MASTER IN BUSINESS ENGINEERING PROFESSIONAL FOCUS IN DATA SCIENCE

Outlier Detection and Evaluation Methods for Classification

Davidt, Michel

Award date:
2021

Awarding institution:
University of Namur

Outlier Detection and Evaluation Methods for Classification

Michel DAVIDT

Directeur: Prof. Benoît FRENAY

Mémoire présenté
en vue de l'obtention du titre de
Master 120 en ingénieur de gestion, à finalité spécialisée
en data science

ANNEE ACADEMIQUE 2020-2021

Université de Namur, ASBL
Faculté des Sciences économiques, sociales et de gestion – Département des Sciences de gestion

Rempart de la Vierge 8, B-5000 Namur, Belgique, Tel. +32 [0]81 72 49 58/48 41

Contents :

1. Introduction

An outlier is a data point that differs from other points. These are present everywhere, in any possible real-world area. Inside a dataset, they are more visible than any other common data point. They can be a good subject of study, or possibly falsify the results: that is reasons of why outliers are important to take into account and to detect them.

It is almost impossible to avoid outliers, regardless of the origin of the dataset (except if it has been made synthetically). Depending on the application, the outliers can be handled differently. Another important point is to evaluate these methods used to detect the anomalies. This will be more explained in the next chapter, after defining the word outlier. As for the evaluation methods, the importance will be explained later.

Outlier detection is a large subject in data mining, present in a lot of different types of data and in a large variety of applications (fraud detection, illnesses detection, cyber security, etc). When a model is trained and used for outlier detection, it can be used for detecting outliers that the model didn't learn, it is called Novelty Detection [1]. A lot of different techniques have been introduced by different researchers, but also different types of techniques: there are techniques are based on distance between data points, others are based on density of the data points, or even others that are based on clustering. Five techniques will be described later in this thesis.

1.1 Applications

There are a wide number of applications for outlier detection, in a lot of different areas. Here are different applications with different areas:

1) Fraud Detection: When a card a stolen by someone, the purchasing behavior of the card user will be different than the purchasing behavior of the owner. If such a change is detected, banks can intervene to block the card.

2) Health Care Analysis: Outlier detection can be used in the medical area. Unusual patterns for a body can be taken from different devices to detect an unhealthy person and act on it. An example could be the detection of abnormal fetal hearts (a dataset with measures for fetal hearts has been used in the experimental results, see chapter 5).

3) Value-at-Risk [2]: The Value-At-Risk (or VaR) is a concept used by financial institutions (more exactly the banks) to evaluate their market risk of one or several financial assets at the same time. The market risk corresponds to the risk occurred when a financial asset has variation on its price. The

VaR consists in the worst potential but feasible loss. It could actually corresponds to an outlier's definition, on a time series.

4) Sensor networks: Data can be collected by sensors network. A sensor network includes a mix of different sensors, collecting different types of data: discrete, continuous, or even audio and video. However, outliers (noise) can make the sensors not working as they should (it is called sensor fault detection) [3].

5) Industrial Damage Detection: The industrial machines can have damages if they are used continuously. They need to be detected as soon as possible to be sure that the losses are minimized. To collect data about it, there is need to use sensors.

1.2 Contribution

The aim of this thesis is to show that all models does not detect the same outliers and so, have a difference of performance (with a variation of a parameter, depending on the outlier detection technique), with focus on the classification, but also to show that depending on the dataset, the performance could be different (synthetic versus real world dataset, see the chapter 5, section 2), and the reasons why.

Also, the idea is to show that the presented evaluation methods have disadvantages, and that, in practice, it is not that easy to compare different models (sometimes limited by their detection level (see the chapter 5 and 6).

1.3 Plan

The chapter 2 gives a more detailed definition of an outlier, the possible causes of it and different possibilities to handle it. By defining the outliers, different families of outlier detection techniques are also given.

The chapter 3 describes the five different techniques: he KNN and the Neighborhood, the Local Outlier Factor, the Isolation Forest, the Gaussian Mixture Model, and the DBSCAN. Other statistical tests have been mentioned. Other techniques related to the different groups are briefly mentioned and the advantages/disadvantages of the five principal techniques explained are also reviewed.

The chapter 4 describes the evaluation procedure, i.e how to evaluate and compare the performance of the models. These are the two evaluation methods mentioned: the Precision and the Receiver Operating Characteristic curve coupled with the Area Under the Curve.

The chapter 5 gives the methodology to build the different methods and also the evaluation method, and the interpretations of the results are made, with the help of tables and figures (see the Appendix).

The chapter 6 and the last one is the conclusion. It gives the limit of the results obtained, the potential questions that could be asked and the recommendations for a future study.

2. Outlier – definition, causes and handling

This chapter gives several definitions of an outlier in the first section, where the idea is the same for each. Then, several causes of the outliers are mentioned in the second section, and finally how the searchers generally handle them, in a technical way or not, the last section.

2.1 What is an outlier?

There is no real definition of what an outlier is. To show how difficult it is to really define what an outlier is, Ayadi et al (2017) [4] reviewed twelve different versions of the definition of an outlier, from different authors, but without mention them here. The definition of an outlier depends of the technique used and also the data structure (points, time series, etc.). In order to give general definitions, here are the definitions that Ben-Gal (2005) [5] reviewed:

- Hawkins (1980): an outlier is « an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. »
- Barnett and Lewis (1994): «An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs. »
- Johnson (1992): an outlier is « an observation in a data set which appears to be inconsistent with the remainder of that set of data. »

All these definitions give the same idea: outliers are the type of data points that are well distinct from others. They are dissimilar from the inliers. But as mentioned before, the definition of outliers depends on the technique used to detect them, more exactly the type of technique. Wang et al (2019) [6] resumed groups of outlier detection techniques:

The *statistical-based* (or distribution-based) *methods*: identifying the outliers is made by using a certain distribution model. A relevant definition for it is the statistical definition mentioned before for a normal distribution. Two different types exist: the parametric and the non-parametric.

The *distance-based methods*: the outliers here are detected by computing the distance between the data points. Then, an outlier is a point that is far away from his neighbors.

The *density-based methods:* the outliers depend on the density of the data points. A point in a low-density region will be considered as an outlier.

Clustering-based methods: the idea behind these methods is simply to make clusters of the data points. The outliers detected by this kind of techniques will be the data points that are not included in the clusters built.

Graph-based methods: these techniques use graph methods to detect outliers. An outlier in that case would be a data point with much less connectivity (less edge connecting this point) than the majority of the other points.

Ensemble-based methods: in these methods, there is no real « definition » for the outliers. The methods combine the results of dissimilar models to make them more robust. It helps to detect outliers more efficiently. They can help to define the outliers (for example, should the outliers be distance-based?)

Learning-based methods: these techniques use active learning (learning with interactivity with the user) and deep learning. The idea is to learn different models with these methods to detect outliers.

2.2 Causes of an outlier

As said before, it is almost impossible to have no outlier in a dataset. Osbourne and Overbay (2004) [7] reviewed possible causes of the outliers. Here are three of the reasons given in the article which are principally human mistakes:

- *Data error*: outliers can be caused because human may make mistakes when it comes to data collection, recordings or encoding.
- *Intentional/Motivates misreporting*: in case of a survey, participants may deliberately, for different reasons, to report false data. A similar idea would be the adversarial machine learning, where an opponent could deliberately add false data points (see Huang et al [8]).
- *Sampling error*: if we build a complete dataset with different classes and that we sample it randomly. The problem could be that in one sample, only one point inside the sample represents one class, making it an outlier of the sample. With the use of clustering (building class), outliers could be inside one cluster but in reality it should not belong to the same class as the other points of it, because, for example, the dataset didn't have enough data points while with more points the outlier could perhaps have formed with others a class of its own.

If, for example, we consider an univariate time series that measures the temperature of a specific location every day. During the majority of the day of one year, the temperature will be more or less normal for the season, but sometimes, the temperature measured would be extreme values, whether it is positive or negative values. In another words, outliers can be created by “random” events (for temperature, weather conditions such as a heat wave or cold snap, or in the field of finance, the information news of a company varies the price of his shares).

2.3 Handling the ouliers

When outliers are detected, it is important to know how to handle them. The handling of the outliers would depend of the cause, the objective and the application. Aguinis et al (2013) [9] reviewed possibilities to handle them:

- *Correct the value of the outlier*: it corresponds to correct the outlier and give the real values of the data point. It can happen only if there is a human mistake (deliberate or not) behind.
- *Remove the outlier*: corresponding simply to the elimination of the outlier. Eliminating a data point results to a loss of information, especially if the point is an outlier.
- *Study the outlier in detail:* the idea is to conduct a complete follow-up study. It is a way to know how this outlier appeared, where does it comes from and why is it here.
- *Keep*: it is simply keep the outliers, acknowledge that there are outliers, but doing nothing about it.
- *Report the results with and without the presence of outliers*: the idea is to report the two different reports and provide an explanation of the difference between both results.

These are only five outliers handling technique given by Aguinis et al, but there are twenty different techniques in total and there are more technical ways to handle.

In conclusion, defining an outlier can be very wide, depending on the family of outlier techniques. The causes mentioned are principally human mistakes, but are not the only type. For instance, extreme temperatures are “natural”, and are not a mistake or a false reporting. The five techniques given to handle are the most common, but others are more technical.

3. Outlier Detection Methods

A lot of research has been made for outlier detection, and more particularly to build different methods. In this thesis, we will have a focus on the detection with the K Nearest Neighbors, the Local Outlier Factor, the Isolation Forest, the detection with the Gaussian Mixture Model and the detection with DBSCAN. Statistical tests will be part of the first section, while the five other techniques form a section in the same order. These techniques are from different families of methods, and other techniques from the same family are given in the corresponding section. In the case of the Local Outlier Factor, these are improvements. Each technique will be described in detail and mathematically with the outlier score.

3.1 Statistical Tests

Straightforward approaches have been introduced to detect the outliers. There are statistical tests, reviewed by Walfish (2006) [10] :

The **Box plot**: the box plot is a representation of the dispersion of the dataset. It represents Q_1, Q_3 and Q_2, where Q_1 is the lower quantile (25th percentile), Q_3 the upper quantile (75th percentile) and Q_2 the median (50th percentile). The outliers are in what is called the lower and upper fences, set at both sides of the interquantile range (equal to $Q_3 - Q_1$). Any point that exceeds these fences can be considered as outliers. However, it works for

The **Trimmed mean**: the Trimmed mean corresponds to the mean discarded of a certain pourcentage of the data. To give an example, for a percentage of 5%, the 2.5 % highest scores and the 2.5% lowest scores would be discarded of the data and the mean would be computed on the remaining data points. The advantage of the trimmed mean is the resistance to outliers and then can represent the population mean. The outliers would not influence the trimmed mean since a part of them are rejected, depending on the percentage.

The **Extreme Studentized Deviate**: or called ESD, it is a test good in order to identify outliers in an univariate, normal sample. The test looks at the maximum deviation from the mean

$$R = \frac{|x_i - \bar{x}|}{\sigma}$$

where x_i corresponds to a data point, $\bar{x}$ is the mean and σ the standard deviation of the dataset. The value R is compared with a tabled value called critical value, depending on the confidence level wanted. If R exceeds the critical value, then the data point x_i is considered to be an outlier.

The **Dixon-type** test: the Dixon-type test is based on a ratio on the ranges. Data points are ordered such that $x_1 < ... < x_n$. This test is flexible enough to use it for specific data points and works well with small sample size. There are different ratios to identify outliers, depending on which data point the test is made. For example, for the highest/smallest data point or the second highest/smallest data point

$$R_{10} = \frac{x_n - x_{n-1}}{x_n - x_1} \qquad R_{10} = \frac{x_2 - x_1}{x_n - x_1}$$

$$R_{11} = \frac{x_n - x_{n-1}}{x_n - x_2} \qquad R_{11} = \frac{x_2 - x_1}{x_{n-1} - x_1}$$

where the 2 first equations are for the highest/smallest data point while the 2 equations below are for the second highest/smallest data points. If the distance between a data point and his neighbor is high enough, then R would exceed the critical value and be considered as an outlier.

