

# **End semester report on R & D Project (NU 302)**

**Academic Year- 2020-21**

on

## **Anomaly Detection in High Dimensional Data Using Cluster Based Isolation Forest (CBIF)**

A dissertation

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

by

- |                      |              |            |
|----------------------|--------------|------------|
| 1. Astha Kumar       | (BT18GCS016) | B.Tech CSE |
| 2. Kawal Nain Singh  | (BT18GEC165) | B.Tech CSE |
| 3. Riona Chakrabarti | (BT18GCS163) | B.Tech CSE |

Under supervision of

**Suman Sanyal**



NIIT University, Neemrana, Rajasthan-301705

May 2021



## **DECLARATION BY STUDENT(S)**

I/We hereby declare that the project report entitled **Anomaly Detection in High Dimensional Data using CBIF** which is being submitted for the partial fulfilment of the Degree of Bachelor of Technology, at NIIT University, Neemrana, is an authentic record of my/our original work under the guidance of Suman Sanyal. Due acknowledgements have been given in the project report to all other related work used. This has previously not formed the basis for the award of any degree, diploma, associate/fellowship or any other similar title or recognition in NIIT University or elsewhere.

Place: NIIT University, Neemrana, Rajasthan-301705

Date: 17 May 2021

1. **Astha Kumar** (BT18GCS016) **B.Tech CSE**

2. **Kawal Nain Singh** (BT18GEC165) **B.Tech CSE**

3. **Riona Chakrabarti** (BT18GCS163) **B.Tech CSE**



### **CERTIFICATE BY SUPERVISOR(S)**

This is to certify that the present R&D project entitled **Anomaly Detection in High Dimensional Data using CBIF** being submitted to NIIT University, Neemrana, in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology, in the area of CSE, embodies faithful record of original research carried out by **Astha Kumar, Kawal Nain Singh and Riona Chakrabarti**. They have worked under my guidance and supervision and that this work has not been submitted, in part or full, for any other degree or diploma of NIIT or any other University.

Place: NIIT University, Neemrana, Rajasthan-301705

Name of the Supervisor(s) with signature

Date: 16 May 2021

## **Acknowledgment**

First and foremost, we would like to say that it has been a pleasure working on this research project titled, 'Anomaly Detection in High Dimensional Data using Cluster Based Isolation Forest.' This would not have been possible without the kind support and help of many individuals and we would like to extend our sincere thanks to all of them individually.

We would like to extend our sincere gratitude to our Dean Research, Dr. Ratna Sanyal, Professor, Computer Science and Engineering, Ph.D, NIIT University, Neemrana, for giving us the opportunity to work on this project.

We would also like to express our deep appreciation for our Research Supervisor, Dr. Suman Sanyal, B.Sc., M.Sc., Associate Professor, Data Science, NIIT University, Neemrana. His vision, sincerity and guidance throughout our research has deeply inspired us. Our panelists' suggestions continuously challenged us to work harder. We would like to express our gratitude to our parents for their kind cooperation and encouragement which helped us in completion of this project.

Lastly, our appreciation also goes to our college, NIIT University, Neemrana, for helping us obtain the resources that were prudent for the research and development of this project.

## **TABLE OF CONTENTS**

1.	Introduction.....	6
2.	Problem Statement.....	10
3.	Literature Review.....	11
4.	Proposed Methodology.....	12
4.1.	Workflow.....	12
4.2.	Technology.....	14
5.	Result and Analysis.....	17
5.1.	Isolation Forest.....	17
5.2.	Cluster Based Isolation Forest.....	18
5.3.	Comparison.....	21
6.	Conclusion and Future Scope.....	22
7.	References.....	23
8.	Annexure.....	23

## 1. Introduction

Outlier Detection, often referred to as anomaly detection is the process of identifying and detecting points of observations from a given dataset that deviate from the rest of the data and do not conform to an expected pattern. Outlier detection is one of the primary steps of data mining and is widely used in the field of network intrusion detection, medical diagnosis, industrial system fault, flood prediction and intelligent transportation system.

Anomalies can be categorized as Global and Local. Global anomalies refer to those points in a dataset that deviate significantly (value may be higher or lower) as compared to the rest of the dataset. For example, if we are given a dataset of 100 points, where 99 points lie between 0-100 and the 100th point is 10,000, this 100th point may be construed as a global outlier. Local anomalies, on the other hand are those points, which when compared to the entire dataset, do not deviate from it. However, when compared to its neighboring points, it has a significant deviation. For example, if we were to store the heights of all 20 year olds in a University, there may be chances of a few students being extremely taller or shorter than other students. However, if you compare it with the heights of all 20 year olds in a country, the frequency of people being taller or shorter will increase, therefore, not making them an outlier.

Further, based on the number of features, anomalies can also be classified as Univariate and Multivariate. The former kind of anomalies, univariate, are found in the distribution of values within a single feature space as the name suggests. The latter, multivariate, are found within an n-dimensional (multi-dimensional or higher dimensional) space. It is often very computationally heavy to analyse high dimensional data, hence, a model that is fitted with a labeled training set that can analyse the anomalies in the test set *efficiently* is required.

