B. What Is Anomaly Detection?

Anomaly detection is the process of identifying rare or unusual events or observations that do not conform to an expected pattern. Anomaly detection is identifying data points in data that don't fit the normal patterns. It can be useful to solve many problems including fraud detection, medical diagnosis, etc. Machine learning methods allow to automate of anomaly detection and make it more effective, especially when large datasets are involved.

C. What Is Machine Learning?

A machine learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 1, January 2023

DOI: 10.17148/IJARCCE.2023.12103

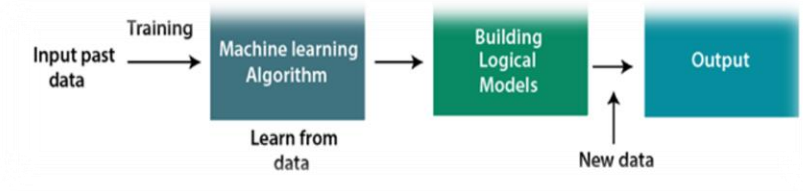


Fig. 1 Working of Machine Learning

D. Features of Machine Learning.

Machine Learning uses data to detect various patterns in each dataset. It can learn from past data and improve automatically. It's a data-driven technology. Machine learning is much like data mining. In supervised learning, the model is trained on a labeled dataset where the correct output is provided for each example. This can be useful for anomaly detection because the model can learn to recognize normal patterns and then flag anything that does not conform to those patterns as an anomaly. Deep learning is a type of machine learning that involves training artificial neural networks on large datasets. Deep learning models can learn complex patterns in data and can be effective for anomaly detection tasks.

E. Supervised Learning Algorithms Used

As all Anomaly Detection is carried out over labeled data so Machine Learning Models were created using the supervised learning algorithm. The list of Algorithms used in this paper is mentioned below.

1) *Decision Tree classifier:* A decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules ad each leaf node represents the outcome. Using this algorithm machine learning model, we could achieve an Accuracy of 98.3% with the below Confusion Matrix

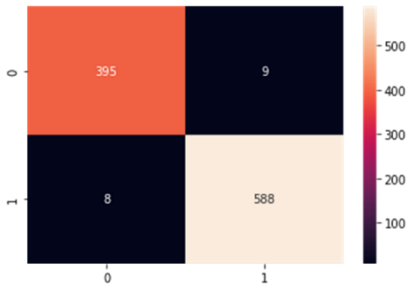


Fig. 2 Confusion Matrix generated for Decisions Tree Algorithm Applied

2) *Logistic regression:* Logistic Regression is another supervised algorithm that is used to solve classification problems. In classification problems, we have dependent variables n a binary or discrete format such as 0 or 1. Using this algorithm machine learning model we could achieve an Accuracy of 98.6% with the below Confusion Matrix

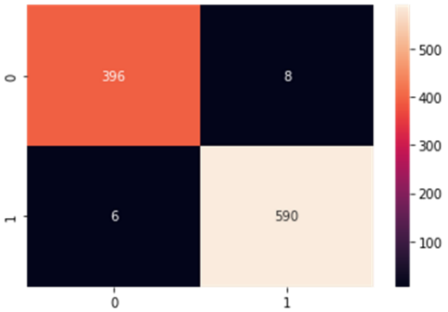


Fig. 2 Confusion Matrix generated for Logistic Regression Algorithm Applied

3) Random Forest Classifier: Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Using this algorithm machine learning model we could achieve an Accuracy of 99.3% with the below Confusion Matrix.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 1, January 2023

DOI: 10.17148/IJARCCE.2023.12103

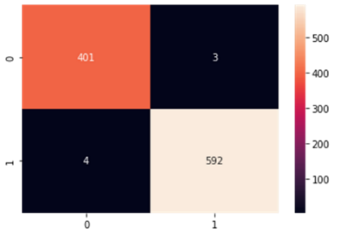


Fig. 2 Confusion Matrix generated for Random Forest Classifier Applied

4) *Support Vector Machines:* Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for classification as well as regression problems. However, primarily, it is used for Classification problems in Machine Learning. Using this algorithm machine learning model, we could achieve an Accuracy of 99.5% with the below Confusion Matrix

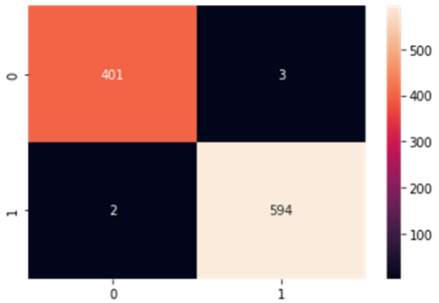


Fig. 3 Confusion Matrix generated for Support Vector Machin Algorithm Applied

F. Deep learning in AI:

Deep learning can be used for anomaly detection using autoencoders. Autoencoders are a type of neural network that is trained to reconstruct an input data point from a lower-dimensional representation, known as encoding. During training, the autoencoder learns to compress the input data into the encoding and then reconstruct the original data from the encoding.