A simple possibility to detect outliers has been given by Davies and Gather (1993) [11]. For any confidence coefficient α (called also the coverage rate) between 0 and 1, the α-outlier region for a normal distribution N(μ, σ²)

$$out(\alpha, \mu, \sigma^2) = \{x : |x - \mu| > \; z_{1-\alpha}\sigma\}$$

where x is a data point inside the α-outlier region, and z_q is the q^{th} quantile of the standard normal distribution.

Goldstein and Dengel (2012) [12] introduced the histogram-based outlier score (HBOS), a fast unsupervised method for outlier detection. For each feature of the dataset, the algorithm builds an univariate histogram. When the feature is categorical, the value of each category is counted and the relative frequency is computed. For numerical features, there is two different methods that can be used:

- A static bin-width histogram (the standard one);
- A dynamic bin-width histogram.

The standard technique uses k width bins and put the frequency into each of them to estimate the density and it would correspond to the height of the bins. For the dynamic histogram, it works as follows: First the values are sorted, then, a fixed amount of successive values (n/k, where n is the number of data points inside the dataset and k is still the number of bins.) are grouped into one bin. Each bin has then the same number of data points and so the area is the same for each bin. The width of the bins is defined by the first and the last value. Then, the height of each individual bin can be computed. The bins that cover a larger interval (larger difference between the first and the last value) will be less high than the others, so less density. Except when more than k data points would have the same value. In this case, more than n/k values should be on the same bin.

As said before, for each dimension a, the individual histogram is built, where the height of bins corresponds to the density estimation. The histograms are normalized (the maximum height should not excess 1), so that the weight for each histogram is equal. The HBOS score for a data point x is calculated as follows, for each feature

$$HBOS(x) = \sum_{i=0}^{m} \log(\frac{1}{hist_i(x)})$$

where $hist_i(x)$ is the corresponding height of the bin for the feature i. This technique assumes the independence of the features.

3.2 Detection with the help of K Nearest Neighbors

The K Nearest Neighbors can be used as an unsupervised learning technique for outlier detection. It was introduced by Chen et al (2010) [13]. The technique uses the KNN in order to use the mean distance between a data point and his k neighbors. A certain threshold can be fixed on the computed mean k-distance to distinguish outliers to inliers.

To give more details about the technique, we can define an information system to model for an unsupervised approach as (U, A, V, f). U is called an Universe, it is a non-empty finite set of data points. A contains the features (with m the total number of features) and V is the union of the feature domains, such that

$$V = \bigcup_{a \in A} V_a$$

where V_a is the value domain of the feature a. f : U x A -> V is called information function such that $f(x,a) \in V_a$, for each $a \in A$ and $x \in U$. By considering U and a distance function D : f(x,y) -> R^+, R^+ is the set of real positive values. The neighborhood $n^q_B(x)$ of x in the subset B (included in A), for any $x \in U$, $q \in R^+$ is

$$n_B^q = \{y | x, y \in U, D_B(x,y) \leq q\}$$

where D is a distance function. This distance function satisfies 4 properties:

- The value of the distance function is positive ($D_B(x,y) \geq 0$)
- $D_B(x,y) = 0$ only if x and y are one and the same.
- The distance function is symmetric, meaning that $D_B(x,y) = D_B(y,x)$.

- The triangular inegality : $D_B(x,y) + D_B(y,z) \geq D_B(x,z)$.

When the distance function is applied on a set of data points, it is called a distance metric. 3 metrics are used most of the time, and can be summarized by the Minkowsky distance, defined as follows

$$D_p(x,y) = (\sum_{i=1}^{m} |f(x,a_i) - f(y,a_i)|^p)^{\frac{1}{p}}$$

where $f(x,a_i)$ corresponds of the value of x on the ith dimension of the space A, and p is a parameter that gives several distances possibilities :

- When p = 1, the Minkowsky distance refers to the Manhattan distance.
- When p = 2, the Minkowsky distance equals the Euclidean distance.
- When p tends to infinite, the Minkowsky distance is the Chebyshev distance.

Then, for outlier detection, the technique uses a simplified version of VDM (Value Difference Metric) introduced first by Stanfill and Waltz (1986) [14]. The aim is to provide an appropriate distance function and is defined as follows

$$VDM(x,y) = \sum_{a \in A} d_a(x_a, y_a)$$

where x and y are data points with a possible distance to compute, x_a is the value of x on feature a and $d_a(x_a,y_a)$ is the distance between the value of x and y on feature a. It is defined as

$$d_a(x_a,y_a) = (\frac{|n_a^{q_a}(x)|}{|U|} - \frac{|n_a^{q_a}(y)|}{|U|})^2$$

where |U| = n is the number of data points and $n_a^q(x)$ is the neighborhood of the data point on the feature a as defined above.

Chen et al [13] then defined the NOOF (Neighborhood-based Object Outlier Factor), that measures the degree of outlierness based on the KNN. The NOOF is equal to the sum of all the VDM between one data point and all the others

$$NOOF(x_i) = \sum_{j=1, j \neq i}^{n} VDM(x_i, x_j)$$

and by fixing a certain threshold ϑ, for any data point, if $NOOF(x) > \vartheta$, then the data point x is called a neighborhood-based outlier. If not, then the data point is considered to be an inlier.

The algorithm described above has a time complexity (the amount of time necessary to run the algorithm) of O(m*n²), while he has a space complexity (the amount of memory space necessary to solve a computational problem. In this case, detecting outliers) of O(m*n), where m is the cardinality of A and n the cardinality of U.

K-Nearest neighbors graph

Other techniques have been implemented with the use of KNN. Indeed, Hautamäki et al (2004) presented ODIN, an Outlier Detection algorithm using Indegree Number, with the help of a k-nearest neighbor graph [15]. A k-nearest neighbor graph is a weighted directed graph where every node represents a vector, while the edges between the nodes correspond to pointers to their neighbor vectors. Each node has k edges, the k nearest neighbors, computed by using a given distance function. The construction of the graph takes a time complexity of O(N²).

In this case a node is considered to be an outlier, given a KNN graph created for a certain dataset when this node's indegree is less than an indegree threshold. The indegree threshold is defined as follows

$$T = \max(L_i - L_{i-1}) * t$$

where t ∈]0,1[is a parameter defined by users and L_i is the k-distance (defined in the next section) of the ith vector. To put it differently, T is the maximal distance between a vector and the previous one multiplied by a parameter fixed by the user.

Advantages and Disadvantages

A KNN model for outlier detection, just like other models, has advantages and disadvantages. Wang et al (2019) made a paper that reviews a large variety of different techniques for outlier detection and gave advantages, disadvantages and gaps of every type of outlier detection technique. The KNN is classifies as a distance-based approach [6]. This kind of approach does not rely on an assumption on the data distribution.

However, in high dimensional space, when data points have a high number of features, the neighborhood and the KNN are expensive. It is also difficult for KNN to work well in presence of data stream.

3.3 Local Outlier Factor

The LOF (Local Outlier Factor) is an unsupervised outlier detection technique based on a data point's density, made by Breunig et al (2000) [16]. It uses the KNN as model. The idea is to compute an outlier score that compares a data point's density to the density of his neighbors. When the outlier score of a data point is high, it means this point is more likely to be an outlier.

The LOF score has different concepts need to be defined. First, we have the k-distance, defined by Breunig et al as follows: the k-distance of a data point x (k-dist(x)) is the distance between p and another point x ∈ D (D is the dataset) such that:

- For at least k objects with y' ∈ D\{x}, d(x,y') ≤ d(x,y)
- For at most k-1 objects with y' ∈ D\{x}, d(x,y') < d(x,y)

In another words, the k-distance corresponds to the max distance between a data point and his k neighbors.

Then, Breunig et al defined the k-distance neighborhood of a data point x. Given the k-distance of p, the k-distance neighborhood of x contains every data point whose distance from x is less great than the k-distance

$$N_{k-distance(x)}(x) = \{y \ \in D\backslash\{x\}|\ d(x,y) \leq k - distance(x)\}$$

where y are data points and the k nearest neighbors of x and k-distance(x) the k-distance defined above.

The LOF technique uses also the reachability-distance. The reachability-distance of data point x with respect to data point y is

$$reach - dist_k(x,y) = \max\{k - distance(x), d(x,y)\}$$

then, they define the local reachability density (lrd) that uses the reachability distance defined as

$$lrd_{MinPts}(x) = 1 \Big/ \frac{\sum_{y \in N_{MinPts(x)}} reach\text{-}dist_{MinPts}(x,y)}{|N_{MinPts}(x)|}.$$

To define a density, two parameters are necessary:

- The parameter MinPts, designing the minimum number of points inside.
- A parameter for the volume.

With these two parameters, Breunig et al determined a density threshold in order to make sure the clustering algorithm is working. MinPts has been kept as a parameter and use the values of reach-dist$_{\text{MinPts}}$(x,y), y ∈ N$_{\text{minPts}}$(p) to have a measure of volume, so that the density in the neighborhood of a data point can be computed, in order to compare with his own density.

Finally, the local outlier factor can be defined as

$$LOF_{MinPts}(x) = \frac{\sum_{y \in N_{MinPts}(x)} \frac{lrd_{MinPts}(y)}{lrd_{MinPts}(x)}}{|N_{MinPts}(x)|}.$$

The LOF compares the density of p with his neighbors. When the lrd of x is really low, the lrd of his MinPts's neighbors is high, so is the local outlier factor, meaning that the more high the LOF is, the more the data point is likely to be an outlier. According to Wang et al [6] (but made by Schubert et al (2014) [17]), a simplifiedLOF also exists. The computation of the density is different: instead of using reachability-distance, the k-distance of p can be used

$$dens(x) = \frac{1}{k - distance(x)}$$

The SimplifiedLOF, more than simplifying the LOF has improved the performance of the LOF. However it didn't change the complexity of the algorithm. Other similar measures have been put in place as improvements of the Local Outlier Factor.

Improvements of the Local Outlier Factor

a) Local Outlier Probabilities

Kriegel et al (2014) [18] proposed another outlier detection technique similar to the LOF: providing an outlier score but this time mixed with probabilities and statistics. To build this technique, they introduced a probabilistic distance of a data point x ∈ D to a context set S. The context set is included by D. The probabilistic distance is referred to as pdist(x,S) and has the following property

$$\forall s \in S, \quad P[d(x,s) \leq pdist(x,S)] \geq \varphi$$

the property means that with a sphere with a radius of pdist around a data point x include each points inside the context set with a probability called φ. But instead of using φ, they use λ, defined as

$$\lambda = \sqrt{2} * erf^{-1}(\varphi)$$

where the erf corresponds to the Gaussian error function (called also simply error function). This transformation permits to have a statistical definition of an outlier: indeed, outliers can be defined as data points that deviate more than λ times the standard deviation (called σ) from the mean of the data distribution. The standard deviation in the paper of Kriegel et al (2014) is an approximation, assuming that x is the center of the context set and that the set of distances between s and o is approximately a half-Gaussian

$$\sigma(x,S) = \sqrt{\frac{\sum_{s \in S} d(x,s)^2}{|S|}}$$

they assume S to be approximately normally distributed around the data point x. To obtain the data points inside S, a KNN model is used, so that the assumption is acceptable. To estimate the density around a data point x, they defined the probabilistic set distance of x to S with a significance λ

$$pdist(\lambda, x, S) = \lambda * \sigma(x, S)$$

and then, they defined the PLOF (Probabilistic Local Outlier Factor) that computes the ratio of the estimation of the density around x with probabilistic set distance and the expected values of the estimation of densities around all the data points that are inside the context set. The PLOF of a data point x with respect to a significance λ and the context set S, can be defined as

$$PLOF_{\lambda,S}(x) = \frac{pdist(\lambda, x, S(x))}{E_{s\ \in S(x)}[pdist(\lambda, s, S(s))]} - 1$$

there is also a nPLOF, that is an aggregate value of PLOF in order to normalize. The nPLOF is then defined as

.

$$nPLOF(x) = \lambda * \sqrt{E[(PLOF)^2]}$$

and finally, the Local Outlier Probability (LoOP) can be defined. It represents the probability for a data point to be an outlier

$$LoOP_S(x) = \max\{0, \mathrm{erf}(\frac{PLOF_{\lambda,S}(x)}{nPLOF * \sqrt{2}})\}$$

if the LoOP is close to 0, then it means the data point is in a high density region while when the LoOP is close to 1 means that the data point is a density-based outlier. Since it is a probability, the outlier score is here comparable with, for instance, the outlier score of each data point of a dataset.

b) Connective-based Outlier Factor (COF)

As improvement of the Local Outlier Factor, there is the method introduced by Tang et al [19] called the COF (Connective-based Outlier Factor). It uses any chaining distance (such as the shortest path) in order to estimate the density, instead of using the euclidean distance to select the KNN as the Local Outlier Factor. They differentiate « low density » and « isolativity », defined as the degree of a data point's connectivity to others. So, a data point can be in high density and being isolated.

c) Other density-based outlier methods

Wang et al (2019) [6] reviewed several other density-based methods for outlier detection such as the Local Correlation Integral (LOCI) and the multi-granularity deviation factor (MDEF), by Papadimitriou et al (2003) [20]. It solves the problem of multi-granularity that the LOF and COF has. When a data point deviates more than 3 times the standard deviation of the MDEF's neighbors, it is considered as an outlier. They also did another algorithm called aLOCI (or approximated LOCI). It approximates the counting of the neighborhood of a data point.