Efficiency of an algorithm refers to the computational resources employed by the algorithm. Before actually running the algorithm, one needs to analyze how much resources are going to be used while running. For maximum efficiency, the algorithm needs to be optimized so as to use the minimum amount of resources. There are two types of efficiency that we need to consider- Space and Time and depending on the desired outcome, one of them needs to be optimized for.

The effectiveness of an anomaly detection algorithm will depend on the accuracy and precision with which it is able to classify data as anomalous and non-anomalous points.

Accuracy refers to the ratio of correctly predicted observations (in this case, anomalies) to the total number of observations (the entire dataset) and is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Here, TP = True Positives (correct predictions of anomalous data)

TN = True Negatives (correct predictions of non-anomalous data)

FP = False Positives (wrong predictions of anomalous data)

FN = False Negatives (wrong predictions of non-anomalous data)

Precision refers to the ratio of correctly predicted positive observations (correct predictions of anomalous data) to the total number of predicted positives (total anomalous data points) and is calculated using the following formula:

$$Precision = \frac{TP}{TP + FP}$$

Here, TP = True Positives

FP = False Positives

There are a multitude of ways in which anomalies can be present depending on the environment such as: point anomalies, contextual anomalies, and collective anomalies. Point anomalies occur as single points of observations that lay separate from the rest of the data or distribution. Contextual anomalies can be called noise in the data that deviates from the said data in the same context. Collective anomalies can be said to be a subset of the anomalies that together deviate from the rest of the observations. The individual point in that subset may not be categorized as an outlier, however, the entire subset may.

Various methods exist for the sole purpose of detecting such anomalies from a dataset which are broadly classified as- distribution-based methods, distance-based methods, density-based methods, and clustering methods. Additionally, the distribution-based method obtains the distribution model of data that needs to be tested in advance, which depends on the global distribution of the dataset, however, it is not applicable to the dataset with uneven distribution. The distance-based approach requires the users to select a reasonable distance, scale parameters, however, is less efficient on high dimensional datasets. The density-based local outlier identification method ascribes the level of outlierness of the observation points outlined by local density. In the clustering method, cleaning of the dataset and clustering based on similarity is done. With this, anomalies are removed on the key attribute subset instead of on the complete dimensional attributes of the dataset.

The above outlier detection methods, except the density-based approach, adopt global anomaly standards to process data objects, which cannot be performed on the datasets with skewed distribution. In almost all practical applications, the distribution of data tends to be imbalanced, and there is a lack of indicators that can classify data. Data imbalance occurs because of skewed class proportions. Even if tagged datasets are available, their applicability to outlier detection tasks is often unknown.

Anomaly detection algorithms can be categorized as Supervised, Semi-supervised, and Unsupervised. The supervised learning algorithms train the data by using a dataset which is pre-labelled as normal and anomalous data points. Semi-supervised learning method is the type of machine learning that deploys a combination of labelled data in small amounts and unlabelled data in a substantial amount to train models. Lastly, the unsupervised learning algorithm of outlier detection does not need a training dataset and thus, does not employ a technique of manually labelling the data. This is based on the assumption that only a small portion of the data will be anomalous, whereas a large portion of the data will be normal.

One such type of unsupervised learning method is the Isolation Forest or iForest which does not detect anomalies in the conventional sense. They do not point to the observations that deviate from what a normal data should look like, instead, as the name suggests, it identifies anomalies by isolating anomalies in the data. Isolation Forest implements the principle of decision trees. It



segregates the anomalies by randomly determining a parameter from a given set of parameters and assigning a split value between the minimum and maximum values among the parameters. This partitioning produces multiple smaller paths in the main tree for the erroneous data values to differentiate between that and the normal data.

The other principle that iForest works on is recursion. The algorithm recursively gives rise to partitions in the dataset by identifying the split value.

In comparison to the data observations that are not anomalies, the points that are anomalies need lesser partitions to be isolated. Therefore, the anomaly points are going to be the shorter paths within the trees. Here, one can assume that the length of the trail is equal to the edge numbers traversed from the root node.

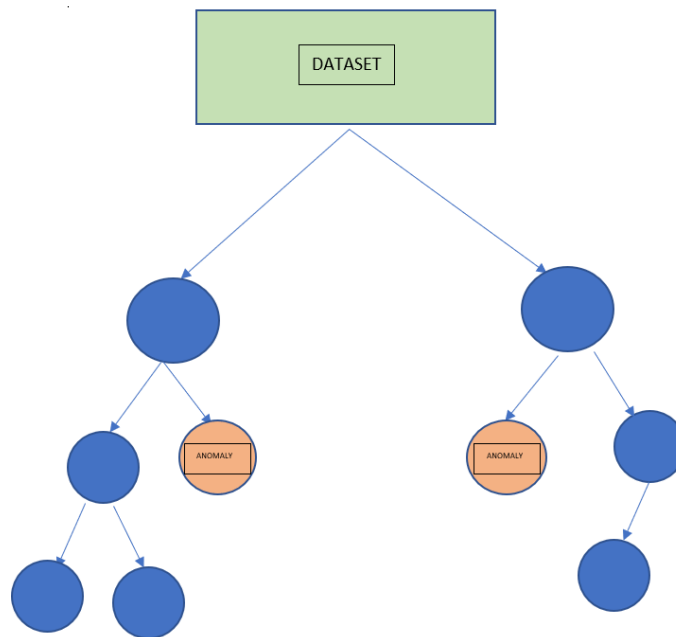


Figure 1.

