Master Degree Project

UNIVERSITY OF SKÖVDE

EVALUATION OF CLUSTER BASED ANOMALY DETECTION

Master Degree Project in Informatics with a specialisation in Data Science
One year Level 15 ECTS
Autumn term 2019

Ajay Sreenivasulu

Supervisor: Alexander Karlsson
Examiner: Nikolaos Kourentzes

ABSTRACT

Anomaly detection has been widely researched and used in various application domains such as network intrusion, military, and finance, etc. Anomalies can be defined as an unusual behavior that differs from the expected normal behavior. This thesis focuses on evaluating the performance of different clustering algorithms namely k-Means, DBSCAN, and OPTICS as an anomaly detector. The data is generated using the MixSim package available in R. The algorithms were tested on different cluster overlap and dimensions. Evaluation metrics such as Recall, precision, and F1 Score were used to analyze the performance of clustering algorithms. The results show that DBSCAN performed better than other algorithms when provided low dimensional data with different cluster overlap settings but it did not perform well when provided high dimensional data with different cluster overlap. For high dimensional data k-means performed better compared to DBSCAN and OPTICS with different cluster overlaps.

Table of Contents

1. INTRODUCTION

In today's world, there is a huge amount of data being generated every day in different sectors and stored in the databases; therefore, there is a rising demand for analyzing more efficient and effective methods to make use of information present in the data (Ankerst et al. 1999). There are various ways of getting information from the data such as classification, exploratory data analysis, and clustering. One way to make use of the data is to highlight when there is unusual behavior and finding the outliers or anomalies in the data.

Anomaly detection can be defined as "the problem of finding patterns in data that do not conform to expected behavior" (Chandola et al. 2009). These unusual patterns are also referred to like surprises, peculiarities, anomalies, noise or outliers. The most commonly and widely used terms are anomalies and outliers (Hodge & Austin 2004). According to Chandola et al. (2009), there is a huge demand for anomaly detection in various applications such as fraud detection in credit cards, military monitoring for enemy activities and industrial damage detection.

The cause of anomalies in the data could be due to errors by instruments, human errors, malicious activities or miscalculation of missing value, therefore it is necessary to identify anomalies to maintain integrity and consistency in the data (Hodge & Austin 2004). A few examples of applications that use anomaly detection are: 1)Fake news identification 2) processing of loan applications – to identify fake applications or identify problematical users 3) identifying unanticipated records in the data – in data mining to identify errors, deception or valid but unanticipated records 4) monitoring time series – supervising safety-critical systems like high speed milling or drilling 5) intrusion detection – identifying illegitimate access in computer networks, For instance, an anomalous traffic pattern in a network could result in a hacked computer sending out confidential information to an unauthorized network (Hodge & Austin 2004).

In simple terms, an anomaly is any pattern that is different from the expected behavior. A simple approach to identify the anomaly is to find the data points which does not confide to a defined boundary, the points inside the boundary are expected to behave normally (Chandola et al. 2009). Though this seems to be a simple approach, several factors make this very challenging:

- Defining a boundary of all possible observations which are expected to be normal is difficult. Along with this, distinguishing the points that lie along the boundary between normal or an anomaly behavior is a bit challenging. (Chandola et al. 2009).
- The anomalies which are a result of malicious activity can sometimes be challenging to detect because their activities seem to be normal, which is another difficulty in defining normal behavior.
- Most of the time, noise is present in the data which is similar to the anomalies and therefore makes it difficult to identify them.
- Defining normal behavior is challenging since many domains keep evolving constantly, which may become irrelevant in the future.
- With many domains evolving constantly, a normal behavior defined in one domain cannot be applied to another domain as anomalies are different for different application domains. For instance, in the domain of medicine a small change in the normal behavior (e.g., change in body temperature) can be considered as an anomaly, However this small change in the normal behavior may not be considered as an anomaly in the stock market domain (e.g., change in the stock value) (Chandola et al. 2009).

Due to these challenges, the anomaly detection problem is generally not easy to solve.

To Identify the anomalies, many detection systems, and machine learning techniques have been developed. One way of identifying the anomalies is through clustering. Cluster analysis helps to group the data based on the behavior and structure without any previous knowledge about the data. The data points which do not confine to any of the groups can be identified as an anomaly (Thang & Kim 2011). Clustering and anomaly detection techniques are a class of unsupervised machine learning.

This thesis is focused to find efficient clustering-based anomaly detection and evaluate the performance of the algorithms. The data set used in the thesis is done by simulating the data from the MixSim r package. After performing a literature study three clustering-based algorithms were selected namely K-Means, DBSCAN, and OPTICS.

2. BACKGROUND

This section focuses on the overview of the data mining approaches, clustering algorithms followed by a more detailed view of the DBSCAN and OPTICS approaches.

2.1 Data Mining Methods

There are three basic approaches in data mining (Steinhauer & Huhnstock 2018) (Soni 2019).

Type 1: - Unsupervised Learning: - In unsupervised learning, the train datasets are not labeled. The algorithm tries to make sense from the given dataset by extracting important features in the data set.

Type 2: Supervised Learning:- In supervised learning, train datasets are labeled data. The model is fed with input and output data and then it classifies the unseen data. The data in the classification algorithm requires an equal spread of classes.

Type 3: Semi-Supervised Learning: - In this approach, the model is trained with both labeled and unlabeled datasets. The dataset contains a small amount of labeled dataset and a large amount of unlabeled dataset. The algorithm learns as the data is fed incrementally.

Supervised or semi-supervised learning algorithms for anomaly detection require labeled data in the training phase to model the behavior of the data. However, in the real world most of the data available are unlabeled, therefore to perform analysis on unlabelled data clustering is a suitable mechanism (Steinhauer & Huhnstock 2018)(Soni 2019).

2.2 Clustering Methods

There are various clustering techniques in the literature. Each technique has its own advantages and disadvantages (Saraswathi & Immaculate 2014). The techniques can be categorized as:

- Partitioning Clustering
- Hierarchical Clustering
- Density-based Clustering

Partitioning Clustering: In Partitional clustering, the dataset is partitioned level by level i.e one level partition. In the initial step, it performs a set of partitions or clusters say ‘N’, where

'N' is the number of partitions to perform. The algorithm then repeats the above step and also relocating the data points from one cluster to other clusters thus improving the partition. It divides the data into N clusters by fulfilling the below requirements.

1) At least one point is present in the group.
2) Each data point exists in the only group.

The clusters obtained from the partitioning algorithms are convex in shape and very restrictive (Ester et al. 1996). Two popular partitioning algorithms are K-means and k-medoids (Saraswathi & Immaculate 2014). The major disadvantage in Partitioning is that there can be a number of possible solutions and also the users require prior knowledge of the data which is not practical in most of the applications or domains.

K- means: - K-means is a partitioning clustering method (Saraswathi & Immaculate 2014) that focuses on separating the data points into clusters where each data point belongs to the closest mean cluster. K-means is the simplest and widely used unsupervised algorithm. It is very susceptible to outliers and noise as a small amount of such data can have a significant impact on the centroids. K- means algorithm doesn't work efficiently with an arbitrary size, shape, density clusters and also dependence on the user to define the number of clusters (Saraswathi & Immaculate 2014). The computational complexity for k-means is O(nkl) where n represents the total number of data points present in the dataset, K represents the number of clusters and l represents the total number of iterations (Tajunisha & Saravanan 2010).