Another density-based method, proposed by Ren et al (2004) [21], called Relative Density Factor (RDF). It uses a vertical data model to detect outliers. Data points with a high RDF value are considered to be outliers. Another technique is the INFLO (Influenced Outlierness), made by Jen et al [22]. INFLO uses symmetric relationship between the nearest neighbors and the reverse nearest neighbors. More the INFLO is high for a data point; more the data point is likely to be an outlier.

However, all these variations of the LOF have difficulties when it comes to high dimensional data (Wang et al, 2019) [6]. Other measures have been put in place. For instance, the High Contrast Subspace method (HiCS), by Keller et al (2012) [23], the Global-Local Outlier Score from Hierarchies (GLOSH), by Campello et al (2015) [24].

Advantages and Disadvantages

The main advantage, given by Wang et al, is that LOF and all his variations are said non-parametric, meaning that there is not assumption on the distribution of the data. They were the baseline to build other outlier detection algorithms. Modern density methods outperform distance-based methods.

However, most of them are sensitive of the parameter settings: such as the k of the nearest neighbors, or even the λ of the Local Outlier Probabilities. Also, the INFLO has poor performance in the case of continuous flow of data (in other words, data stream), because there is no update of his measure. As mentioned before, it does not work well for density-based techniques when data are high-dimensional.

3.4 Isolation Forest

The isolation forest (or iForest) is an unsupervised outlier detection method introduced by Liu et al (2008) [25]. The idea is to isolate outliers with isolation trees. In their paper, they defined isolation as « separating an instance from the rest of the instances ». The idea is that outliers should be easier to isolate than normal data points.

The definition of the Isolation tree (or iTree) is the following: Assuming that T is a node of an Isolation tree, he can be an external-node (not a parent of other nodes) or an internal-node with a test and parent of two nodes (called T_l and T_r). Then an isolation tree has a structure similar to the BST (Binary Search Tree, it is a data structure where all nodes possess a key, where a parent's node has a higher key than the left child and less high than the right child**).** A test divides data points between the 2 children, given a feature a and a split value p. The isolation tree is built recursively like this until:

- The tree reached a certain height limit ;
- The number of data points left is equal to 1 ;
- The data points left inside D have the same values.

The outlier detection is made by a ranking of the degree of outlierness. The isolation forest sort the data points with the path length and serves as an outlier score. The path length h(x) is the number of edges that x traverses in the iTree from the root until x arrives in an external node. Since the structure of an iTree is similar to the BST, the estimation of the average path length should be the same than the one for an unsuccessful BST (the average path of an unsuccessful BST is given by Preiss (1999), mentioned in the article of Liu et al

$$c(n) = 2H(n-1) - \frac{2n-2}{n}$$

where H(i) is called the harmonic number and can be estimated by by ln(i) + γ (γ is the Euler's constant). c(n) is used afterwards to normalize h(x). The outlier score s(x,n) is given by

$$s(x,n) = 2^{\frac{-E(h(x))}{c(n)}}$$

where E(h(x)) is the average of h(x), computed with the path length on a number of isolation trees. Given the equation of the outlier score, different values can be found given E(h(x))

$When\ E\big(h(x)\big) \to c(n), s \ \to \frac{1}{2}$

$When\ E\big(h(x)\big) \to 0, s \ \to 1$

$When\ E\big(h(x)\big) \to n-1, s \ \to 0$

Where s ∈]0,1] and h ∈]0,n-1]. Assessments have then been made by:

- If a data point returns s close to 1, it is considered to be an outlier.
- If a data point is smaller than 0.5, it is considered to be an inlier.
- If all data points are close to s equal to 0.5, then no outlier is detected and no data point isolated.

For this technique there are two different stages: the training stage, where the isolation trees are built with sub-sampling (so that outliers are easier to isolate) and the evaluating stage, where data points traverses the trees and the outlier score is computed.

a) Training Stage

Isolation trees are built with partitioning the dataset, until the data points are isolated or that the height limit has been reached. The tree height limit is automatically set by the size of the sub-sampling ψ. The relationship between them is $l = \text{ceiling}(\log_2 \psi)$, where l is the average height of trees. The number of trees t and the sub-sampling size ψ are the inputs of this stage while the output is the Isolation Forest. Ψ controls the size of the training set of a tree, while t controls the size of the ensemble.

Liu et al found that if ψ increases to a desired value, the Isolation Forest's detection is reliable and increasing more the size of the training set would not increase the information gain. This desired value found is equal to 256 and provides enough details to detect outliers rather correctly. For the number of tree, they found that path length was already converging with a number of trees set to t = 100.

b) Evaluation Stage

In the Evaluation stage, the outlier score s is computed for each data point. The path length h(x) is a counter of the number of edges the data point x traverses. When this value is obtained by all the isolation trees, the outlier score is computed for each point, with formula described above. Points are also sorted by s, so that the top m outliers can be found.

Another type of ensemble-based technique: The DSCO

As another type of ensemble-based technique, Zhao and Hryniewicki (2018) [26], called the Dynamic Combination of Detection Scores for Outlier Ensembles (DSCO). It is an unsupervised approach for outlier detection. The main idea is to make a choice and a combination of the outlier scores in a local region, in the absence of ground truth (checking the accuracy of the results in the real world).

The design of the algorithm is the following: it has two stages. The first one is the Generation, it build the detectors and are all fit on all the data points, while the second one is the Combination, that selects the best detectors for a certain region defined by the data points used for test, and this detector is then used to predict the outlier score.

Advantages & disadvantages

Compared to the LOF and the KNN, the ensemble-based technique does not necessarily (in particular the Isolation Forest) use distance or density measures to detect outliers. It is a computational cost that the Isolation Forest does not have.

Furthermore, it has a linear time complexity and low memory requirement. In the worst case, if we assume that all data points are distinct, each of them would be isolated in a node, meaning that number of internal nodes in that case would be n-1, while number of external nodes is equal to n. So, the total number of nodes would be equal to 2n-1. Then, the memory requirement is linear, growing with the number of data points. In addition, the Isolation Forest has the capacity to scale up. It can handle large datasets but also high dimensional data, even if there is a lot of features that are irrelevant.

According to Zhao and Hryniewicki [26], their algorithm has his own advantages compared to standard outliers detection techniques. The DSCO outperforms techniques that are static on the majority of datasets and also have more precision. It has a great extensibility and compatible with base detectors, with the Local Outlier Factor and k Nearest Neighbors as examples.

In general, according to Wang et al [6], ensemble-based methods are more stable and give better models in terms of prediction, and are suitable for high dimensional data. They are also robust for scenarios with data stream. However, that kind of techniques is poorly developed: it implies that it's difficult to evaluate the features of the ensemble or even choose the right detector.

3.5 Gaussian Mixture Model

The Gaussian Mixture Model (or GMM) is used as an unsupervised outlier detection technique made by Yang et al (2009) [27]. This technique is categorized into the group of approaches called statistical or distribution-based approach, because data points are modeled by using a certain data distribution. In this case, it is a mix of Gaussians. The idea is to first apply a globally optimal Expectation Maximization algorithm (EM) in order to fit the GMM to a certain dataset. A Gaussian is centered for each data points and the estimated mixture proportions can be used as probabilities of being the center of a cluster. A data point with a low probability is more likely to be an outlier.

For a set of data points D, a Gaussian Mixture Model for clustering tries to maximize the scaled log-likelyhood function

$$l(\pi_{1:v}, \mu_{1:v}, \lambda; D) = \frac{1}{n} \sum_{i=1}^{n} \log[\sum_{j=1}^{v} \pi_j \, p(x_i | \mu_j, \lambda)]$$

where v corresponds to the number of mixture components, $\pi_j = p(\omega_j|\lambda)$ is the strength of the j^th^

component ω_j, the sum of all π_j is equal to 1. The probability $p(x_i|\mu_j,\lambda)$ represents one Gaussian, λ is a vector of parameters and μ is the mean vector (unknown), it is estimated with the others parameters via the globally optimal Expectation Maximization algorithm, made by Dempster et al (1977) [28]. This algorithm is iterative and contains two different steps:

- E-step (or Estimation step): The estimation of the missing variables (here, the posterior probability).
- M-step (or Maximization step): Maximize the parameters of the Gaussian model with the dataset.

Since the aim is to detect outliers, they assumed that each data point was a center of a cluster, meaning that they will have as much models as they have data points (m = n). It also means that the mean vector equals the dataset. The π_j, the mixture proportion represents the probability of the point x_j to be a center of a cluster. The log-likelyhood function can then be simplified

$$l(r_{1:v}; D) = \frac{1}{n}\sum_{i=1}^{n} \log[\sum_{j=1}^{n} \pi_j\, p(x_i|x_j,\lambda)]$$

so, it simplifies the EM algorithm where at the tth iteration, the vector of parameters $\lambda_t = \{\pi_1(t),...,\pi_m(t)\}$

For the E-step, the algorithm computes for each cluster i= 1,...,n and also for each data points k = 1,...,n the probability

$$p(x_i|x_k,\lambda_t) = \frac{p(x_k|x_i,\lambda_t)p(x_i|\lambda_t)}{p(x_k|\lambda_t)} = \frac{p(x_k|x_i)\pi_i(t)}{\sum_{j=1}^{n} p(x_k|x_j)\pi_j(t)}$$

with $p(x_i|\lambda_t) = p(\omega_i|\lambda) = \pi_i$.

The M-step of the algorithm simply updates the mixture proportions

$$\pi_i\,(t+1) = \frac{1}{n}\sum_{k=1}^{n} p(x_i|x_k,\lambda_t) = \frac{1}{n}\sum_{k=1}^{n} \frac{p(x_k|x_i)\pi_i(t)}{\sum_{j=1}^{n} p(x_k|x_j)\pi_j(t)}$$

where $p(x_k|x_i)$ represents a Gaussian, meaning that

$$p(x_k|x_i) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(d(x_j,x_k))^2}{2\sigma^2}}$$

With $d(x_j,x_k)$ the distance between the two data points x_j et x_k. And σ is the parameter to estimate.