As we can see in Figure (1), the orange leaf nodes have a shorter path length than the other blue leaf nodes. This could indicate that the orange nodes are anomalous and blue nodes are non-anomalous.

However, like any method or algorithm, iForest too, poses certain problems. The main drawback of iForest is not with the model itself, but in the way it computes the anomaly score, wherein

there is a possibility that the reliability may reduce due to the tree branching introducing a bias in the ranks of anomaly score. A solution to this is proposed in this research article which will talk about a possible combination of Clustering and iForest method.

The remainder of the report is organized as follows: Section 2 introduces the problem statement and objective of this report. Section 3 cites the related work on outlier detection. Section 4 details the outlier detection algorithm. Section 4 discusses the datasets, metrics for performance evaluation and the experimental results compared with other methods and Section 5 concludes the report.

## **2. Problem Statement & Objective**

An anomaly is an abnormality in a data set that can show inconsistencies in the data and contribute to not getting the desired results from the data. Therefore, detecting these anomalies or anomalies becomes extremely crucial. Anomaly Detection is a technique that finds these unusual patterns and points present in any given data set.

The objective of this Research and Development project is to contribute to the area of research of Anomaly Detection in High Dimensional Data by analysing and showing simulations in the following manner-

1. A two-step progressive distance-ensemble based model to overcome the challenges posed by clustering and iForest individually.
2. Two types of datasets are being taken- synthetic and real time data. The synthetic dataset gives an output of random instances, whereas the real time data gives an output of customer trading reports. It is difficult to find datasets with little to no inconsistencies, hence, comparing synthetic and real-time datasets will enable viewers to analyse the accuracy difference and understand real-life applications.
3. In this, clustering, based on distance, is the first step and is implemented to group the data into various clusters. Here, our aim is to use k-means clustering which is one of the most efficient algorithms to implement.

4. Next, iForest is applied on each individual cluster, iterated over them and based on the decision trees formed, anomalies are extracted.

A detailed analysis of both the methodologies will be further outlined in the report.

### **3. Literature Review**

Methodologies for anomaly detection are influenced by the nature, spread and distribution of the data, and the way we define the anomalies leading to the formulation of a wide variety of use cases. Even though there are a number of computational techniques that exist, each of these must be tailored to overcome their flaws while effectively detecting anomalies in a particular use case. The increase in the number of IoT devices, with the aid of advanced data collection and better computing resources has led to increased prioritisation of real-time, large-scale data which, in turn, require better techniques for the process of outlier detection with improved speed, complexity and accuracy.

In this report, the focus is on removing anomalous points from the data set to improve its overall quality and the process of data analysis on the data set. However, there are a number of methodologies that prioritise the anomalies themselves as the main source of insight regarding some usual occurrence in the real world. In such situations, understanding the root cause of those anomalies can cause detection and prevention of events that may cause significant harm to life or properties.

The techniques used in anomaly detection can be broadly classified as supervised, semi-supervised and unsupervised. The techniques used can be density-based, ensemble-based or even utilise models such as support vector machines, kernel methods, or neural networks, specifically models such as autoencoders, long-short term memory, or self-organizing maps [4].

Algorithms which follow an ensemble-based approach include Feature bagging and Isolation Forest. The Feature Bagging algorithm combines the outlier predictions from multiple machine learning algorithms thus generating more accurate predictions than any individual model. Isolation Forest, on the other hand, generates an ensemble of trees and identifies an outlier

depending on its path length on the trees. The iForest method suffers a significant drawback - its inability to detect local anomalies in case of several clusters of normal instances in the dataset. Originally proposed by Liu FT [5], the iForest algorithm has undergone several modifications to make it more efficient for certain uses. Its low time complexity and high accuracy of the technique makes it popular. Guillaume Staerman utilised Isolated Forest to detect anomalies in functional data. Liefu Liao and Bin Luo created the E-iForest algorithm which utilised the dimension entropy for selection of isolation attributes and isolation points in the training process.

Combining cluster-based analysis to outlier detection algorithms can improve the detection rate and reduce the number of false positives. Techniques like the DBSCAN algorithm, DBSCAN,  $k$ -Means or fuzzy set techniques are some examples. The  $k$ -Means clustering algorithm was applied in a combination with the iForest algorithm to create the Cluster Based Isolation Forest algorithm which was then applied to financial data in order to detect fraud [6].

## **4. Proposed Methodology to Specific Gap/Challenge**

### 4.1. Workflow:

#### 4.1.1. Cluster Based Isolation Forest

As one of the important tasks of data mining, outlier detection is widely used in fields of network intrusion detection, medical diagnosis, industrial system fault etc. The proposed method is a two-step distance-ensemble based model called Cluster Based Isolation Forest that involves the implementation of two steps-

- K means clustering, and
- Isolation Forest.

Before we can implement it in python, we need to understand the mathematical logic behind each module.