K- Medoids: - The second method in partitional clustering is k-medoid, which is stronger in anomalies and noise. K-medoid represents clusters by selecting the actual data points, rather than taking the mean value of data points in a cluster as a point of reference. Medoids are the clusters' most central point, with a minimal sum of distances to other data points (Jin & Han 2011). Median is strong to the outlier and exactly determines the cluster centers, whereas the mean is affected by the anomaly and cannot determine the cluster center correctly (Jin & Han 2011).

Hierarchical Clustering: The hierarchical algorithms splits the dataset into a hierarchy of groups. The output obtained from the algorithm can be described as a tree-like structure

called dendrogram; where the root node describes the complete data set and the leaf node describes the single object of the data. The clustering creates a tree structure by merging and splitting the data. The merging and splitting operations are stopped after the required number of clusters have formed. The dendrogram can be trimmed at various levels to obtain the clustering results. The hierarchical clustering consists of divisive (top-down) and agglomerative (bottom-up) approaches (Saraswathi & Immaculate 2014).

Density-Based Clustering: Density-based clustering algorithms determine the density of the region. Density-based clustering can be used on the dataset that has noise and anomalies to find clusters of different shapes (Ester et al. 1996). The density-based clustering considers clusters as thick areas of objects in the data that are separated by low-density areas representing noise.

DBSCAN

One of the most widely used density-based clustering algorithms is Density-based Spatial Clustering of Applications with Noise (DBSCAN) which was developed by Ester et al (1996) and has been widely used in the applications of outlier detection. The basic idea behind DBSCAN is that a cluster has to contain a minimum number of points within the specified radius. DBSCAN algorithm uses two parameters i.e. minimum points (“minpts”) and epsilon(“eps”) to perform clustering. Epsilon denotes the radius and Minpts represents the minimum number of neighbors to form a cluster. The algorithm starts with a random point and continues to expand to form a cluster until the minpts and epsilon criteria are satisfied, the process continues until all the points are visited and forms a different cluster. The data points that are not present in any of the clusters are represented as noise. DBSCAN can determine an arbitrary pattern, noise and different size clusters accurately (Ester et al. 1996) (Thang & Kim 2011) however, it cannot handle data with varying density. The complexity of DBSCAN is O(nlogn), where n represents the number of data objects.

OPTICS

Another example of density-based methods is OPTICS (Ordering Points To Identify the Clustering Structure). It was developed by Ankerst et al(1999) It works similar to DBSCAN, but it identifies clusters in data of varying density. OPTICS Store the order in which the

objects are processed, additionally reachability-distance and core-distance for each object is stored. OPTICS stores the output ordering in a list called orderSeeds. The points in the order Seeds are ordered by the smallest reachability distance from the closest core point. Core distance can be defined as the smallest distance (e`) between the point A and its nearest point (B) within the radius (e) and the point(b) should be within the given MinPts range. Reachability distance can be defined as the maximum distance between the point(A) and core point(B) within the radius(e), such that reachability cannot be smaller than core distance. The core distance is set to undefine if the point is not a core point (Ankerst et al. 1999) (Chandola et al. 2009).

3. Problem

The area of anomaly detection is being used in various application domains to discover unexpected behaviors or patterns in the data. Identifying anomalies is important in decision-making applications (Ahmed et al. 2015).

The anomaly detection developed in recent years greatly depends on the advances made in the field of machine learning and artificial intelligence. With evolving technologies, large amounts of data are generated in terms of volume, velocity, variety, veracity, and value. The traditional way of detecting anomalies is inefficient with large size and high dimensionality data. In high dimensional data, detecting outliers is challenging as the data is sparse. This sparsity in the data may indicate that every point may behave like an outlier as the derived distance between points becomes similar (Aggarwal & Philip 2019).

The field of machine learning has been effective in many fields and different methods have been implemented to detect anomalies in high dimensionality data. For instance, (Song et.al 2017) have used a hybrid semi-supervised method by using deep autoencoders and ensemble k-nearest neighbors to detect anomalies in high dimensional data. (Celik et al.2011) has used DBSCAN to detect anomalies in temperature data. However, most of the research work done was on a particular application domain where each domain has different dimensionality in the dataset i.e. either on low or high dimensional data.

In this thesis, we have considered both low and high dimensionality with different levels of overlap between the clusters to find efficient clustering-based anomaly detection and evaluate the performance of the clustering algorithms. The figure1 illustrates the overlap

between clusters using two dimensions. Fig 1.1 provides a visual description of low overlap in which clusters are well separated similarly fig 1.2 indicates a high overlap in which clusters are not well separated.

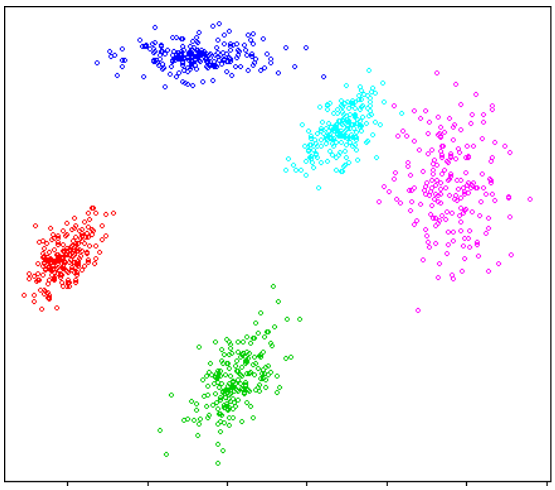


Fig 1.1 Low cluster overlap

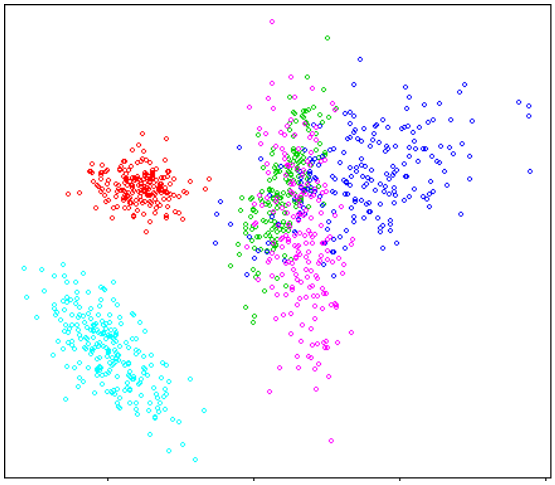


Fig1.2 High cluster overlap

Figure 1:- Illustration of low and high cluster overlap using 2-dimensional data.

4. Methods

To Identify a suitable method that is able to handle different cluster overlap and high dimensional data a literature study is carried out. The study helps to get a summary of past research. The focus of empirical research is to decide whether the proposed methods perform better compared to other methods. It is difficult to evaluate the proposed model performance on real data as we would need a lot of real data with different overlaps and is impractical. So, we intend to simulate the data as we can control the overlap and dimensions in the data.