After, they denote the probability p(x_k,x_j) as s_{kj}. All the elements s_{kj} give the matrix S, the affinity matrix of the graph build with the dataset. Each element of the matrix represents the strength of the connection between the data points. For example, s_{kj} is the strength between x_j and x_k. Putting the element s_{kj} in the equation above gives

$$\pi_i\ (t+1) = \frac{1}{n}\sum_{k=1}^{n} \frac{s_{kj}\ \pi_i(t)}{\sum_{j=1}^{n} s_{kj}\pi_j(t)} = \frac{1}{n}\sum_{k=1}^{n} \frac{s_{kj}\ \pi_i(t)}{z_k(t)}$$

where z_k(t) defined as

$$z_k(t) = \sum_{k=1}^{n} s_{kj}\ \pi_i(t).$$

z_k(t) represents how all other the other data points are influencing x_k at the t^{th} iteration.· Moreover, $s_{kj}\pi_j$(t) represents how the data point x_k is influenced by the data point x_j with s_{kj} representing the strength of the connection between them, and π_j(t) the importance of the data point x_j. This, when it converges to the final iteration (equal to h) gives the outlier factor of the Gaussian Mixture Model technique for outlier detection

$$F_k = z_k(t_h) = \sum_{k=1}^{n} s_{kj}\ \pi_j(t_h)$$

when F_k tends to be small, the data point x_k is more likely to be an outlier, because the connections between him and the other data points are very weak. When F_k tends to be high, then the data point is more likely to be an inlier.

Variation of the GMM technique

A variation of the technique have been introduced by Tang et al (2015) [29], using the GMM and the subspace learning. Rather than learning into the entire space, subspace learning would take into consideration only a part of it. Most outliers are considered to be rare neighborhood activities in a subspace. It is widely used for outlier detection with high-dimensional data. (Wang et al (2019) [6]).

The technique mentioned uses the Locality Preserving Projections (LPP), introduced by He and Niyogi (2003) [30]. The LPP can be considered as the Principal Component Analysis (PCA), where the idea is to reduce dimensionality of data points by reducing as much as much possible information.

The LPP is a linear approximation dimensionality reduction algorithm. This algorithm builds a

(Laplacian) graph that incorporates the neighborhood information. Then, a transformation matrix is computed, so that the data points are put in a subspace. The algorithm preserves as much as possible that neighborhood information.

The Kernel Density Estimation Method

Latecki et al introduced a non-parametric statistical-based approach with kernel Density functions (2007) [31]. It uses a kernel (Gaussian) function to estimate the probability density function of a dataset. The idea is to compare density of a data point compared to his neighbors, such as a density-based method is trying to do. The distribution density is estimated by using the following formula, given n data points with dimensionality dim

$$p(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{b(x_i)^{dim}}\ K(\frac{x - x_i}{b(x_i)})$$

where b(x_i) is called bandwidth function and is implemented at the data points. The simplest bandwidth function is when the bandwidth is constant. K is the Gaussian kernel function, defined as

$$K(x) = \frac{1}{(2\pi)^{dim}}\ e^{-\frac{||x||^2}{2}}$$

where $||x||^2$ corresponds to the norm of the data points. The Gaussian kernel function has a zero mean and a standard deviation equal to 1. In other words, the kernel function is a standard Gaussian.

Their experimental results showed that their method had a better performance than the classical density-methods such as the Local Outlier Factor or Local Correlation Integral.

Advantages and disadvantages

According to Wang et al [6], the statistical-based (or distribution-based) methods have a fast evaluation process when the model is built. They fit generally well to real datasets and they are easier to implement.

However, for parametric models (as the Gaussian Mixture Model), the quality of the model is almost not reliable, because of the assumptions on the data distribution. Because models built by this kind of method applies only to an univariate feature space (meaning that there is only one feature for each data point), they are not applicable with high dimensional data. It is particularly true for the statistical tests of the first section of this chapter.

3.6 DBSCAN for Outlier Detection

DBSCAN is an unsupervised approach introduced by Ester et al (1996) [32], used first to build clusters to make different classes, but can also be used for outlier detection, just as Çelik et al (2011) [33] did to detect « extreme temperatures » with DBSCAN on a time series data. The idea of the algorithm is to make cluster based on a ε-neighborhood of a data point x. The ε-neighborhood should contain a minimal number of data points. All points reachable by the ε-neighborhood form one cluster. For that reason, there are points that would not be included in one cluster. These points can be considered as outliers.

The ε-neighborhood of a point x of the DBSCAN technique is denoted N_ε(x) and defined as

$$N_\varepsilon(x) = \{y \ \in D \ | d(x,y) \leq \ \varepsilon\}$$

where ε can be seen as the radius of a sphere, and d(x,y) is a distance function computing the distance between the data point x and y. The shape of the neighborhood depends of this distance function. For example, with the Manhattan distance, the shape of the neighborhood, in a two dimensional space, would be rectangular.

As a naive approach, Ester et al [32] considered that the clusters built should have at least a minimal number of points in the ε-neighborhood, called MinPts. This approaches fails, because there is two different types of data points inside a cluster:

- The data points inside the cluster, called core points.
- The data points that are on the border of the cluster, called border points.

For a border point, the ε-neighborhood would contain less data points than a core point. Then, the minimum number of point of a ε-neighborhood should be relatively low, so that border points are not considered as outliers. Then, they required that for each point x in the cluster C, there is at least one point y (belonging in the cluster C) where his ε-neighborhood includes x and N_ε(x) contains at least MinPts points. It is the definition of a point that is directly density-reachable from another.

Mathematically speaking, the point x is directly density-reachable from a point y with respect to the radius ε and a number of minimum points, MinPts, if:

- $x \in N_\varepsilon(y)$;
- $|N_\varepsilon(y)| \geq$ MinPts.

Where $|N_\varepsilon(y)|$ is the number of points inside the ε-neighborhood. The definition of directly density-reachable is symmetric for 2 core points, meaning that if we assume that x and y are core points, then x is directly density-reachable of y and y is directly density-reachable of x. But it's not the case with a core point and a border point.

The density-reachable definition has also been given : the point x is density-reachable from a point y with respect to the radius ε and a number of minimum points, MinPts if there is a chain of points x_1,...,x_n with x_1 = y and x_n = x such that x_{i+1} is directly density-reachable from x_i. By definition, The relation is transitive but still not symmetric, due to the core point condition not respected for the border point in the directly density-reachable definition.

They then defined the density-connectivity that is, compared to density-reachable relation, symmetric. A point x is said density-connected to a point y with respect to the radius ε and a number of minimum points, MinPts if there is a point z such that w and y are both density-reachable with z, with respect to the radius ε and the number of minimum points MinPts. For the points that are density-reachable, the relation is reflexive.

Finally, they end with the definition of cluster and noise that will be the outliers. A cluster with respect to the radius ε and a number of minimum points, MinPts is a non-empty set (also a subset of D) that satisfies the following conditions:

- ∀ x,y : if x ∈ C and y is density-reachable from x with respect to ε and MinPts, then y ∈ C ;
- ∀ x,y ∈ C : x is density-connected to y with respect to ε and MinPts.

The noise (but in our case, the outliers) is the set of points that does not belong to any cluster made by the algorithm

.

$$Outliers = \{p \ \in D | \forall i : x \ \notin \ C_i\}$$

Other techniques based on clustering

Wang et al [6] classified different clustering-based techniques into subgroups:

- The Partitioning Clustering Methods, or the distance-based clustering algorithms. The number of clusters made by these algorithms is generally initially given or randomly chosen ;
- The Hierarchical Clustering Methods, where the algorithm partitions the dataset into groups with different levels. The structure is similar to a tree. It requires a maximum number of clusters ;
- The Density-based Clustering Methods, such as DBSCAN
- The Grid-based Clustering Methods ;
- The Clustering Methods for High-Dimensional Data.

The K-means are an example of Partitioning Clustering Method. This technique is introduced by MacQueen (1967) [34]. The idea is to find k different clusters, where the center of these clusters is called centroids. They are generally artificial points, chosen by minimizing the error. The data point is assigned to the closest centroid. Then, the center of the clusters is modified. The algorithm is iterative until the centroids do not move anymore.

As an example of Hierarchical Clustering Method, Guba et al (1997) [35] introduced CURE (Clustering Using Representatives). The method is in between the centroid-based and the all-point extremes. This algorithm can be used to discover non-spherical clusters. It uses random sampling, and then, random samples are partially clustered. The outliers of the partition are then eliminated. When it's done, clustered data in each partition is then clustered again to generate the final clusters.

Advantages and Disadvantages

Clustering methods are very robust and works for many data types. Partitioning Clustering Methods are said to be to be simple and scalable. When the clusters are built, it is possible to add new data points to test if they are considered to be outliers or not.

However, in the clustering techniques, there is no degree of outlierness or outlier score for data points: points are outliers or they aren't. Most of them rely on the user who needs to give the number of clusters as input. An arbitrary shape of the cluster could also be problematic if it doesn't correspond to reality. Clusters assumed in data stream would be also problematic since data are changing over time.

As a conclusion, the different techniques are in different families of techniques. However, they can use the same concepts. For instance, the LOF uses also the k-nearest neighbors; DBSCAN uses a minimum number of points inside a neighborhood of a point. The LOF is one of the first outlier detection technique (and has a lot of improvements), with the Isolation Forest and the Gaussian Mixture Model are the most recent described (other techniques briefly mentioned excluded). The performance (measured with the evaluation methods discussed in the following chapter) may be better for the most recent methods (this will be verified in the chapter 5).

4. Evaluation Methods

There are much more different outlier detection techniques than the ones described in the previous chapter. Researchers tried to improve the root methods and said their methods outperform them without any further analysis. According to the application domain, a certain OD technique could perform better than another. That is why Evaluation Methods are important: to have the possibility to compare the different models. The evaluation can be made on different criteria: The performance (precision), the robustness of the models or even the time complexity (scalability) or memory usage. That's what Domingues et al (2018) [36] did in their paper. They performed different outlier detection methods such as the Local Outlier Factor, the Gaussian Mixture Model or the Isolation Forest. They conclude the iForest was a good method to detect outliers with an excellent scalability and acceptable memory usage when the number of data points is large (to be more precise, the memory usage is still acceptable with a dataset of one million of data points.

The performance of an outlier detection technique corresponds to the precision of detection the outliers. The aim of an outlier detector is to detect data points that are, depending on the application, considered as outliers. However, the technique can make mistakes and detect inliers as outliers (rather false outliers). To be efficient, outlier detection should detect as much as true outliers as possible, but also minimizing the number of false outliers detected.

The robustness is the property that one model is effective enough when it is tested in another dataset, similar but independent of the first one. For example, the performance of the technique would be evaluated for different datasets with, for example, an increasing number of features (impact of dimensionality), or an increasing number of data points (impact of dataset).

The time complexity (scalability) and space complexity (memory usage) depends of the outlier detection method. Generally, it is a function of the number of feature or the size of the dataset. Time and space complexity are increasing functions, but the different methods does not increase the same manner. There are techniques would be less scalable because of a higher growth in the complexity function. It is the same for the space complexity.

To evaluate an outlier detection method, there is need to evaluate on different type of datasets : the synthetic datasets, that contains data points made artificially, created according to defined constraints and conditions and the real-world datasets that can be obtain from publicly available databases. Synthetic datasets are less complex than real-world datasets and shows a better validity for outlier detection method. That's why there is a necessity to evaluate the outlier detection methods on these 2 types of datasets.

In this thesis, only a focus on the performance will be done. Two different methods will be presented, the Precision (section 1) and the Receiver Operating Characteric (ROC) (section 2), associated withe the Area Under the Curve or AUC (section 3).

4.1 Precision

The Precision is defined as follows in Wang et al's paper [6]

$$Precision = \frac{o}{t}$$

where o is the number of outliers correctly identified while t is the total number of outliers. The problem with just this evaluation method is that it takes only into account the number of true outliers identified by the method, but not the number of inliers that are considered to be outliers. The evaluation method is incomplete. It is also difficult to know t in advance (or even estimate it) if the aim of model is to predict which points are outliers.