#### a. K-means Clustering:

This is a type of unsupervised learning method that divides the entire dataset into smaller clusters or groups such that the intra-cluster similarity is maximum and inter-cluster similarity is minimum. Here, 'k' is referred to as the number of clusters that need to be formed. Once a suitable value of 'k' is determined, centroids are decided and each data

point in the dataset is assigned to each centroid. The centroids are then iteratively updated based on the points, until no point changes its cluster.

While there is no defined set of rules to follow when it comes to deciding the value, the most popular method is to run the algorithm with a range of values and then calculate the sum of squared error (SSE) for each data point with the centroid value of the cluster. After this, plot a line graph and the point that first reaches closest to the x-axis is the optimal value for k. This is also called the elbow method as the line graph resembles an elbow, shown in Figure 2.

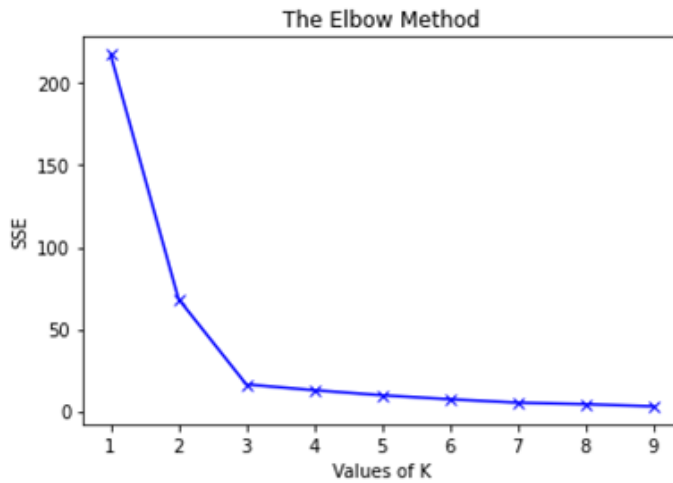


Figure 2.

To calculate the SSE, following calculation needs to be carried out for each data point in the cluster:

$$SSE = \sum_{i=1}^n (x_i - \bar{x})^2$$

Where, n = number of observations;

$x_i$  = value of the  $i^{th}$  observation;

$\bar{x}$  = mean of all observations.

b. Isolation Forest:

This algorithm works on the principle of isolating the anomalies by forming decision trees over specific attributes. The shorter path lengths constitute the anomalies since, few instances would result in a deviation or anomaly. Also, since anomalies are easier to distinguish as compared to normal data, it would result in shorter path lengths.

This algorithm can be used either on datasets containing one variable or datasets containing two or more variables. It is used as a function and has one parameter known as *rate*, which controls the target rate of anomaly detection. A rate equal to 0.1 will train the algorithm to detect anomaly in 1 out of 10 data points on average. The rate must lie between 0 and 0.5. As we can see from the name of this algorithm, this is an isolation-based anomalies detection algorithm. In this algorithm, a binary tree is used to classify the data set. If a data item in the beginning is identified as the leaf node, then chances are that it will be identified as the anomaly point.

The CBIF algorithm consists of three stages: clustering, building iTrees, and evaluating. The clustering stage involves using a dataset for clustering, where the dataset is split into k different clusters, using k-means clustering, until all clusters contain all samples. The second stage which is building of iTrees consists of creating the iTree for subsample of each cluster. Lastly, the evaluation stage obtains the path length of every data point within the iTree and calculates the anomaly score of every data point based on the above-mentioned path length.

#### 4.2. Technology:

The algorithm first randomly samples multiple sample data sets from the overall data set and builds multiple isolation trees (iTrees) according to classification of attribute values, then calculates the average path length of the data items and then finally normalizes them. The closer the result is to 1, the more likely the data item is an anomaly. The given formula is used to calculate the anomaly score of datapoint 'x.' The notations for the same are given in Table 1.

$$S(x, \psi) = 2 - E(h(x))/c(\psi)$$

X	a data point	X	a data set of n instances
k	number of clusters	n	Quantity of observation points in a dataset is $n= x $
Q	a set of attributes	H	returns the path length of x
q	an attribute	$\psi$	subsampling size of every cluster
T	a tree or a node	S	an anomaly score
t	number of trees	L	class of data points

Table 1.

The iForest algorithm identifies the anomalies by categorizing the cost of a particular observation point, and the time complexity as  $O(n)$  according to the algorithm. At the same time, a major drawback is that this algorithm is sensitive to anomalies with short average path length, causing problems that cannot identify local anomalies with deep path length in iTrees. The IF algorithm performs well in terms of time complexity and quantitative description of the anomaly degree of data, but one of the major disadvantages is that it is unable to accurately identify the local anomalies, which leads to one-sided results. Therefore, the IF algorithm can be improved to identify the local anomaly points while retaining its advantages.

To solve this shortcoming of the IF algorithm that is difficult to identify local anomaly points, an improved CBIF (Cluster-Based Isolation Forest) algorithm is proposed. The main idea of the CBIF algorithm is, first it uses the clustering algorithm to divide all samples into different clusters, once this is done, it converts the local anomalies before clustering into global anomalies of adjacent clusters, and then finally it calculates the anomaly score in each data point of clusters. K-means clustering and iForest are the two

algorithms selected mainly due to their linear time complexity for refining the recognition capabilities of local anomalies and accuracy. The symbols and notations involved in the improved algorithm are shown in table 1.