Since the data is simulated with different cluster overlap and dimensions it is impractical to perform an analytical evaluation. The proposed methods are evaluated numerically which allows comparison of how well the methods perform in detecting anomalies with different overlaps/dimensions. The experiment is devised and implemented on a test data set and frequently used numerical evaluation metrics are applied in the area of anomaly detection. We are considering precision, recall and F1 score as evaluation metrics because these are the standard metrics used for evaluating anomaly detection (tatbul et al.2018).

5. Results of Literature Review

Previously anomalies were detected using statistical methods. The anomalies detected by these methods were based on Mean and standard deviation values. Nowadays anomalies are detected by using machine learning algorithms. A comparative study has been performed by Celik et.al (2017) using DBSCAN and statistical methods to detect anomalies in temperature data. According to Celik et al DBSCAN is more accurate in finding anomalies compared to statistical methods because the statistical approach can only find anomalies that are below and above the threshold values but it fails to detect those anomalous points that appear less frequently which are present within the threshold range. However, DBSCAN can detect anomalies that are less frequent as well as points above and below the threshold values (Celik et al. 2011).

Based on kanagala & Krishnaiah's work different clustering methods like K-Means, DBSCAN and OPTICS have been compared based on the performance measure like accuracy, cluster size, and outliers formation for different parameter values. K-Means algorithms will form clusters with good quality using large data. The drawback of K-Means is that it cannot find outliers and also does not perform well with arbitrary shapes. These drawbacks can be overcome by using DBSCAN and OPTICS. DBSCAN can form clusters with arbitrary shape and size. It can also perform faster analysis compared to other clustering methods. However, when clusters with different densities are located to each other it will not be able to distinguish them. These problems are addressed using the OPTICS algorithm. OPTICS ensures quality clustering where clusters with high density are given more priority compared to low density (Kanagala & Krishnaiah 2016).

Syarif et al. 2012 have used K-Means, k-Medoids, EM clustering, and distance-based outlier detection algorithm to detect anomalous activity in the network. In order to evaluate the above-mentioned algorithms, the intrusion dataset was used which consists of forty different intrusions categorized into four different categories. The performance metrics used were accuracy and false-positive rate. In conclusion from the experiment results, the distance-based algorithm achieved the best accuracy compared to other algorithms. However, the false-positive rate was higher for all the algorithms (Syarif et al . 2012).

Song et.al have used a hybrid model for detecting anomalies in high dimensional data. Deep autoencoder (DAE) and ensemble K-nearest neighbors have been used as a hybrid model. DAE acts as a dimensionality reduction algorithm wherein high dimensionality data is reduced to a compact size. The ensemble K-NN is used to detect anomalies on the low dimension data. Four different data sets have been used in the experiment and the performance of the hybrid model is compared with standalone algorithms like SVDD, OCSVM. The area under curve (AUC) has been used as the performance metric. The statistical analysis and experiment results conclude that the hybrid model performed better compared to other standalone algorithms (Song et al. 2017).

Tajunisha & Saravanam (2010) proposed a method to improve the performance of the k-means algorithm for high dimensional data. In the proposed method dimensionality reduction was done for the dataset by applying Principal component analysis (PCA) and then find the initial centroids. For high dimensional data, k-Means does not perform well in detecting anomalies because of the noise present in the data. Depending on how the initial centroids are chosen the accuracy of k-means varies. The proposed methods are compared with other existing methods on the iris dataset and the results show that the proposed method is accurate compared to the existing methods.

Based on the literature survey we can conclude that DBSCAN and k-Means have been widely used in detecting anomalies. The drawback of DBSCAN can be overcome by using OPTICS. k-Means does not perform well on high dimensional however by using dimensionality reduction techniques the accuracy can be improved.

5.1 Approach

From the literature research, we have chosen K-Means, DBSCAN and OPTICS methods to detect anomalies with different cluster overlap and high dimensionality

K-Means is selected because it is a simple and widely used algorithm. DBSCAN was chosen as it can automatically identify the number of clusters, it can detect clusters with arbitrary shape, size and it can also handle outliers/noise present in the data. There is not much work on OPTICS as an anomaly detector but OPTICS was chosen as it can identify clusters with a varied density in which DBSCAN fails. K-Means and DBSCAN have been used as an anomaly

or outlier detector in various applications such as network intrusion, time-series data, traffic anomaly detection and found that DBSCAN performs better in identifying outlier.

The DBSCAN will label the points as either core point, border point or Noise point. Any point in the dataset, if there are at least minPts within the radius ε is called core point. A point is a border point if they are present at the edge of the dense cluster and the number of points around them is less than the minPts. Noise Points are points that do not belong to either core or border points (Glory et al. 2012) (Ester et al 1996). Noise does not imply that the data points are outliers , it implies that the DBSCAN is not able to form clusters within the given criteria and labels as Noise.

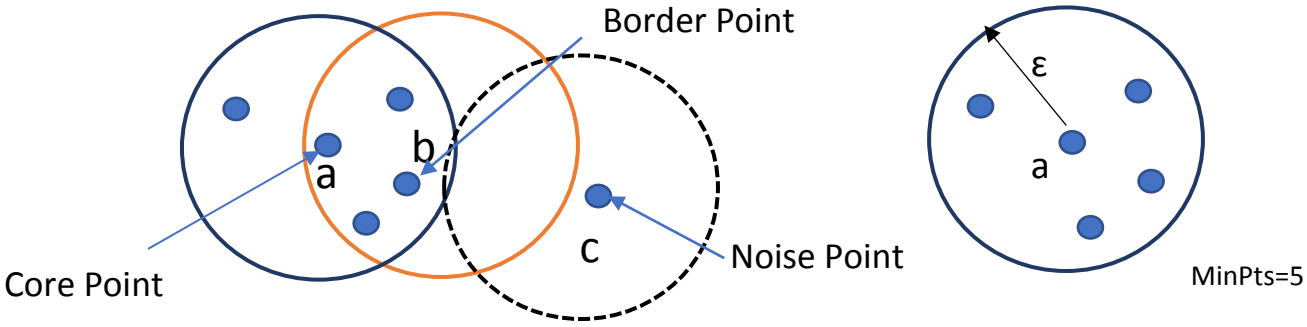
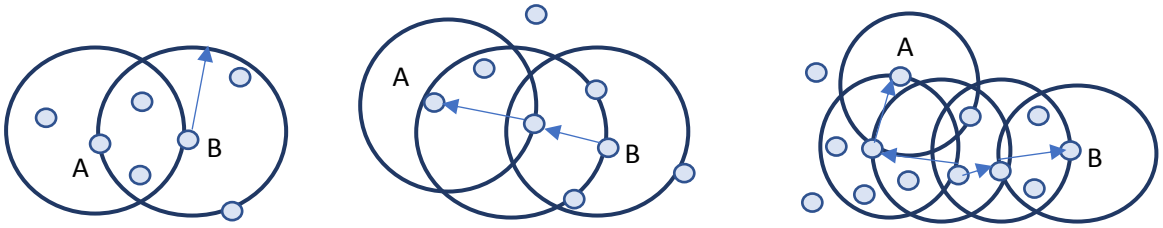


Figure 2:- Illustration of DBSCAN core point , border point ,noise point and radius(Ɛ) .