Once the autoencoder is trained, it can be used to detect anomalies by comparing the reconstruction error of new data points to a threshold. The model can flag the data point as an anomaly if the error is above the threshold. This approach can be effective because the autoencoder has learned to reconstruct normal data points with low error, so any data points with a high reconstruction error are likely to be unusual or anomalous.

Any deep learning model tries to generalize the data using an algorithm and tries to make predictions on the unseen data. We need an algorithm that maps the examples of inputs to that of the outputs and an optimization algorithm. An optimization algorithm finds the value of the parameters (weights) that minimize the error when mapping inputs to outputs. These optimization algorithms or optimizers widely affect the accuracy of the deep learning model.

G. An Optimizer in DL:

While training the deep learning optimizer's model, we need to modify each epoch's weights and minimize the loss function. An optimizer is a function or an algorithm that modifies the attributes of the neural network, such as weights and learning rates. Thus, it helps in reducing the overall loss and improving accuracy. The problem of choosing the right weights for the model is a daunting task, as a deep learning model generally consists of millions of parameters. It raises the need to choose a suitable optimization algorithm for your application.

We have used RMS Prop (Root Mean Square) Deep Learning Optimizer: RMS prop is one of the popular optimizers among deep learning enthusiasts. RMS prop is ideally an extension of the work RPPROP. RPPROP resolves the problem of varying gradients.

H. Autoencoder in DL:

Autoencoders are a specific type of feedforward neural network where the input is the same as the output. They compress the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact "summary" or "compression" of the input, also called the latent-space representation.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 1, January 2023

DOI: 10.17148/IJARCCE.2023.12103

An autoencoder consists of 3 components: encoder, code, and decoder. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code.

I. Implementation:

Implementing an autoencoder for the following data set, with the hidden layer mentioned in the encoder and decoder below as shown in the table. We will now implement the autoencoder with Keras. The hyperparameters are: 140 nodes in the hidden layer, a code size is 32, and binary cross entropy is the loss function.

Table1
MODEL: "SEQUENTIAL"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 17)	2397
dense_1 (Dense)	(None, 16)	288
dense_2 (Dense)	(None, 8)	136
dense_3 (Dense)	(None, 4)	36
dense_4 (Dense)	(None, 2)	10

```
tf.keras.layers.Dense(17, activation="relu"),
tf.keras.layers.Dense(16, activation="relu"),
tf.keras.layers.Dense(8, activation="relu"),
tf.keras.layers.Dense(4, activation="relu"),
tf.keras.layers.Dense(2, activation="relu")
```

The output of the Dense method is a callable layer, using the functional API we provide it with the input and store the output. The output of a layer becomes the input of the next layer. All the layers use the relu activation function, as it's the standard with deep neural networks. The last layer uses sigmoid activation.

```
reconstructions = autoencoder.predict(normal_train_data)
train_loss = tf.keras.losses.mae(reconstructions, normal_train_data)
plt.hist(train_loss[None,:], bins=50)
plt.xlabel("Train loss")
plt.ylabel("No of examples")
plt.show() # Normal train data Histogram
```

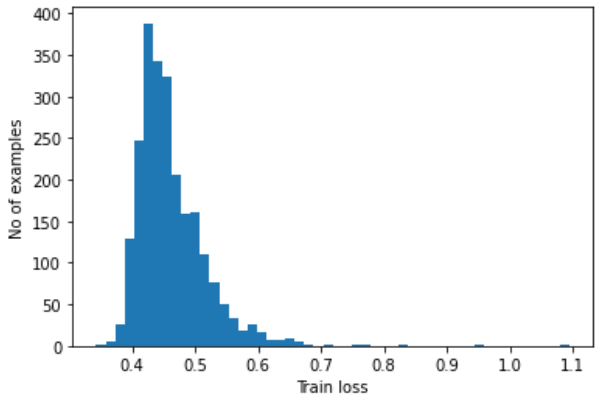


Fig. 4 Normal train data Histogram

```
threshold = np.mean(train_loss) + np.std(train_loss)
print("Threshold: ", threshold)
Threshold:  0.5160779402829193
```