There are three steps in the Cluster-Based Isolation Forest which are- clustering, building iTrees and evaluation. The clustering stage uses a given sample set for clustering, and the sample set is divided into  $k$  different clusters until all clusters contain all samples. The second stage, building of iTrees, builds the iTrees for the subsamples of every cluster traversed. The last stage, evaluation, acquires the length of the path of each observation point in the iTree in accordance with the iTree and enumerates the outlier score of each data point based on path length of iTree.

Two steps are to be followed to build iTrees, first we must create a single iTree, which is built repeatedly by subset  $C_i'$  of  $k$  clusters from cluster stage, among  $C_i'$  is randomly selected which is from  $C_i$ ,  $C_i' \subset C_i$  ( $1 \leq i \leq k$ ). Secondly, merging into iForest which includes all iTrees from the first step. The second stage is ended until every data point of  $k$  clusters is isolated.

Some of the libraries used in our project are 'matplotlib', 'numpy', 'pandas', 'Kmeans', 'IsolationForest'. Matplotlib is a cross-platform data visualization and graphical plotting library. Matplotlib is the numerical extension of numpy. Numpy is an open source library in python which contains multi-dimensional array and matrix data structures, and it can be used to perform a number of mathematical operations on these data structures. Pandas library is very useful for data analysis. Pandas provide powerful easy to use data structures, as well as the means to quickly perform operations on these structures.

Isolation Forest Algorithm construction:

**Algorithm 1** iForest ( $X, t, s$ )

**Input:**  $X$  - input dataset,  $t$ -number of trees,  $s$ -subsampling size

**Output:** a set of  $y$  iTrees

```
Initialize Forest
set height limit  $l = \text{ceiling } \log_2 s$ 
for  $i = 1$  to  $t$  do
     $X' \leftarrow \text{sample } X, s$ 
    Forest  $\leftarrow \text{Forest} \cup \text{iTree } X', 0, l$ 
end for
```



```
return Forest
```

## Algorithm 2 Cluster Based Isolation Forest

Range in the for loop is given from 0 to 3 because the number of clusters generated are 3.

From line number 4 the iForest algorithm comes into use. We fit the iForest algorithm on a column of our data frame.

Then we pass the PROFIT\_YTD column of our data frame in the function.

```
for i in range(0,3):
    mask = df_churn['Clusters'] == i
    df_new = pd.DataFrame(df_churn[mask])
    iforest = IsolationForest(n_estimators=100,
contamination=float(.2))
    iforest.fit(df_new[['PROFIT_YTD']])
    predicted=iforest.predict(df_new[['PROFIT_YTD']])
    df_new['anomaly']= pd.Series(predicted).apply(lambda x: 'yes'
if x==1 else 'no')
    show_scatter_3d(df_new, x_name, y_name, z_name,
predicted=predicted, centers=centers);
```

## 5. Result and Analysis

### 5.1. Isolation Forest

The Isolation Forest algorithm is highly efficient in terms of time complexity but suffers a deficiency in the identification of local anomalies. To prove this, we have used a dataset generated by sklearn's make\_blobs function.

On running just the iForest algorithm on a synthetic database with a contamination factor of 0.01, we get anomalous points in the first and third cluster but not in the second cluster, as shown in Figure. 3. This shows that the algorithm is accurately classifying the global anomalies but

cannot detect anomalous points within the clusters. The green dots represent the anomalies in the dataset.

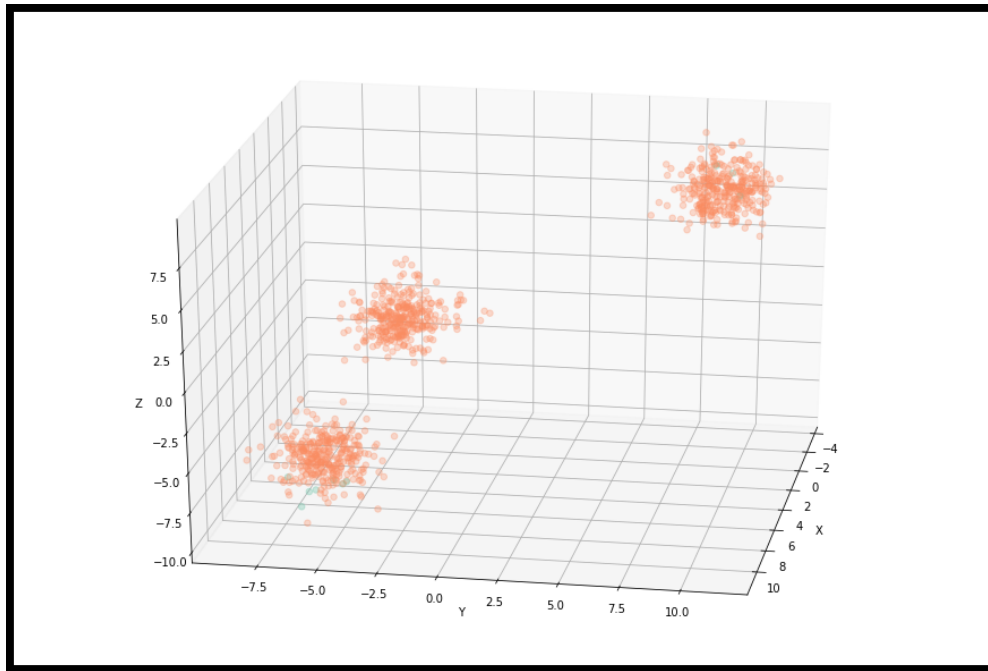


Figure 3.