The figure2 depicts the Core Point, Border Point, Noise Point with MinPts=5. Point A is core point because it consists of minimum 5 points including point 'a' within the Ɛ radius and b is a border point because it contains points less than MinPts within the ε radius and finally point c is noise point (Glory et al. 2012) (Ester et al 1996).

From the figure3 a data point A is directly density reachable from B if data point B is in ε radius of point A and point B is a core point. A data point A is density reachable from point B if there exist core points leading from point B to point A. A data point A is density connected to point B if there exists a core point C in a way that both the points A and B are density reachable from point C with reference to MinPts and ε radius (Glory et al. 2012) (Ester et al 1996).



Directly Density reachable Density Reachable Density- Connectivity

Figure 3:- Illustration of directly density reachable, density reachable and density connected

DBSCAN Algorithm

Step1:- Select a Point "A".

Step2:- Select all the points that are density reachable from point "A" such that they are within the radius and having a minimum number of points (minPts).

Step3:- Cluster is formed if point "A" is a core point.

Step 4:- The algorithm will look for other points if point "A" is a border point and no other points are density reachable.

Step 5:- The process is repeated until all the points in the data have been visited.

Step 6:- The points that do not belong to any cluster are labeled as Noise or anomalies.

To implement DBSCAN, the parameters eps and mints value should be set. According to Hahsler et al. (2017), the value of minPts should be equal to the number of dimensions of the data plus one for better results and for eps value, K- Nearest neighbor distance is plotted against the data in the decreasing order of the distance which forms an elbow shape that is considered to be an optimal eps value. The figure4 illustrates the K-Nearest neighbor distance plot to find the optimal esp value. Since the user needs to select the eps value from the KNN plot there might be slight change in the value .

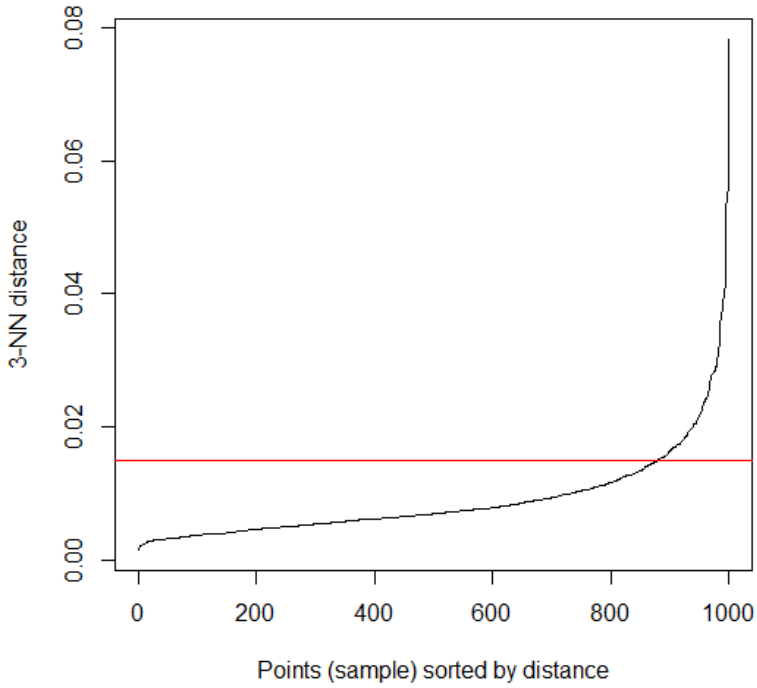


Figure 4: - Illustration of the K-NN Distance plot for choosing optimal eps value

As seen in the introduction section OPTICS works similar to DBSCAN but it can identify clusters with a varied density in which DBSCAN fails. Additionally OPTICS stores reachability distance and core distance for each object (Ankerst et al. 1999)(Chandola et al. 2009). The figure5 shows the illustration of reachability distance and core distance.

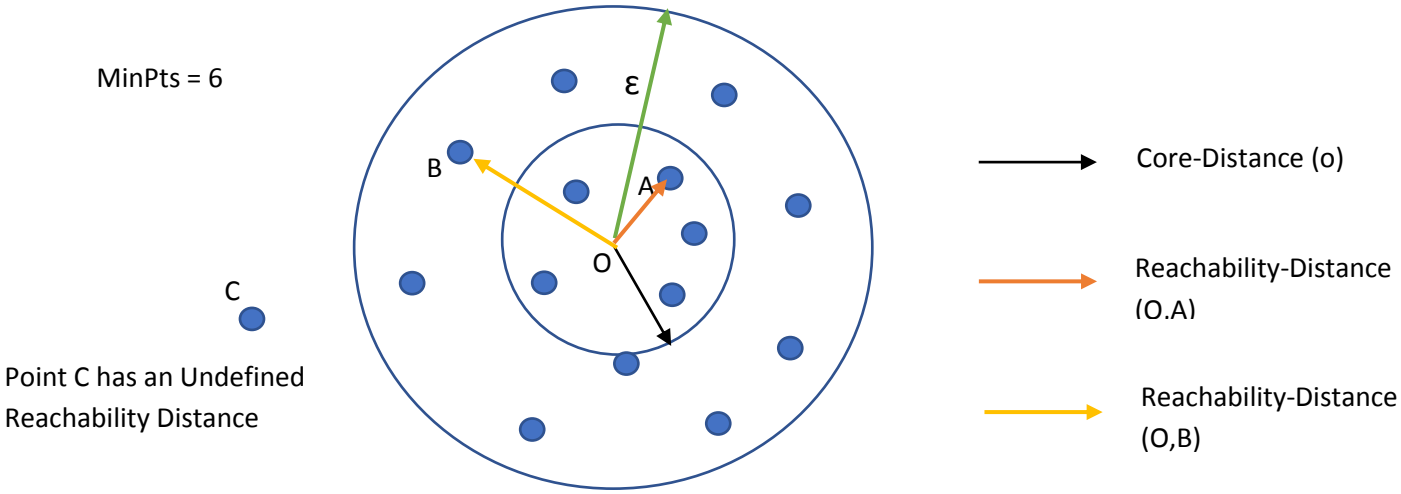


Figure 5: - Illustration of core-distance and reachability distance with minPts=6

The values of the reachability distance can't be lower than the core distance (O) as no points will be directly density reachable from O (Chandola et al. 2009).

The reachability plot shown in the figure6 helps to visualize and understand how the data is structured and clustered. The clustering order computed by the OPTICS is plotted across the X-axis and reachability distance along the y-axis. The three bumps in the reachability plot imply the clusters, the points having lower reachability distance indicate they are more similar and closer compared to points having higher reachability distance (Chandola et al. 2009). The points having higher reachability distance are labelled as noise .