4.2 Receiver Operating Characteristic (ROC) curve

According to Brown and Davis (2006) [37], the Receiver Operating Characteristic curve was introduced during the World War II military radar operations, in order to detect friendly and hostile aircrafts, based on a signal. A hostile aircraft considered as a friendly one would have incurred to a huge loss, as well as for sending military support to a friendly aircraft. It has been used in numerous applications after, including in Machine Learning to evaluate performance of a binary classifier. For example, F. Tortorella (2000) [38] used the Receiver Operating Characteristic curve for outlier detection. It can be used as a performance method for outlier detection by putting all classes as only one class, the inliers, while the second class would correspond to the outliers. With multiple classes, it is enough to gather class as only one, the inliers, with the outliers as second class and do it for each class.

The idea is based on the two types of error: The Type I error and the Type II error. In hypothesis testing, the Type I error corresponds to the rejection of the null hypothesis while it could have been true. For example, in the medical science, a type I error would be to diagnose an illness to a healthy patient. The Type II error corresponds to the acceptation of the null hypothesis while it is actually wrong. In the medical science, it would be to diagnose no illness to a sick patient. The type I error is also called false positive while the Type II is also called false negative.

As said before, outlier detection technique would detect real outliers (True Negative or TN), but also detect outliers that are actually inliers (False Negative or FN). There would be also outliers that are not detected by the approach (False Positive or FP). Inliers considered as inliers are the True Positive (or TP). These four values forms together a table called confusion matrix.

The ROC curve is a curve using two parameters that are ratios based on the False Positive/Negative and True Positive/Negative: the True Positive Rate (TPR) and False Negative Rate (TNR). The TPR is defined as

$$True\ Positive\ Rate = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{True\ Positive}{Positive}$$

where the “Positive” corresponds to the number of inliers of the dataset. As for the False Negative Rate, it is defined as

$$False\ Positive\ Rate = \frac{False\ Positive}{False\ Positive + True\ Negative} = \frac{False\ Positive}{Negative}$$

where the “Negative” corresponds to the number of outliers in the dataset. These two rates are linked to each other

$$True\ Positive\ Rate = 1 - False\ Negative\ Rate$$

The ROC curve is defined on a two-dimensional space passing through the points (0,0) and (0,1), with the x-axis and y-axis, corresponding respectively to the False Positive Rate and the True Positive Rate. For each False Positive Rate and True Positive Rate come with a decision threshold. The decision to put a data point into the positive class (inlier) or into the negative class (outlier) is defined by C. D. Brown and H. T. Davis [37] as

$$Decision = \begin{cases} +\ if\ a\ \geq\ \vartheta \\ -\ if\ a <\ \vartheta \end{cases}$$

where ϑ is the decision threshold and x is a continuous variable.

One way to fit the ROC model is briefly described by Brown and Davis is to use a binomial ROC model, because for there are data points where the x distribution of the positive or the negative event is said near-normal or is normalizable by a monotonic transformation. The choice of that transformation will not affect the ROC curve. The parameters for the binormal ROC model are the distribution standard deviations σ^+ and σ^- and μ^+ and μ^- that are the distribution means. The ROC curve is then defined as

$$ROC(\vartheta) = \ \phi(a + b\ \phi^{-1}(\vartheta))$$

With a defined by the parameters of the binormal model

$$a = \frac{\widehat{\mu^{+}} - \widehat{\mu^{-}}}{\widehat{\sigma^{+}}}$$

And b defined as

$$b = \frac{\widehat{\sigma^{-}}}{\widehat{\sigma^{+}}}.$$

4.3 Area Under the ROC Curve

The Area Under the ROC Curve (or AUC) corresponds to the area under the curve in the two-dimensional space with the True Positive Rate and the False Negative Rate as axes. The AUC is frequently used as a summary measure. The perfect ROC curve corresponds to an AUC equal to one: the curve reaches (0,0), (0,1) and (1,1). It forms an angle on the upper left corner. The ROC curve for a « random classifier » corresponds to a straight line; the AUC is equal to 0.5. The AUC can be considered as a probabilistic measure. It is equal to the probability that the values x for a random of positive/negative points will be correctly classified. Then, the more the ROC curve is close to the upper left corner, more the model is efficient. More the curve is close to the lower right corner, less the model is performing .Also, if the AUC is close to 1, and then more the model is efficient since an AUC close to 1 means a ROC curve close to the upper left corner of the Two-dimensional space.

If the data is suitable with the binormal model mentioned above, the AUC is estimated as

$$AUC = \phi(\frac{\widehat{\mu_D} - \widehat{\mu_{\bar{D}}}}{\sqrt{\widehat{\sigma_D}^2 + \widehat{\sigma_{\bar{D}}}^2}})$$

with the parameters estimated of the ROC curve.

As a conclusion, the precision is not really a good performance method, but it can be a way to know the rate of real outliers detected if the total number of outlier is known in advance. The ROC curve is excellent for showing, the real measure of the performance is the Area Under the ROC curve, taking into account the number of false outliers detected and penalize for each of them.

5. Methodology & Experimental Results

In this chapter, the datasets used for the experimental results are described: a synthetic dataset made randomly with outliers made manually and the real-world dataset. These are discussed in section 1. Then, the methodology used to build outlier detection and performance methods are described in detail in the section 2, to finish with the results obtained by the implementation of these techniques in section 3.

5.1 Datasets

The five outlier detection techniques explained in the previous chapter are used to detect outliers on two different datasets: a synthetic dataset and a real-world dataset. The Real World dataset comes from the Outlier Detection DataSets (ODDS, http://odds.cs.stonybrook.edu/). They provide different type of datasets (multi-dimensional point datasets, time series graph datasets, time series point datasets (multi or univariate), Adversarial/Attack scenario and security datasets) from different domains that are especially made for outlier detection.

- The Synthetic dataset was made artificially. It contains 630 data points and 2 features. 600 were generated randomly, into a two-dimensional space [0,100] x [0,100]. As for the 30 others, they were added on the dataset as outliers. Their features have been set manually, to make sure that all the outliers are rather far enough compared to the inliers to be detected by the different techniques. **Figure 1** shows a scatter plot of the dataset and distinguish inliers and outliers.

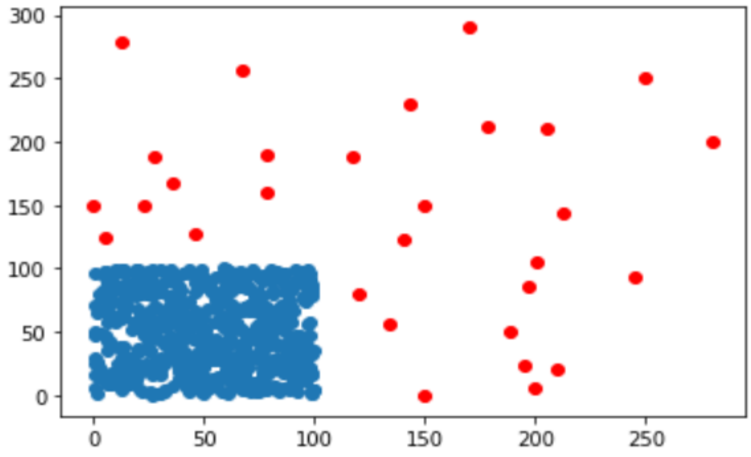


Figure 1: scatter plot of the synthetic dataset. Blue points are inliers and red are outliers.

- The Cardio (Cardiotocography) dataset: this dataset is a classification dataset containing measurements of the Fetal Heart Rate (FHR) and uterine contraction (UC) features on fetal cardiotocograph. They were classified by expert obstetricians where 3 classes for the fetal state are present: Normal, Suspect and Pathologic. In the original dataset, they were also classified with respect to a morphologic pattern. 2126 data points are on the original dataset but in the ODDS, they reduced the sample to 1831 points, the suspect class has been discarded. 176 outliers are present in the dataset. They form the Pathologic class. Each data points has 23 dimensions on the original dataset, but only 21 in the dataset used, that are the following:

1. LB: Fetal Heart Rate (beats per minute);
2. AC: The number of accelerations per second;
3. FM: The number of Fetal Movements per second;
4. UC: The number of Uterine Contraction per second;
5. DL: The number of Light Decelerations per second;
6. DS: The number of Severe Decelerations per second;
7. DP: The number of Prolongued Decelerations per second;
8. ASTV: The percentage of time with abnormal short term variability;
9. MSTV: The mean value of short term variability;
10. ALTV: The percentage of time but for long term variability;
11. MLTV: The mean value of long term variability;

12-21. Widrth, Min, Max, Nmax, Nzeros, Mode, Mean, Median, Variance, and Tendency: All these features are statistics on the fetal histogram;

22. CLASS: Morphologic pattern code;
23. NSP: Fetal state class code.

5.2 Methodology

The 5 outlier detection techniques have been implemented by using the Python language. The Scikit-learn package has been used for the 5 techniques as well as for the ROC curve.

a) Neighborhood and KNN's :

To use the KNN as an outlier detection technique, the model was created and then fit with both datasets. The mean of the k-distances for each point are then computed. The mean distances for each point are sorted and the distances above the 95th percentile are considered as outliers by the algorithm. For both datasets, the algorithm has run 9 times, with the parameter k (the nearest neighbors) ranging from 1 to 9.

b) Local Outlier Factor

The model for the Local Outlier Factor was first built with a parameter k, designing the nearest neighbors and a contamination rate, and then predicted with both datasets. The contamination rate corresponds to the rate of outliers inside the dataset. It was arbitrarily set to 5%, rather than use the real rate of both datasets, because it would change for each dataset, and a model cannot detect new outliers by knowing the contamination rate in advance. The outliers are the data points where the prediction is equal to -1. Same as the previous technique, the algorithm has run 9 times, with the parameter k ranging from 1 to 9.

c) Isolation Forest

First, the Isolation Forest is created with a certain number of estimators (corresponding to the number of trees), a contamination rate equal to 5%, same as the Local Outlier Factor, in order to make a better comparison. The model has also a « behaviour » variable, set on « new », to avoid a potential future warning. After the model fit with the dataset, outliers are predicted by the isolation forest and the outliers detected by the algorithm has a result of -1, just as the Local Outlier Factor. The algorithm has run 11 times, with the number of tree varying from 100 to 200, with a step of 10.

d) Gaussian Mixture Model

After being built and fit (with a certain number of components, corresponding to the number of Gaussian into the mixture, all the other parameters have been set by default) with the datasets, the posteriors are estimated for each data point and each component. The data points that have less than 75% for each component are considered to be outliers. The threshold was set to 75% in order to have outliers detected (see the section Results for more details). The algorithm has run 9 times with the number of components ranging from 2 to 10. Only component gives a posterior equal to 1 for each data point.

e) DBSCAN

The DBSCAN model has been built (With the parameter epsilon, set to 7 for the synthetic dataset and 2.75 for the Cardio Dataset, the epsilon set to 2.75 for the synthetic dataset would detect all points as outliers, while in the other side, set to 7 for the Cardio Dataset would result to no outlier detection. The other parameter is the minimum number of points to be considered as a core point) then fit, predict with both datasets. The outliers are the data points where the predictio n has a result of -1.

f) Evaluation Method

For each technique, the number of true outliers and false outliers is computed, based on the index (containing the index of the outliers detected by a technique) extracted of each model. To count the true outliers, an array of target (with a length equals the number of outliers detected) is initialized to zero. When the observed target class is equal to 1 and corresponds with the outlier index, the value 1 is added in the array for the corresponding index. When it's done for each point detected as an outlier, counting all points where the target is equal to 1 corresponds to the number of true outliers. The false outliers are simply the other points in the target array. In other words, it is the length of the array minus the number of true outliers.