## 5.2. Cluster Based Isolation Forest

The python code implemented had two major components- k means clustering and Isolation Forest. This dual component code was implemented to detect the local anomalies which was made possible by using sklearn's make.blob functionality to create a dataset. The synthetic dataset generates random instances and helps in providing a better visual representation of how clustering is working.

We generated a 10 dimensional dataset with random numeric values in the columns.

	X	Y	Z	A	B	C	D	E	F	G
0	5.736657	-5.495132	-6.225783	-7.679217	-5.675882	10.204134	6.499985	-6.153485	6.012104	-5.607725
1	6.856021	-3.128796	-5.362158	-8.365592	-5.891344	8.623532	8.743615	-7.107779	5.734039	-4.473739
2	-2.632855	8.364998	5.760151	-4.492198	-5.883384	-2.821694	-10.137816	-5.904026	5.706769	-10.675624
3	-0.929539	-5.094169	1.039638	5.257847	-9.899305	-7.026905	3.282882	6.138319	-5.325245	0.459992
4	7.974444	-6.290471	-7.347292	-7.390165	-7.305787	8.869790	6.115273	-7.500320	5.510690	-4.122081

a. Clustering: The first step for clustering is to decide what the value of 'k' should be.

i. In the first part of this code, we are implementing clustering on the synthetic dataset as is illustrated by Figure. 4.

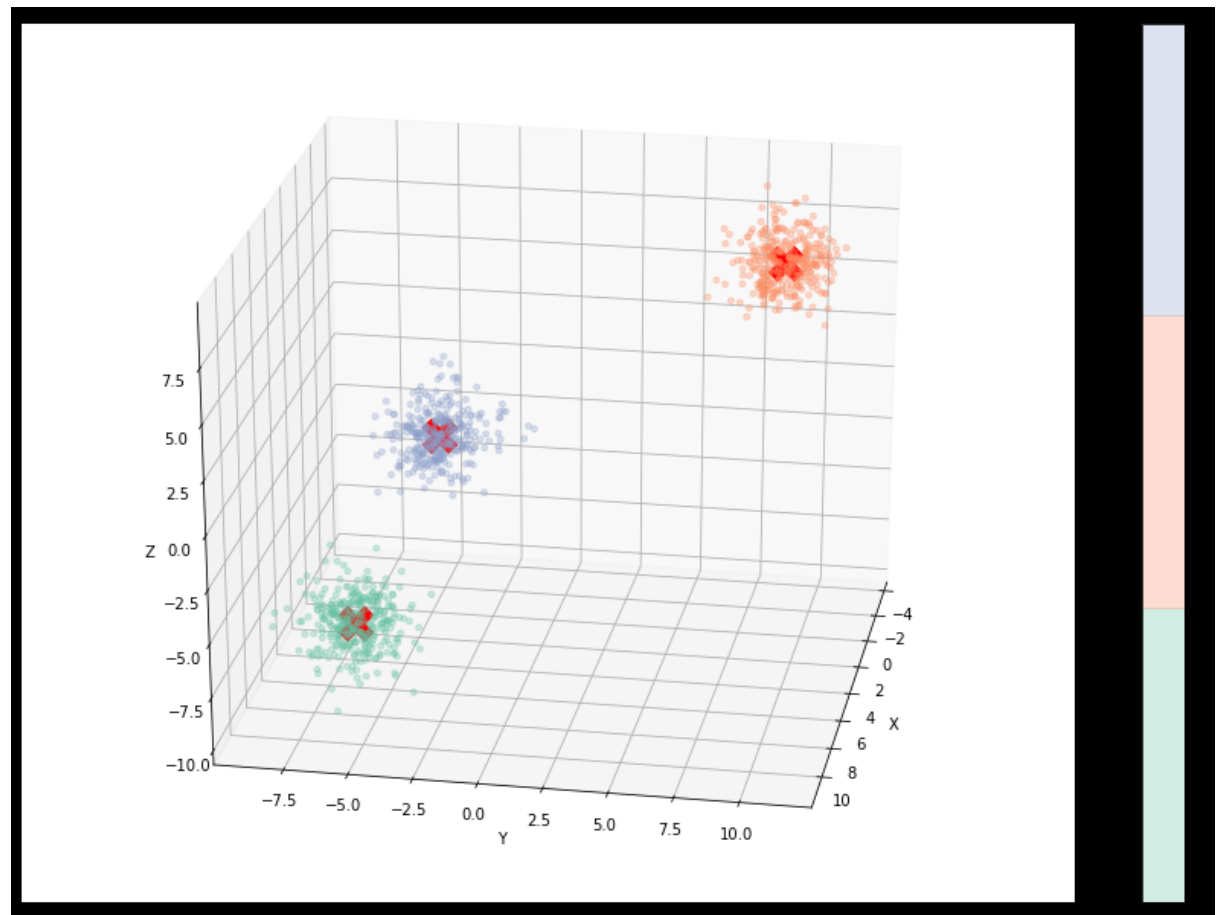


Figure 4.