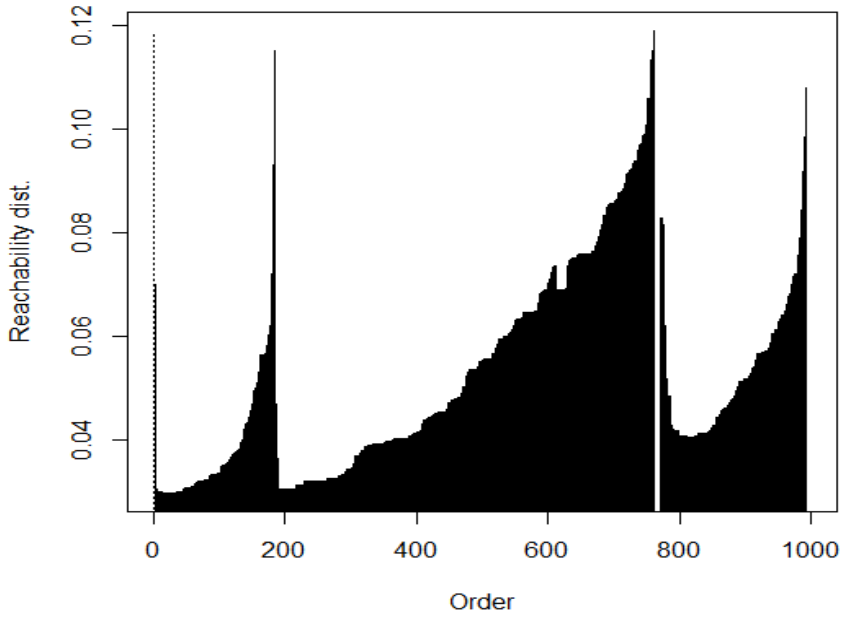


Figure 6: Illustration of the reachability plot which shows the cluster and the reachability distance of the clusters. The 3 valleys in the figure represent three clusters.

OPTICS Algorithm

Step 1:- Start with point A.

Step 2:- Fetches the neighborhood of A and determines the core distance, if the number of points in the neighborhood is less than the minPts, reachability distance is set as undefined.

Step 3:- The current point A is written to the output.

Step 4:- If point A is the core point, then for each point 'B' in the neighborhood of 'A', reachability distance from point 'B' is updated and point 'B' is pushed into the order seeds if it is not yet processed.

Step 5:- If Point A is not a core point, it moves on to the next point in the orderseeds list.

Step 6:- The process is repeated until there is no input or Order Seeds is empty.

Step 7:- The points having high reachability distance are far from the clusters and marked as anomalies.

K-Means will form clusters such that every data point belongs to one of the kth clusters and Euclidean distance is used as a similarity measure.

K-Means Algorithm

Step1: Select the number of k.

Step2: Select k random points as centroid .

Step3: Calculate the Euclidean distance between each data point to the k centroids and assign the points closest to the centroid.

Step4: Calculate the average of all the points in each cluster to find the new centroid point.

Step 5: Repeat step 3 and step4 until there is no change in the centroid (Saraswathi & Immaculate 2014).

The optimal value for k can be obtained by using elbow method or silhouette method. In this experiment silhouette method is used to identify the optimum value for k. The silhouette method was developed by Pete J. Rousseeuw. The method measures the quality of the clusters and the average distance between clusters. The silhouette plot is robust as it shows how similar or close points are within one cluster compared to other clusters (Rousseeuw 1987).The results produced by the elbow method were not reliable. Figure7 shows the silhouette plot and the vertical line indicates the optimum value for k.

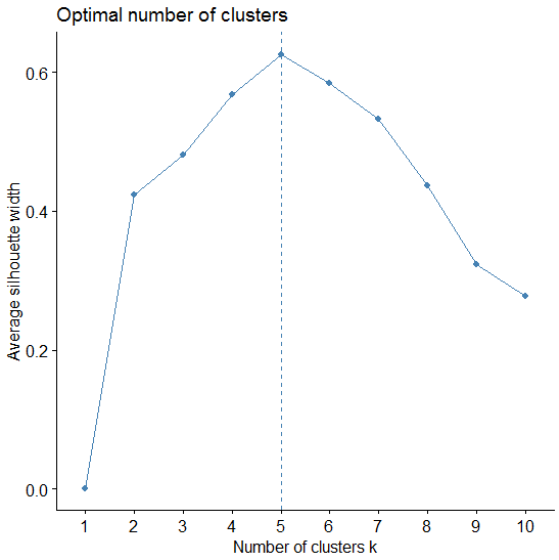


Figure 7: - Illustration of silhouette plot to select optimum k value for k-means.The vertical shows the value for k which is 5 in this case.

Identifying anomalies in k-Means is not possible as each point belongs to one of the kth clusters. However, to overcome this we incorporated a threshold value, the threshold is the maximum distance of the data point in the train data for each centroid. The distance of each data point in the test data is compared with the threshold value of each centroid and points having greater than the threshold value are marked as anomalies.

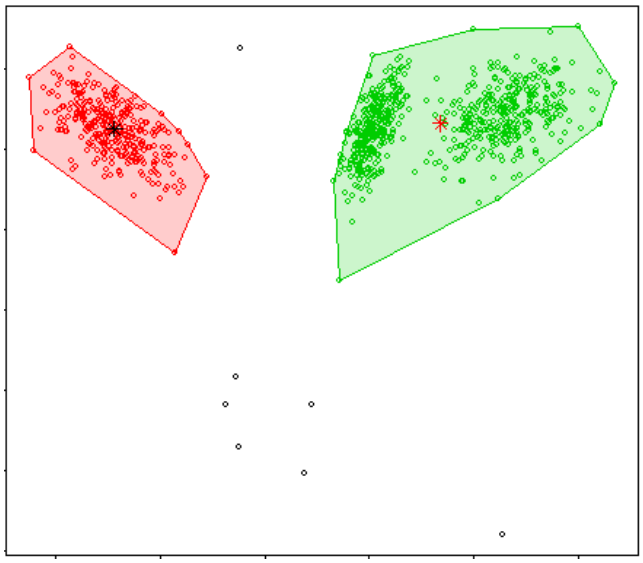


Figure 8: - Illustration of threshold value implemented in k-Means for detecting anomalies. the red and green boundary indicates the boundary for each cluster.

The figure8 illustrates the threshold value set for two clusters in train data. The threshold values obtained from the train set are 0.1952587 for the red cluster and 0.3422304 for the green cluster. The points are assigned to respective clusters (red and green color) if the distance between the points and centroids is within the threshold range and points that are outside the cluster represent anomalies as the distance to the centroid is greater than the threshold value.

6. Experiment Design

This section focuses on the dataset and metrics used to evaluate the performance of K-means, DBSCAN and OPTICS algorithm in anomaly detection

The dataset used in the experiment is generated using the MixSim package (Melnykov et al. 2012) available in the R environment. MixSim is a package to generate the data and study the performance of clustering algorithms. The package simulates the data using Gaussian

mixtures with various levels of overlap among the components and different dimensions. The cluster complexity can be controlled by using average pairwise overlap, where pairwise overlaps refer to the probability of the sum of two misclassifications (Melnykov et al. 2012).

In this thesis, the complexity of the data is varied with average overlap from low (0.001) to High (0.35) levels and with different data dimensionality. In the figure9 the left side is the low complexity data where the clusters are well separated and anomalies are in between the clusters and can be easily detected whereas the right side is the high complexity data where clusters are overlapped and anomalies are inside the overlap and difficult to identify. The points in the red circle are anomalies.