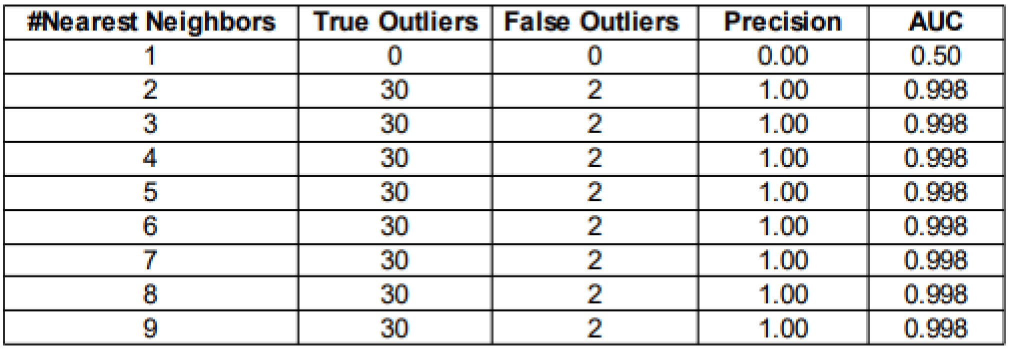
The precision is computed with the number of true outliers and the number of outliers present in the dataset. The True Positive Rate, the False Positive Rate and the Thresholds has been built with the observed target and the target scores. The AUC is computed with the TPR and FPR found.

5.3 Results

To have a better view for the results, a table for each technique and for both dataset including the variable parameter, the true outliers, the false outliers, the precision and the AUC. The results for the Precision and the AUC inside the tables rounded two digits after the decimal point, except when the difference was really small: it was rounded three digits after the decimal point. A figure represents the plot of the mean distances for each data points with both datasets, with the presence of the 95th percentile as threshold. Two others figures (one for each dataset) represent the ROC curve with the models with the parameter that gives the best performance for each outlier detection technique.

a) Neighborhood and KNN's

Table 1 represents the results for the synthetic dataset, and **Figure 2** is a plot of the mean distances computed by the KNN with k equal to 2, while **Table 2** represents the results for the Cardio dataset and **Figure 3** is the plot of the distance for k equal to 9.



#Nearest Neighbors	True Outliers	False Outliers	Precision	AUC
1	0	0	0.00	0.50
2	30	2	1.00	0.998
3	30	2	1.00	0.998
4	30	2	1.00	0.998
5	30	2	1.00	0.998
6	30	2	1.00	0.998
7	30	2	1.00	0.998
8	30	2	1.00	0.998
9	30	2	1.00	0.998

Table 1: results of the KNN on the Synthetic Dataset

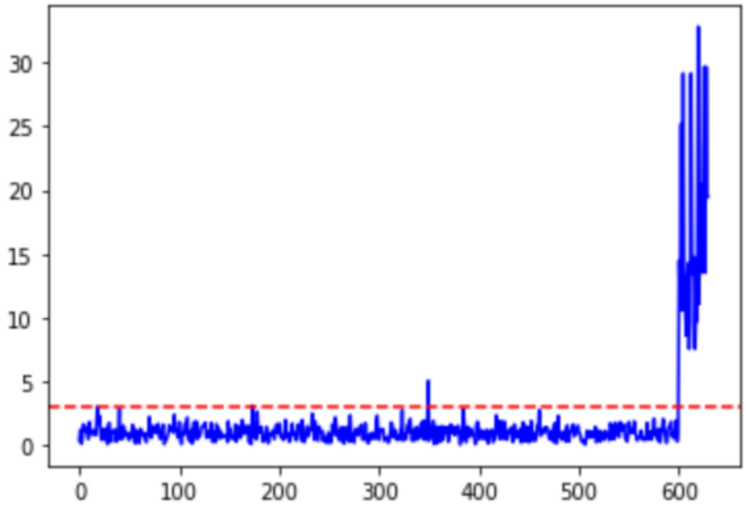
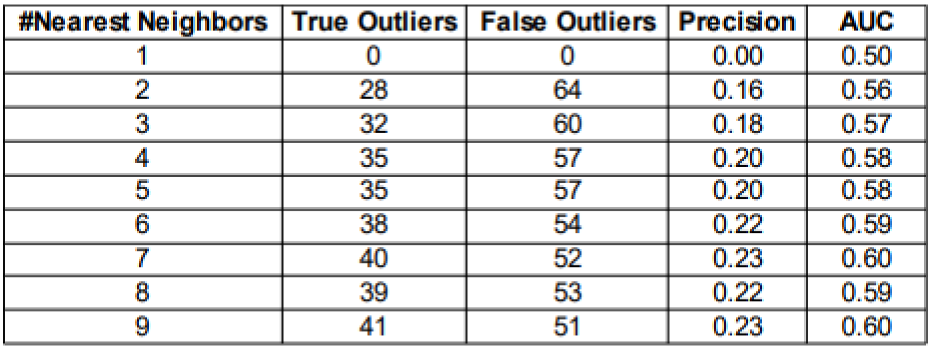


Figure 2: plot of the k-distances for each data points of the Synthetic Dataset (k = 2)

Synthetic Dataset: The table 1 shows that all outliers have been correctly detected for a number of nearest neighbors between 2 and 9. All the outliers are correctly identified due to to the large distance between the outliers and the inliers, while the distances between the inliers are really small. However, 2 inliers are detected as outliers by the algorithm, due to the fact there was 32 data points greater than the 95th percentile of the mean distances. The best model chosen is then when the number of neighbors is equal to 2: increase the number of nearest neighbors does not impact the performance but would increase the time and space complexity.



#Nearest Neighbors	True Outliers	False Outliers	Precision	AUC
1	0	0	0.00	0.50
2	28	64	0.16	0.56
3	32	60	0.18	0.57
4	35	57	0.20	0.58
5	35	57	0.20	0.58
6	38	54	0.22	0.59
7	40	52	0.23	0.60
8	39	53	0.22	0.59
9	41	51	0.23	0.60

Table 2: results of the KNN on the Cardio Dataset

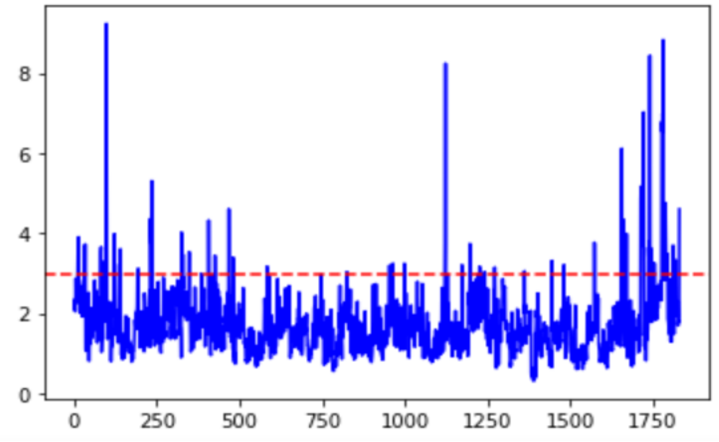
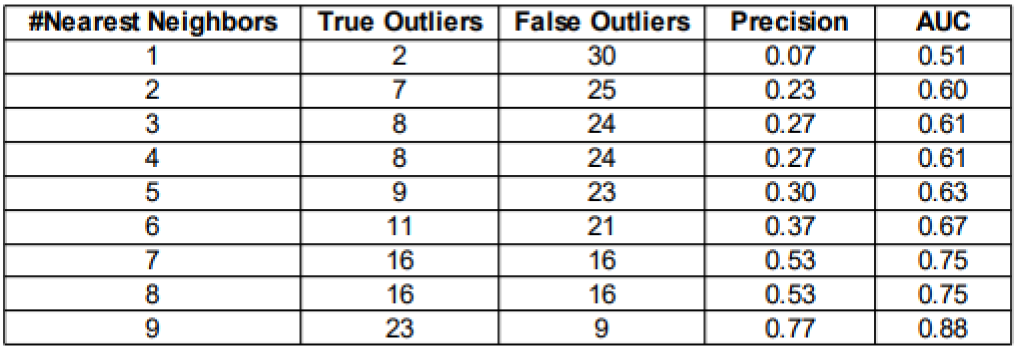


Figure 3: plot of the k-distances for each data points of the real-world dataset (k = 9)

Cardio Dataset: The table 2 shows an increase of the performance with the real-world dataset. As opposed to the performance with the synthetic dataset, it does not detect all the outliers and so, has less performance, as expected. The number of true outliers is increasing while the number of false outliers, except for k = 8. The Precision increases too with the AUC, because each model detects the same number of outliers. Also, even if the model would detect only true outliers, all the true outliers would not be detected, because the number of outliers forms more than 5% of the dataset.

b) The Local Outlier Factor

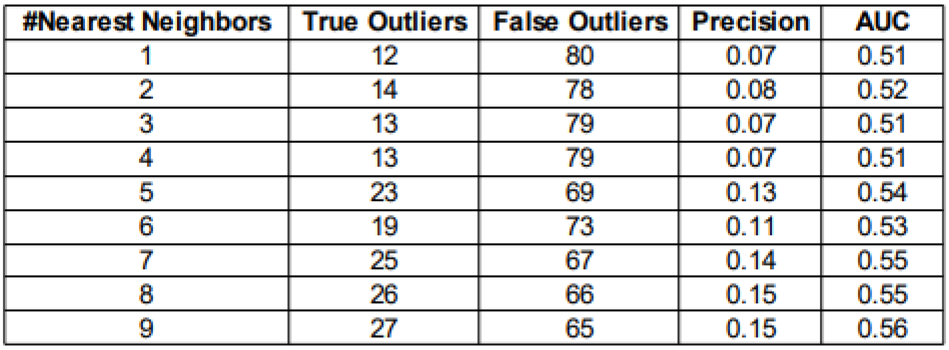
Table 3 give the results for a synthetic dataset while **Table 4** for the Cardio dataset,for the Local Outlier Factor, for neighbors varying from 1 to 9.



#Nearest Neighbors	True Outliers	False Outliers	Precision	AUC
1	2	30	0.07	0.51
2	7	25	0.23	0.60
3	8	24	0.27	0.61
4	8	24	0.27	0.61
5	9	23	0.30	0.63
6	11	21	0.37	0.67
7	16	16	0.53	0.75
8	16	16	0.53	0.75
9	23	9	0.77	0.88

Table 3: results of the LOF on the Synthetic Dataset

Synthetic Dataset: The different models do not detect immediately the 30 outliers of the datasets. It actually doesn't detect them all. However, the number of true outliers increases with the number of nearest neighbors. Since the number of outliers detected is the same for each model due to the contamination rate, the number of false outliers decreases, making increase the AUC. The results show that KNN method performs better than the LOF. The best model corresponds to the model with k equal to 9, with a higher AUC.