Here, as we are implementing k-means clustering, the value of  $k$  will determine the number of clusters formed. Therefore, with  $k=3$ , three distinct clusters were generated.

ii. Second part of the code consisted of generating a visualization for real data as illustrated in Figure. 5.

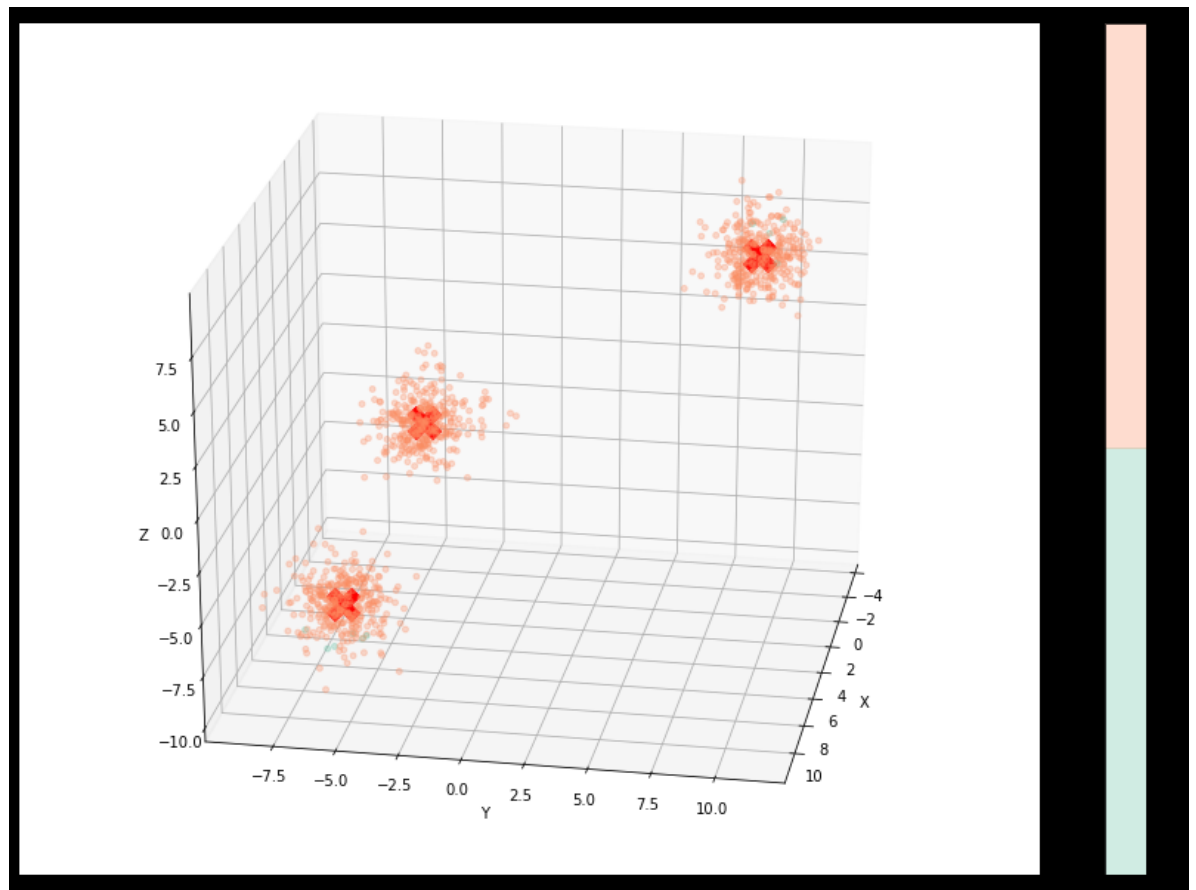


Figure 5.

Similar to Figure 1, value for  $k$  was kept 3 and three clusters were formed. However, in comparison to Figure 1, Figure 2 has an overlap between the red and blue cluster. This overlap consists due to certain irregularities in the real dataset.

In both the Figures, the ‘x’ mark in each cluster depicts the centroid of that cluster that helps us find out the intra and inter cluster similarity.

### 5.3. Comparison between iForest and CBIF

Comparison between iForest and CBIF was visualized by running the algorithm over a dataset that contains customer’s trading data to obtain the analysis and a three dimensional visualization. From the original dataset, we extracted the columns with numerical data to run this algorithm:

	AGE	TOTALUNITSTRADED	DAYSSINCELASTTRADE	DAYSSINCELASTLOGIN	PROFIT_YTD
0	47	58	13	2	-152.76525
1	25	13	10	4	1349.63500
2	42	28	5	4	1123.61250
3	52	36	6	3	-652.56550
4	40	8	9	4	-1496.14950

The green dots in Figures 6(i) and 6(ii) in table 2 show the anomalous points, whereas the orange dots show the non-anomalous points.