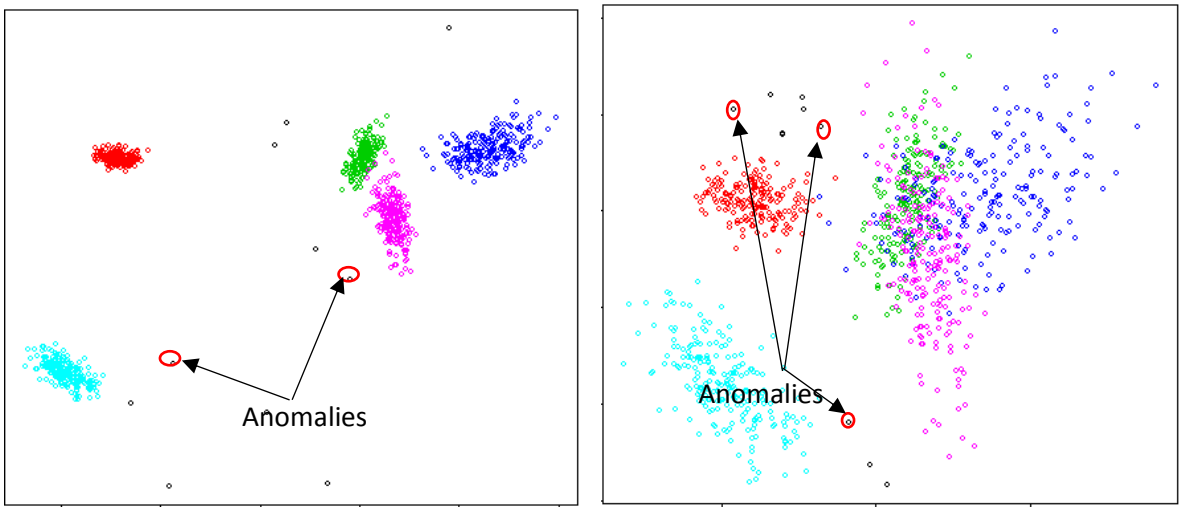


Figure 9: Illustration of anomalies present in the test data with low and high complexity

The dataset is generated with low dimensional (4D) and high dimensional data (16D), each of these dimensions consists of 10 different datasets with different levels of complexity varying from low to high overlap. The data set is divided into a train set and test set, the train set consists of normal points and the test set consists of both normal points and outliers. The table1 shows the parameters and the values used in MixSim and simulate function for generating the train data with Gaussian mixtures for low (4D) dimensionality. The BarOmega indicates the average pairwise overlap, K is the number of mixture components and p is the number of dimensions. The parameter n is the sample size of the data, Pi is mixing proportions vector, Mu is Mean vectors and S is the covariance matrices.

The parameters used for generating the test data is the same as used in generating the train data set with an additional parameter n.out for generating the outliers. In the case of high dimensionality, 10 datasets are generated (dataset 11 – dataset 20) with different overlaps. The values of BarOmega and the number of mixture components (k) remain the same as seen in the below table except for the number of dimensions (p) is set to 16.

Dataset	**Parameters**
Dataset1	MixSim(BarOmega=0.001,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset2	MixSim (BarOmega=0.005,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset3	MixSim (BarOmega=0.01,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset4	MixSim (BarOmega=0.05,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset5	MixSim (BarOmega=0.1,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset6	MixSim (BarOmega=0.15,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset7	MixSim (BarOmega=0.2,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset8	MixSim (BarOmega=0.25,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset9	MixSim (BarOmega=0.3,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)
Dataset10	MixSim (BarOmega=0.35,K=5,p=4) Simdataset(n=1000,Pi,Mu,S)

Table 1: An overview of Parameters and values used for generating the data.

There are various methods for evaluating the clustering methods for anomaly detection. We focused on three performance indicators namely Precision, Recall, and F1 Measure as they are the standard metrics used for evaluation (Tatbul et al. 2018). We are interested in using these metrics because precision measures the percentage of all detected anomalies that are true anomalies and recall measures the percentage of all true anomalies that are detected by the system. F1- Score measures the quality of the anomaly detector by combining recall and precision. F1-Score is the harmonic mean Recall and precision (Siriporn & Benjawan 2008) (Eskin et al.2002). The recall is also known as sensitivity or true positive rate.

$$\text{Recall} = \frac{TP}{TP+FN}$$

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

$$\text{F1 - Score} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall}+\text{Precision}}$$

TP refers to the number of anomalies correctly classified by the system and FP refers to the number of anomalies incorrectly classified by the system .Figure 10 illustrates an example of precision and recall

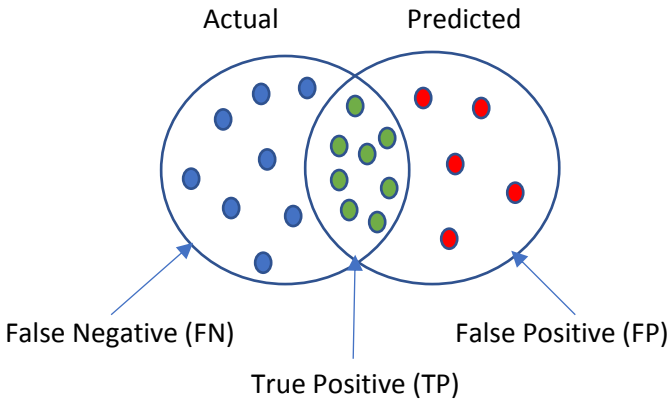


Figure 10: Illustration of performance measure Recall and Precision

The values are calculated by accessing the labels in the test set to measure the performance of the algorithms; however, the presence of label data is not assumed in real-world scenarios.

The approach to performing the experiment :

Step1: Mixsim and simulate functions are used to generate the data

Step2: Two datasets are generated, One is train data which does not contain outliers and test set which contain outliers.

Step3: K-Means, DBSCAN, and OPTICS are used to train the data and make predictions on the test data

Step4: We evaluate these algorithms with the help of the confusion matrix and get

precision, Recall, and F1 Measure.

Step 5: Step1 to Step4 is repeated with different overlap.

7. Results

The results of the experiment conducted are shown in the below tables. Table2 represents the results of low complexity and table3 represents results of high complexity.

		Dataset 1	Data set 2	Data set 3	Data set 4	Data set 5	Data set 6	Data set 7	Data set 8	Data set 9	Data set10
Methods	**Metric/Average Overlap**	**0**	**0.01**	**0.01**	**0.05**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**	**0.35**
K-Means	**Precision**	0.57	0.46	0.6	0.41	0.6	0.8	0.9	0.8	0.9	0.92
K-Means	**Recall**	0.4	0.4	0.6	0.47	0.3	0.8	0.8	1	0.9	0.55
K-Means	**F1 Score**	0.47	0.43	**0.6**	0.44	0.4	**0.8**	**0.8**	0.9	**0.9**	0.69
DBSCAN	**Precision**	0.75	0.75	0.8	0.92	0.9	0.9	0.9	1	0.9	1
DBSCAN	**Recall**	0.9	0.8	0.5	0.73	0.8	0.7	0.8	1	0.8	0.6
DBSCAN	**F1 Score**	**0.82**	**0.77**	**0.6**	**0.81**	**0.9**	**0.8**	**0.8**	**1**	**0.9**	**0.75**
OPTICS	**Precision**	1	1	1	1	1	1	1	0.9	1	1
OPTICS	**Recall**	0.7	0.33	0.3	0.53	0.4	0.4	0.6	0.9	0.6	0.2
OPTICS	**F1 Score**	**0.82**	0.5	0.4	0.7	0.6	0.6	**0.8**	0.9	0.7	0.33