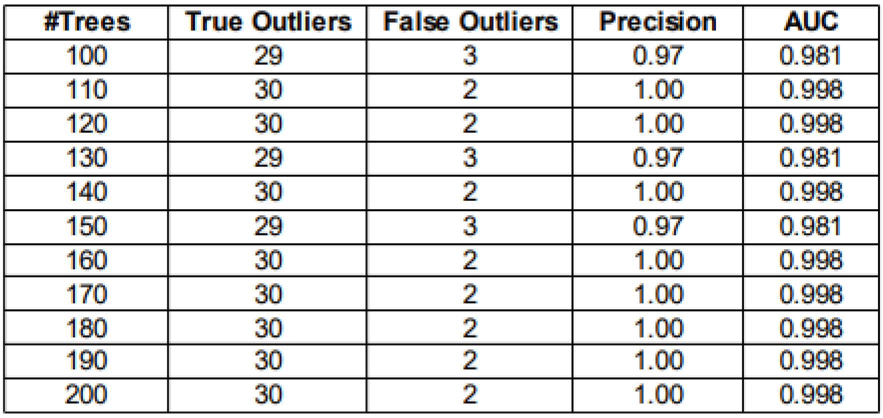
#Nearest Neighbors	True Outliers	False Outliers	Precision	AUC
1	12	80	0.07	0.51
2	14	78	0.08	0.52
3	13	79	0.07	0.51
4	13	79	0.07	0.51
5	23	69	0.13	0.54
6	19	73	0.11	0.53
7	25	67	0.14	0.55
8	26	66	0.15	0.55
9	27	65	0.15	0.56

Table 4: results of the LOF on the Cardio Dataset

Cardio Dataset: The results are rather similar than the Synthetic Dataset: The LOF is less efficient (the method detects less true outliers for the same number of outliers detected) than the KNN for the real-world dataset, but increases with parameter, same for the precision and AUC. The best LOF model is the model with k equal to 9. As for the KNN, the models are less efficient for the real-world dataset.

c) Isolation Forest

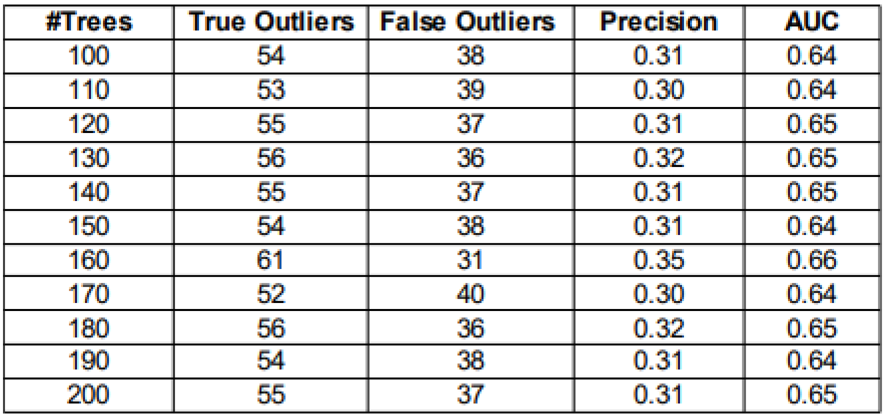
Table 5 is the results for the Synthetic Dataset and **Table 6** is the results of the Cardio Dataset for the Isolation forest. The number of trees is between 100 and 200, with a step of 10.



#Trees	True Outliers	False Outliers	Precision	AUC
100	29	3	0.97	0.981
110	30	2	1.00	0.998
120	30	2	1.00	0.998
130	29	3	0.97	0.981
140	30	2	1.00	0.998
150	29	3	0.97	0.981
160	30	2	1.00	0.998
170	30	2	1.00	0.998
180	30	2	1.00	0.998
190	30	2	1.00	0.998
200	30	2	1.00	0.998

Table 5: results of the Isolation Forest on the Synthetic Dataset

Synthetic Dataset: The results with the Isolation Forest on the Synthetic Dataset are similar with the results with the KNN, but with a less good performance : For the models with a number of tree off 100, 130 and 150, the model does not detect all the outliers (29 instead of 30, meaning that there is 3 instead of 2 false outliers detected). The difference between these two different results for the AUC is very small. The best model here is then with a number of trees of 110, the others would give a higher time and space complexity. The KNN has a bit more performance, due to the fact it is a distance-based technique.



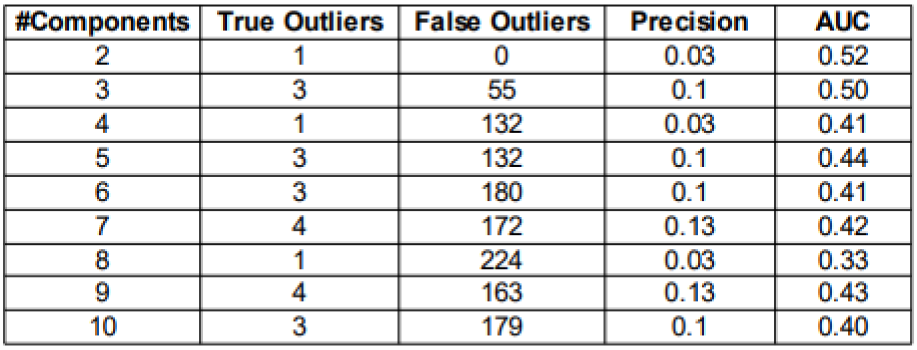
#Trees	True Outliers	False Outliers	Precision	AUC
100	54	38	0.31	0.64
110	53	39	0.30	0.64
120	55	37	0.31	0.65
130	56	36	0.32	0.65
140	55	37	0.31	0.65
150	54	38	0.31	0.64
160	61	31	0.35	0.66
170	52	40	0.30	0.64
180	56	36	0.32	0.65
190	54	38	0.31	0.64
200	55	37	0.31	0.65

Table 6: results of the Isolation Forest on the Cardio Dataset

Cardio Dataset: For the same number of outliers detected, the Isolation Forest detects more outliers than the LOF and the KNN. Precision and AUC are relatively better. The performance for the synthetic dataset is, however still better than for the Cardio dataset. Performance has a peak for a number of tree of 160 : it tends to increase until 160 and then tends to decrease. The difference is rather small between all the models, due to the small difference in the detection of true outliers. Increase the contamination rate could increase the difference but could also decrease the number of false outliers and then decrease the performance.

d) Gaussian Mixture Model

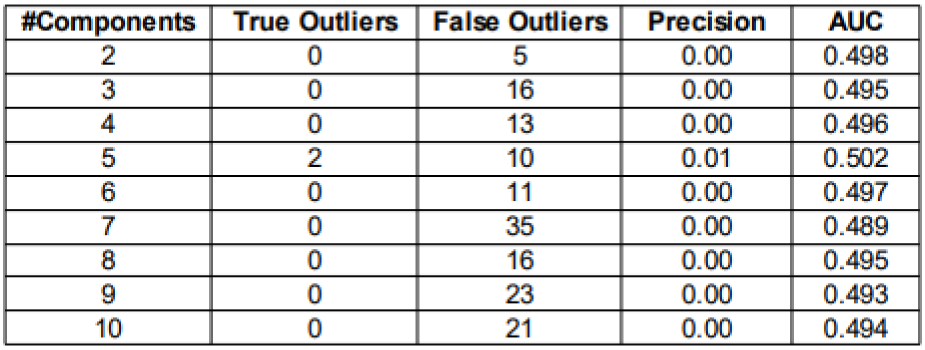
Table 7 and **Table 8** give the results for respectively the Synthetic and Cardio dataset, the number of components varying between 2 and 10.



#Components	True Outliers	False Outliers	Precision	AUC
2	1	0	0.03	0.52
3	3	55	0.1	0.50
4	1	132	0.03	0.41
5	3	132	0.1	0.44
6	3	180	0.1	0.41
7	4	172	0.13	0.42
8	1	224	0.03	0.33
9	4	163	0.13	0.43
10	3	179	0.1	0.40

Table 7: results of the Gaussian Mixture Model on the Synthetic Dataset

Synthetic Dataset: The technique has rather bad results, the performance decreasing with the number of components, except for the two last rows. The number of components equal to 8 gives the worst performance. Each model detects less than 5 true outliers, while, except for 2 components, detects a lot of false outliers. The AUC of the majority of the models built was under 0.50, meaning that they are less performing than a random classifier. Furthermore, this result show at least that the precision is not necessarily a good method while the AUC is taking into account the number of false outliers and penalize the performance score for each false outlier detected.



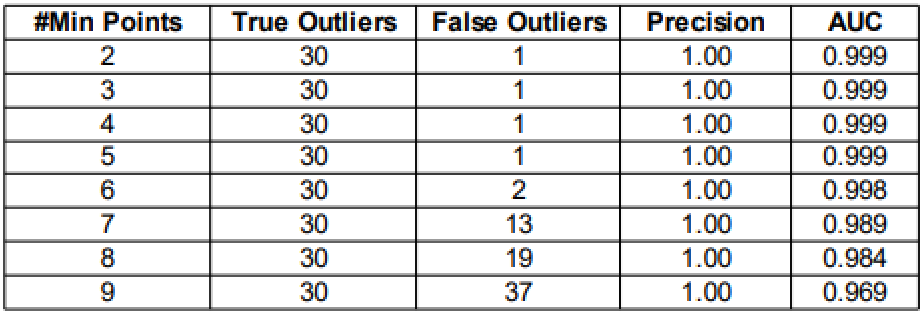
#Components	True Outliers	False Outliers	Precision	AUC
2	0	5	0.00	0.498
3	0	16	0.00	0.495
4	0	13	0.00	0.496
5	2	10	0.01	0.502
6	0	11	0.00	0.497
7	0	35	0.00	0.489
8	0	16	0.00	0.495
9	0	23	0.00	0.493
10	0	21	0.00	0.494

Table 8: results of the Gaussian Mixture Model on the Cardio Dataset

Cardio Dataset: Only one model manages to detect true outliers (with 5 components), all the others detected inliers as outliers but only two, with comparison to the Synthetic Dataset. The performance in terms of precision is better for the Synthetic Dataset but the AUC is generally higher for the Cardio Dataset, due to the lower number of false outliers detected by the Gaussian Mixture Model technique. However, the best model for the Synthetic Dataset is still better than the the best model for the Cardio Dataset (0.52 versus 0.50).

e) DBSCAN

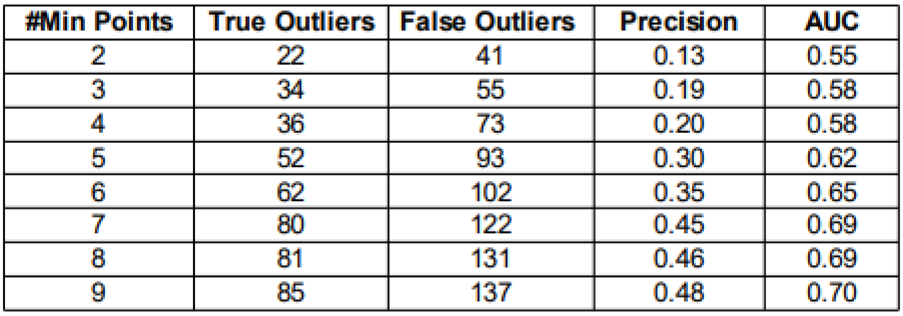
The two last tables, **Table 9** and **Table 10** give the results for the DBSCAN. The parameter of DBSCAN, the number of minimum points varies between 2 and 10.



#Min Points	True Outliers	False Outliers	Precision	AUC
2	30	1	1.00	0.999
3	30	1	1.00	0.999
4	30	1	1.00	0.999
5	30	1	1.00	0.999
6	30	2	1.00	0.998
7	30	13	1.00	0.989
8	30	19	1.00	0.984
9	30	37	1.00	0.969

Table 9: results of the DBSCAN on the Synthetic Dataset

Synthetic Dataset: The model detects all the outliers of dataset, but the performance decreases with the number of minimum points to be a core point of a cluster. Adding more minimum points is more restrictive to build a cluster, then; more data points are not included. Increasing the radius epsilon would decrease the number of false outliers, with a low possibility that it impacts the number of true outliers detected, due to the distance between the outliers and the inliers inside the dataset. The model chosen is then the model with 2 minimum points, because models with 3 minimum points or more does not impact the performance but decreases the time and space complexity.



#Min Points	True Outliers	False Outliers	Precision	AUC
2	22	41	0.13	0.55
3	34	55	0.19	0.58
4	36	73	0.20	0.58
5	52	93	0.30	0.62
6	62	102	0.35	0.65
7	80	122	0.45	0.69
8	81	131	0.46	0.69
9	85	137	0.48	0.70

Table 10: results of the DBSCAN on the Cardio Dataset

Cardio Dataset: Contrary to the Synthetic Dataset, the performance of the model increases with the number of minimum points for a data point to be considered as a core point of the cluster. It is because the algorithm is not able to detect all the outliers present in the dataset (actually, the algorithm does not detect all the outliers even with a number of minimum points set at 9). Again, as expected, it has a higher performance for the Synthetic Dataset than for the real-world dataset. The technique seems also to show a better performance than the others : detecting much more true outliers and more false outliers but does not impact as much the AUC score to have a less high score than the Isolation Forest.

f) ROC curves of the best models

The two last figures, **Figure 4** and **Figure 5** shows the ROC curve for the best models of each technique for the Synthetic Dataset and the Cardio Dataset respectively. Since the curves are just here to visualize the results obtained by the AUC, the two figures are in the appendix.