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 1, January 2023

DOI: 10.17148/IJARCCE.2023.12103

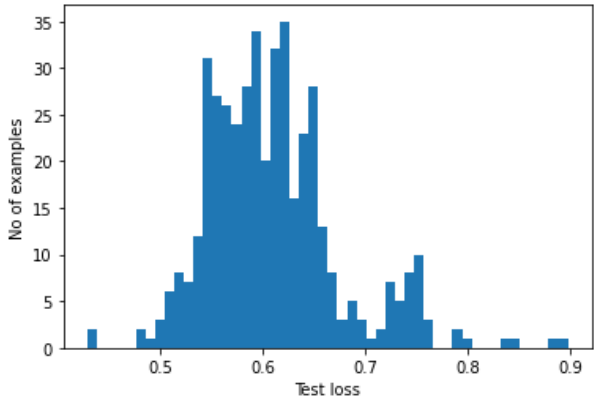


Fig. 5 Anamolious train data Histogram

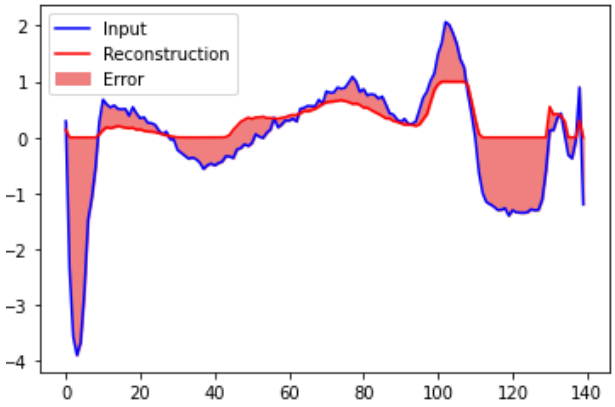


Fig. 6 Reconstruction

Above Fig.6. is a reconstruction for the Good ECG report

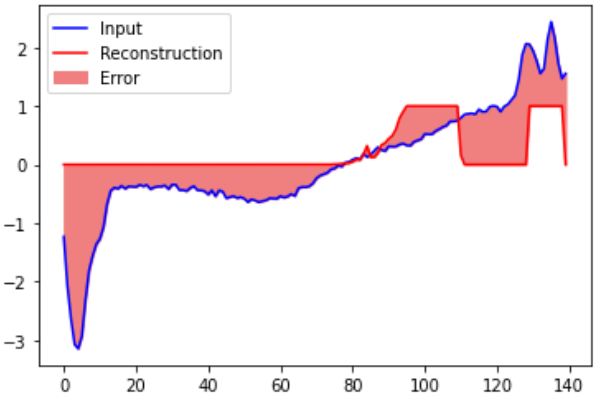


Fig. 7 Reconstruction

Above Fig.7. is a reconstruction for the BAD ECG report

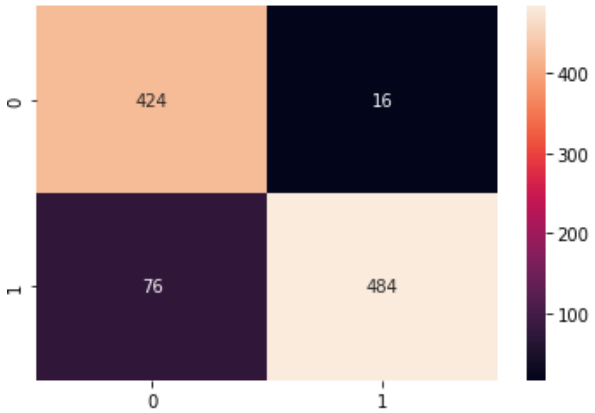
Here is the below confusion matrix and accuracy of 90.8% after applying Deep learning autoencoder technique.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.918 ⁜ Vol. 12, Issue 1, January 2023

DOI: 10.17148/IJARCCE.2023.12103



Accuracy: 90.8% and Confusion Matrix

IV. CONCLUSIONS

After applying various supervised learning and deep learning algorithm to obtain optimum accuracy we found that a Support Vector Machine (SVM) can achieve an accuracy of 99.5%. Better than what was achieved in the base paper with an accuracy of 94.61% using the COTE Algorithm in the said data set.

ACKNOWLEDGMENT

I would like to thank **Prof. Ankush Pawar** for his guidance and support during this research. I am also grateful to **Prof. Vivek Pandey** for his guidance and support throughout the project. We would like to acknowledge timeseriesclassification.com for its valuable contributions to data collection and analysis. Once again, we would like to thank timeseriesclassification.com for providing the Dataset that was essential to this research and comparative analysis

REFERENCES

[1] http://www.timeseriesclassification.com/description.php?Dataset=ECG5000
[2] Data Set link http://www.timeseriesclassification.com/Downloads/ECG5000.zip
[3] https://dl.acm.org/doi/10.1007/s10618-014-0388-4 A general framework for never-ending learning from time series streams.
[4] Autoencoder Tutorial: https://www.kaggle.com/shahules/autoencoder-tutorial
[5] A Beginner's Guide to Autoencoders: https://towardsdatascience.com/a-beginners-guide-to-autoencoders-e77cd3f8d7b2
[6] Support Vector Machines: A Simple Explanation: https://towardsdatascience.com/support-vector-machines-a-simple-explanation-d67b04ec7eb8
[7] How to Create a Heat Map in Python: https://towardsdatascience.com/how-to-create-a-heatmap-in-python-8d665aa43c0c.