Firstly, as stated before in Section 1, the iForest algorithm can only detect global anomalies in comparison to CBIF which can also detect the local anomalies. As is visible in Figure. 6(i), the anomalous points seem to be concentrated on the outer part of the dataset points, indicating the presence of global anomalies. However, it is unable to recognize any anomalies that might be present inside the individual clusters.

Now, applying CBIF on the same dataset, one can visualise by looking at Figure. 6(ii) that clustering before iForest results in the detection of local anomalies. The ‘x’ mark on each cluster represents the centroid for that cluster.

There is a clear contrast between the outputs of the two algorithms. CBIF gives more accurate findings as compared to iForest by dividing the dataset into smaller clusters, converting the global anomalies into local anomalies and highlighting them in green.

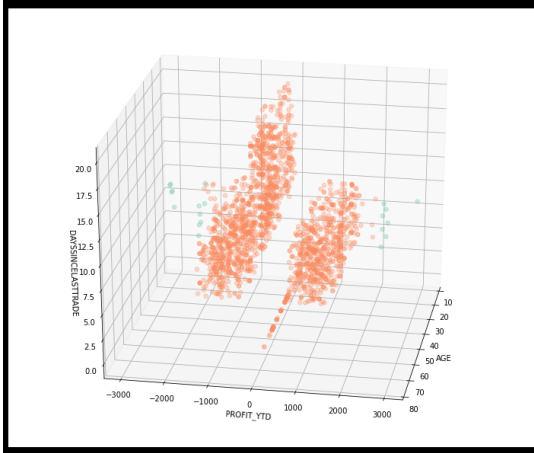
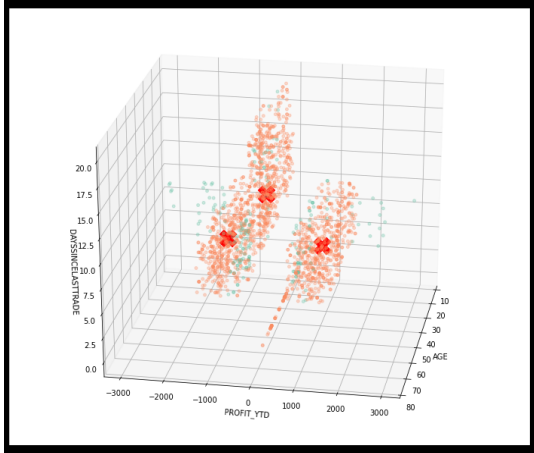
 <p>Figure 6(i)</p>	 <p>Figure 6(ii)</p>
---	---

Table 2

## **6. Conclusion and Future Scope**

Research on the topic of Anomaly Detection has been going on for decades now and various advancements have been made. Future work for this research paper includes hardcoding the code without the use of any inbuilt libraries.

A shortcoming of the code used is that it projects the data in 3-Dimensional space which does not give accurate understanding of its results for higher dimensional data. Therefore, computing the effectiveness of this algorithm, as mentioned above, using some other parameter that will give accurate results for a high dimensional space as well, becomes an important step in developing this algorithm further.

To conclude, this research paper was primarily concerned with the types of anomalies and importance of detecting them in higher dimensions. The paper moves on to talk about the different methods that have been used to detect certain anomalies. We introduced a two-step distance-ensemble based model that can predict local anomalies by first clustering the dataset into smaller groups and then running an Isolation Forest algorithm over each cluster to retrieve decision trees with the path length of anomalies shorter as compared to the normal data.

## **7. References**

- [1]P. Verma, "Isolation Forest Algorithm for Anomaly Detection", *Medium*, 2020. [Online]. Available:  
<https://heartbeat.fritz.ai/isolation-forest-algorithm-for-anomaly-detection-2a4abd347a5>.  
[Accessed: 16- May- 2021]
- [2]"Stock Market Volume Anomaly Detection 1", *Slicematrix.github.io*. [Online]. Available:  
[https://slicematrix.github.io/stock\\_market\\_anomalies.html](https://slicematrix.github.io/stock_market_anomalies.html). [Accessed: 16- May- 2021]
- [3]Z. Cheng, C. Zou and J. Dong, "Outlier detection using isolation forest and local outlier factor", *Proceedings of the Conference on Research in Adaptive and Convergent Systems*, 2019.
- [4]C. Zhou and R. Paffenroth, "Anomaly Detection with Robust Deep Autoencoders", *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.
- [5]F. Liu, K. Ting and Z. Zhou, "Isolation Forest", *2008 Eighth IEEE International Conference on Data Mining*, 2008.

[6]J. Liu, J. Tian, Z. Cai, Y. Zhou, R. Luo and R. Wang, "A hybrid semi-supervised approach for financial fraud detection", *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2017.

[7]F. Siddiqi, G. Gorman, L. A. Barba and S. Corlay, "jupytercon/2020-ClusteringAlgorithms", GitHub, 2021. [Online]. Available: <https://github.com/jupytercon/2020-ClusteringAlgorithms>. [Accessed: 16- May- 2021]