Table 2: - Results of k-Means, DBSCAN, and OPTICS for 4 Dimensionality with different overlap

		Data set 11	Data set 12	Data set 13	Data set 14	Data set 15	Data set 16	Data set 17	Data set 18	Data set 19	Data set 20
Methods	**Metric/Average Overlap**	**0**	**0.01**	**0.01**	**0.05**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**	**0.35**
K-Means	**Precision**	0.71	0.94	1	1	1	1	0.9	0.9	1	0.91
K-Means	**Recall**	1	1	1	1	1	0.4	0.7	1	0.5	1
K-Means	**F1 Score**	0.83	0.97	**1**	**1**	**1**	0.5	**0.8**	**1**	0.6	**0.95**
DBSCAN	**Precision**	1	1	1	1	1	1	1	1	1	0.87
DBSCAN	**Recall**	1	1	1	1	1	0.5	0.5	0.2	0.5	1
DBSCAN	**F1 Score**	**1**	**1**	**1**	**1**	**1**	**0.6**	0.6	0.3	**0.7**	0.93
OPTICS	**Precision**	1	1	1	1	1	1	1	1	1	0.91
OPTICS	**Recall**	1	1	1	1	1	0.3	0.3	0.2	0.2	1
OPTICS	**F1 Score**	**1**	**1**	**1**	**1**	**1**	0.5	0.4	0.3	0.3	0.95

Table 3: - Results of k-Means, DBSCAN, and OPTICS for 16 Dimensionality with different overlap

As discussed in the experiment design section F1 Score is used as the final metric for evaluating the algorithms as it a combination of both Recall and Precision. The figure11 shows the graph of F1-Score for K-Means, DBSCAN, and OPTICS for 4 Dimensions. We can see that DBSCAN performed well when compared to K-Means and OPTICS

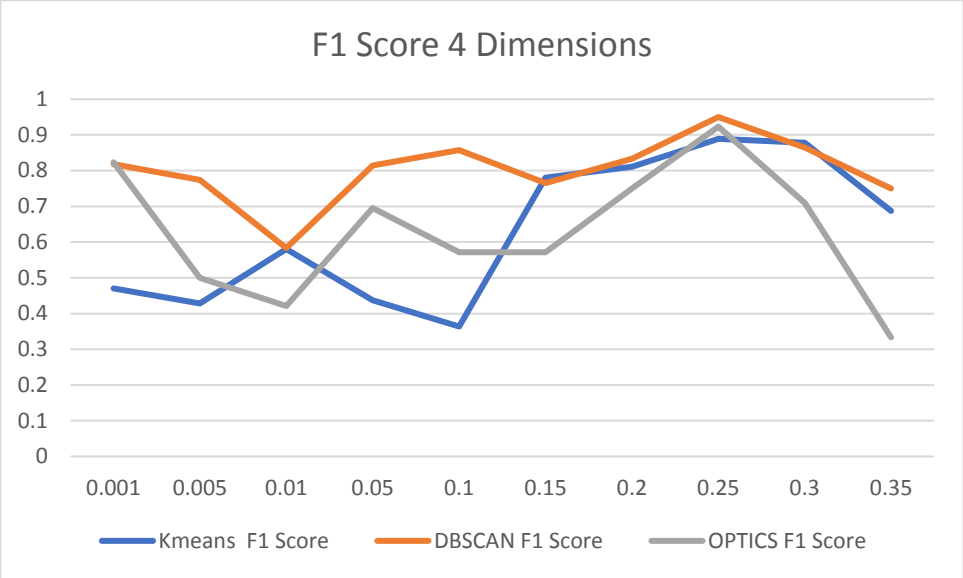


Figure 11: - F1 Score comparison of K-Means, DBSCAN, and OPTICS with different cluster overlap for 4 Dimensions.

Figure 12 shows the F1 Score for 16 Dimensions. For high dimensionality, we see that for overlap between 0.001 to 0.1 all the three algorithms have the same F1 Score and as the overlap increases, K-Means performed well compared to DBSCAN and OPTICS.

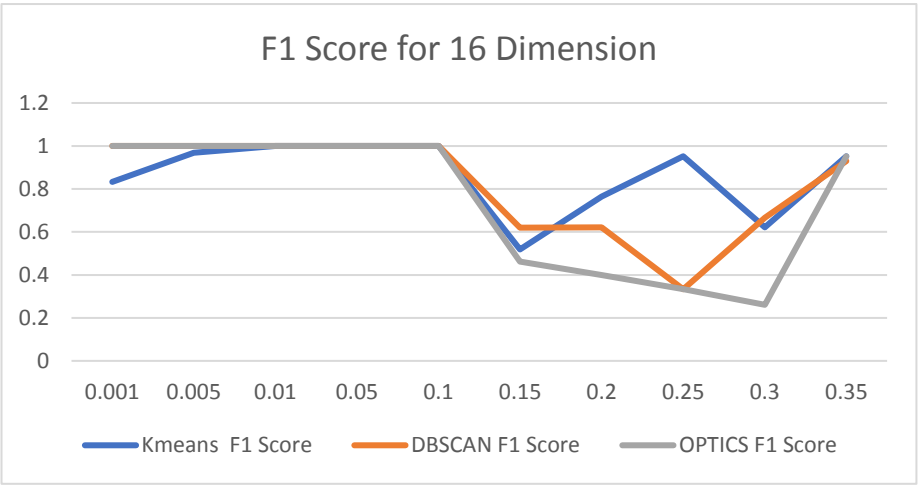


Figure 12: - F1 Score comparison of K-Means, DBSCAN, and OPTICS with different cluster overlap for 16 Dimensions

8. Discussion

The research aim is to find efficient clustering-based anomaly detection and evaluate the performance. From the results obtained, DBSCAN performed better than other clustering algorithms for low dimensional data with varying clusters overlap and this justifies the assumptions made from the literature survey. K-Means performed better compared to DBSCAN and OPTICS for high dimensional data with varying cluster overlap, this result goes against the assumptions made were the DBSCAN would perform better for high dimensional as per the literature survey was done. These unexpected results may be due to the threshold value in k-means the threshold was set after the clusters were formed in the train data set. The application domains like finance, military surveillance requires faster analysis therefore, DBSCAN can be used in these domains as computational complexity less compared to k-Means.

The results suggest that the DBSCAN and K-Means can be used in application domains where the data follows the gaussian mixture technique. The benefit is limited from a social perspective as we have experimented on a simulated data set in a controlled environment without considering the factors involved in real-world scenarios such as noise. The model may or may not perform well in real case scenarios.

In any application domain, the main aim is to detect all of the anomalies which are ideal in a real case. False Negatives are formed when the algorithm fails to classify as anomalies and considers as a normal point. False Negatives can cause ethical issues in all application domains which affects both the society and the industry. For example, while detecting criminal behavior using surveillance, the algorithm should be able to match the behavior of the criminal activity and then classify them as an anomaly. If in case the algorithm matches those activities as normal behavior it can cause adverse effects to the society and many legal issues in using the model.

The limitation of threshold method used in K-Means to identify the anomalies will work with clusters with circular shape and might not be appropriate for non-circular shapes as the normal points outside the threshold region would be labelled as anomalies. This can be taken into further analysis

9. CONCLUSION

In this thesis, the problem of anomaly detection that is used in various application domains was studied. From the previous research, we found out that different methods have been implemented to detect anomalies and by analyzing those results we have chosen unsupervised clustering algorithms namely K-Means, DBSCAN and OPTICS as a suitable approach to detect anomalies on different dimensionality and cluster overlap. In addition, an outline of methods for anomaly detection was discussed and proposed methods were explained in detail. An experiment was designed to check the performance of the clustering algorithms for different data dimensions with varied cluster overlap settings. The data is simulated using the MixSim package available in R . Performance metrics such as Recall, precision, and F1 score were used to evaluate the cluster algorithms. The results obtained from the experiments were reported followed by a discussion section about anomaly detection.

In this work, we have considered only a few parameters for generating the data from the MixSim package. Further analysis can be done by evaluating the performance of the algorithms by adding noise, outliers to the training dataset and generating clusters of non-spherical form. The F1-score of K-Means and DBCSAN differed by a small value for few clusters overlaps but the significance of these differences was not evaluated and this can be taken into consideration for further analysis.

Reference

Aggarwal, C. & Philip, S. Y. (2001). Outlier detection for high dimensional data. *In Timos Sellis and Sharad Mehrotra (Eds.) Proceedings of the 2001 ACM SIGMOD international conference on Management of data (SIGMOD '01)*. New York, USA: ACM, pp.37-46. doi: https://doi.org/10.1145/375663.375668

Alguliyev, R. et al. (2017). Anomaly Detection in Big Data based on Clustering. *In Statistics, Optimization & Information Computing*. doi: 10.19139/soic.v5i4.365.

Ankerst, M et al. (1999). OPTICS: Ordering Points To Identify the Clustering Structure. *In Proceedings of the 1999 ACM SIGMOD international conference on Management of data (SIGMOD '99)*. ACM, New York, pp. 49-60. doi:10.1145/304182.304187.

Celik, M. et al. (2011). Anomaly Detection in Temperature Data Using DBSCAN Algorithm. *In International Symposium on Innovations in Intelligent Systems and Applications.* June 15-18, 2011.Istanbul: IEEE.doi 10.1109/INISTA.2011.5946052.

Chandola, V. et al. (2009) Anomaly detection: A Survey. *In ACM Computing Surveys.* pp. 55. doi: 10.1145/1541880.1541882.

Ester, M. et al. (1996), A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*.Portland: ACM, pp. 226-231.doi: 10.1023/A:1009745219419

Eskin, E. et al. (2002) A Geometric Framework for Unsupervised Anomaly Detection. *In: Barbará D., Jajodia S. (eds) Applications of Data Mining in Computer Security. Advances in Information Security*, Springer, Boston, MA. doi: 10.1007/978-1-4615-0953-0_4

Glory, H. et al. (2012) An Empirical Evaluation of Density-Based Clustering Techniques. *International Journal of Soft Computing and Engineering.*ISSN. pp. 2231-2307.

Hahsler, M. et al. (2017). dbscan: Fast Density-based clustering with R . *Journal of Statistical Software, 91*(1),pp.1 - 30. doi:10.18637/jss.v091.i01.

Hodge, V & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *In Intenational Conference on Artificial Intelligence and computational Intelligence* USA:ACM,pp. 85-126. doi:10.1023/B:AIRE.0000045502.10941.a9.

Steinhauer, J & Huhnstock, N. (2018) Advanced Artificial intelligence[power point presentation]. Skovde: Hogskolan i skovde.

Jin X., Han J. (2011) Partitional Clustering. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning*. Springer, Boston, MA, doi:10.1007/978-0-387-30164-8.

Jin X., Han J. (2011) k-Medoids Clustering. *In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning*. Springer, Boston, MA, doi:10.1007/978-0-387-30164-8

Kanagala, H & Krishnaiah, V.V. (2016). A comparative study of K-Means, DBSCAN, and OPTICS. *In International Conference on Computer Communication and Informatics.* pp 1-6. doi:10.1109/ICCCI.2016.7479923.

Li, L. (2011). Anomaly detection in onboard-recorded flight data using cluster analysis. *In 2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, pp. 4A4-1-4A4-11. doi: 10.1109/DASC.2011.6096068.

Mahendru, K. (2019) . "How to determine the optimal value for k",[blog] ,Jun 17 https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb [2019-12-21]

Melnykov, V. et al. (2012). "MixSim: An R Package for Simulating Data to Study Performance of Clustering Algorithms." *Journal of Statistical Software* [Online], 51.12: pp.1 - 25. Web. 25 Aug. 2019

Rousseeuw, P.J.(1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". *Computational and Applied Mathematics*. 20: pp.53–65. doi:10.1016/0377-0427(87)90125-7.

Saraswathi, S. & Sheela, M.I., (2014). A comparative study of various clustering algorithms in data mining, I*n International Journal of Computer Science and Mobile Computing*, November- 2014, pp. 422-428

Shah, G.H., & Ganatra, A. (2012). An Empirical Evaluation of Density-Based Clustering Techniques.In the International Journal of Soft Computing and Engineering, ISSN. pp. 2231-2307.

Song, H. et al. (2017). A Hybrid Semi-Supervised Anomaly Detection Model for High-Dimensional data.*Computational Intelligence and Neuroscience*, doi:10.1155/2017/8501683.

Soni, D. (2019) Understanding the different types of machine learning models, [blog],Aug 25. https://towardsdatascience.com/understanding-the-different-types-of-machine-learning-models-9c47350bb68a [2019-10-26].

Siriporn, O. & Benjawan, S.(2009). Anomaly detection and characterization to classify traffic anomalies Case Study: TOT Public Company Limited Network. *International Journal of Computer and Information Engineering*. pp 15-23. doi:10.5281/zenodo.1078213.

Syarif, I et.al (2012) Unsupervised Clustering Approach for Network Anomaly Detection. *In: Benlamri R. (eds) Networked Digital Technologies. NDT 2012. Communications in Computer and Information Science*, Springer, Berlin, Heidelberg

Tatbul .N et al. (2018). Precision and Recall for Time Series. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18).Canada, pp.1924-1934.

Tajunisha, & Saravanan. (2010). Performance analysis of k-means with different initialization methods for high dimensional data. *International Journal of Artificial Intelligence and Applications.* doi: 10.5121/ijaia.2010.1404

Thang, T. & Kim, J. (2011). The Anomaly Detection by Using DBSCAN Clustering with Multiple Parameters. *International Conference on Information Science and Applications, Jeju Island*, pp. 1-5. doi: 10.1109/ICISA.2011.5772437.

Zhao Z., Mehrotra K.G., Mohan C.K. (2018) Online Anomaly Detection Using Random Forest. *In: Mouhoub M., Sadaoui S., Ait Mohamed O., Ali M. (eds) Recent Trends and Future Technology in Applied Intelligence. IEA/AIE 2018. Lecture Notes in Computer Science*, Springer, Cham