For the Synthetic Dataset, it clearly shows that the KNN and the Isolation Forest performs better than the others. Both are close to the perfect classifier, with the KNN a bit closer. The GMM is the technique that has the baddest performance, closest to the random classifier. Then, we have DBSCAN that performs better than the LOF.

For the Cardio Dataset, DBSCAN has the best performance, following by the Isolation Forest, the KNN and the GMM. The GMM is rather similar to the random classifier. Compared to the Synthetic Dataset, none of the curves are close to the curve for a perfect classifier. The difference between method's performances is really small compared to the difference for the Synthetic Dataset.

As a conclusion, for both datasets, Detection with DBSCAN and the Isolation Forest seems the most performing methods. Detection wit KNN works well with the synthetic dataset because of the high distance between the inliers and the outliers. The Gaussian Mixture Model, however has the worst performance, close to the random classifier.

6. Conclusion

In this thesis, different definitions of outliers have been given, as well as applications where the outlier detection can be used. Then, 5 different outlier detection technique have been described : The Neighborhood and the K Nearest Neighbors based on distances, the Local Outlier Factor based on densities, the Isolation Forest based on ensemble, the Gaussian Mixture model based on distribution, and the DBSCAN based on clustering. Variations or similar techniques have also been reviewed briefly. Also, other statistical tests about outliers such as the box plot, the trimmed mean, the Extreme Studentized Deviate, the Dixon-type test and the histogram-based outlier score. Thereafter, two evaluation methods have been reviewed: the Precision and the ROC curve paired with the Area Under the Curve. The methodology and results for each technique have been given afterwards.

To conclude, the limits of the study will be mentioned, with the potential questions that can be asked and recommendations for future studies.

6.1 Limits

Just as others similar studies about outlier detection or in general, a part of the results obtained by these algorithms, or how the implementation of the algorithm are limited.

First, the techniques mentioned (more specifically the neighborhood/KNN and the Local Outlier Factor) are rather old and were the first outlier detection techniques to be introduced. More recent techniques could have more performance on the real-world dataset, or even the the variations of these different techniques (The COF/LoOP for the Local Outlier Factor, or even a enhanced version of the Gaussian Mixture Model with the Local Serving Projections).

Secondly, the evaluation method was only performed on the performance of the model, and not the other aspects such as the robustness and the time and space complexity, just as Domingues et al did with different unsupervised methods. It is not enough to evaluate on precision. A model would be considered as better than another one if it has a higher precision, robustness, and less scalability/memory usage. But if they have a performance relatively similar (for example, the KNN, the DBSCAN and the Isolation Forest for the synthetic dataset), it is important to focus on the other sides: one of them could have a time/space complexity more important, resulting that the other would be considered with more performance.

Finally, it is not really a real limit of the study, but rather a problem with one technique: the Gaussian Mixture Model. It performs relatively bad for both datasets, with a performance similar to a random classifier. It detected only a small part of real outliers but a lot of inliers as outliers for the Synthetic

Dataset, a small part of inliers were detected as outliers in the Cardio Dataset. But only one model found true outliers (only 2). However, for the application dataset, the explanation could be the fact that the Gaussian Mixture Model relies on the assumption that the components are Gaussians, but nothing can prove that this hypothesis is founded for the Fetal Hearts. The performance of the model is then unreliable. In the case of the Synthetic Dataset, the data points have been generated with the numpy function random.uniform, building a sample with the uniform distribution. Therefore, it is not necessarily reliable, even if it's possible to transform points of an uniform distribution to a normal distribution, called the Box-Muller Transform, by Box and Muller (1958) [39].

6.2 Questions and Recommendations

First, the results clearly show the difference of performance between a synthetic dataset and a real-world dataset, as expected. As said before, the techniques should be more efficient on a synthetic dataset, because a synthetic dataset would be less complex than a real world dataset. In this case, the Synthetic Dataset artificially created by Python has less data points (630 versus 1831), and less dimensionality (2 versus 21). The difference of complexity between these two has been felt when each algorithm was running into a loop, to vary the parameter. It takes seconds for one model to be executed with the real-world dataset, while for the synthetic one; the time necessary to execute a model was so small that it was not visible.

Also, in terms of performance/precision only, it was easier for the different techniques to detect all the outliers present in the synthetic dataset rather than in the Cardio Dataset. Indeed, the number of outliers present in the synthetic dataset represents less than 5% of the data points, while the outliers in the real-world dataset represents more or less 10% of the dataset. It is then impossible for the algorithm to detect all the outliers with a contamination rate (or the threshold set at the 95th percentile) of 5%. Furthermore, the outliers of the synthetic dataset are also easier to detect, due to less dimensionality (real-world dataset are rather high-dimensional) and a size of the dataset less high, there will be less chances that the inliers would be detected as outliers. Plus, more inliers imply that outliers will be less distinguishable, therefore less easy to detect them.

For the synthetic dataset, the most performing techniques are the KNN and the DBSCAN (followed by the Isolation Forest). These two techniques are related to distances: While the KNN is directly related to a distance with the mean k-distance, the clusters made by the DBSCAN are made via a certain radius. In terms of distance, outliers were rather far than the high-density group of inliers. A question can then be asked: Does these two models are detecting all outliers because of the high distance? What if the outliers were closer, would they all always be detected as outliers by the algorithm? The outliers have been made manually to make sure that they are distinguishable from the inliers, since the values of the features for the synthetic dataset do not have any real signification (no prior knowledge).

For the real-world dataset, the DBSCAN has the best performance, followed by the Isolation Forest. It is clear that the Isolation Forest performs better than the KNN and the LOF, and the GMM too. The Isolation Forest has a higher ROC curve and AUC than the two others and it is rather valid, but not necessarily with the DBSCAN : the AUC is clearly higher, but the algorithm doesn't detect the same number of outliers in total. The DBSCAN is actually not limited by a threshold or a contamination rate. Another question here can be asked: Does the Isolation Forest would be more performing than the DBSCAN without being limited by a contamination rate ? Written differently, does the DBSCAN would be less efficient if the number of outliers is limited?

Finally, another question can be asked about the contamination rate for the Local Outlier Factor and the Isolation Forest, the threshold for the KNN and also the radius for the DBSCAN : How to set these rates, the threshold or the radius ? For DBSCAN, it was impossible to set at a same radius: it actually needed to be set differently for both datasets and to calibrate them before in order to have acceptable results, because the distance between points is different from one dataset to another. For the KNN technique, it is quite difficult to set a threshold distance suitable, depending on the application, but this distance threshold should be set by experts. For the contamination rate, depending on the application, can be sent depending on the opinion of the experts, as a percentage of a sample. For example, for the detection of diseases, an expert could say « Among 500 people, there are about 25 people who have this disease », meaning that the contamination set for a dataset useful to detect that illness could be set to 25/500 = 5%. About the radius, the user should have a global idea about how the cluster looks like in a two-dimensional space; it has to be calibrated first. It is however more difficult to visualize with higher dimensionality.

As a recommendation for a future study, finding a way to compare techniques that are limited by contamination rate with the ones that are able to detect outliers without restriction on the number, with an evaluation method that normalizes with the number of outliers, so that the performance can really be compared. For example, a performance for a data point would be useful to compare every type of outlier detection technique. This would maybe show that the Isolation Forest actually has a better performance than the DBSCAN. Also, a good idea to verify if the Gaussian Mixture Model performs well, is to use a synthetic sample coming from a normal distribution with the same outliers and verify if it has better result. Another recommendation would be to use one type of technique, for example, the LOF and his variations, to see if the performance is better than the original technique as expected.

7. References

[1]: M. A. F. Pimentel, D. A Clifton, L. Clifton, L. Tarassenko, A review of Novelty Detection. *Signal Processing*, Vol. 99, p.215-249, ISSN 0165-1684, 2014.

[2]: F. Stambaugh, Risk and value at risk. *European Management Journal*, p.612-621, 1996.

[3]: K. Singh and S. Upadhyaya, Outlier Detection: Applications and Techniques. *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 1, No. 3, p.307-323, Jan. 2012.

[4]: A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, Outlier detection approaches for wireless sensor networks: A survey. *Computer Networks,* Vol. 129, p. 319-333, Dec. 2017.

[5]: I. Ben-Gal, Outlier Detection. In Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA, 2005.

[6]: H. Wang, M. J. Bah and M. Hammad, Progress in Outlier Detection Techniques: A Survey. in IEEE Access, Vol.. 7, p. 107964-108000, 2019.

[7]: J. W. Osbourne, A. Overbay, The power of outliers (and why researchers should ALWAYS check for them). *Practical Assessment, Research, and Evaluation*: Vol. 9, Art 6, 2004.

[8]: L. Huang, A. D. Joseph, B. Nelson, B.I.P. Rubinstein, and J. D. Tygar, Adversarial machine learning, In Proceedings of the 4th ACM workshop on Security and artificial intelligence (AISec '11), Association for Computing Machinery, New York, NY, USA, 43–58, 2011.

[9]: H. Aguinis, R. K. Gottfredson, and H. Joo, Best-Practice Recommendations for Defining, Identifying, and Handling Outliers. *Organizational Research Methods*, p.270-301, 2013.

[10]: S. Walfish, A review of statistical outlier methods, *Pharmaceutical Technol.*, Vol. 30, no. 11, p.1-5, 2006.

[11]: Davies L. and Gather U., The identification of multiple outliers. *Journal of the American Statistical Association*, p.782-792, 1993.

[12]: M. Goldstein and A. Dengel, Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. In *Proc. Poster Demo Track*, p.59-63, Sep. 2012.

[13]: Y. Chen, D. Miao, and H. Zhang, Neighborhood outlier detection. *Expert Systems with Applications*, 37, p.8745-8749, 2010

[14]: C. Stanfill, & D. Waltz, Towards memory-based reasoning. *Communications of the ACM*, p.1213–1228, 1986

[15]: V. Hautamäki, I. Kärkkäinen and P. Fränti, Outlier Detection Using k-Nearest Neighbour Graph. *Proceedings of the 17th International Conference on Pattern Recognition*, p.430-433, Vol. 3, 2004

[16]: M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, LOF: Identifying density-based local outliers. *ACM SIGMOD Rec.*, Vol. 29, No. 2, p.93-104, 2000.

[17]: E. Schubert, A. Zimek, and H.-P. Kriegel, Local outlier detection reconsidered: A generalized view

on locality with applications to spatial, video, and network outlier detection. *Data Mining Knowl. Discovery*, Vol. 28, No. 1, p.190-237, 2014.

[18]: H. Kriegel, P. Kröger, E. Schubert, and A. Zimek, LoOP: Local outlier probabilities. in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, p.1649-1652, Nov 2009.

[19]: J. Tang, Z. Chen, A. Fu, and D. Cheung, Enhancing effectiveness of outlier detections for low density patterns. in *Advances in Knowledge Discovery and Data Mining*. Springer, Berlin, p. 553-548, 2002.

[20]: S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, LOCI: Fast outlier detection using the local correlation integral. In *Proc. 19th Int. Conf. Data Eng.*, p.315-326, Mar. 2003.

[21]: D. Ren, B. Wang, and W. Perrizo, RDF: A density-based outlier detection method using vertical data representation. In *Proc. Int. Conf. Data Mining*, p.503-506, Nov. 2004.

[22]: W. Jin, A. K. Tung, J. Han, and W. Wang, Ranking outliers using symmetric neighborhood relationship. In *Proc. 10th Pacic-Asia Conf. Adv. Knowl. Discovery Data Mining*, p.577-593, 2006.

[23]: F. Keller, E. Müller, and K. Bohm, HiCS: High contrast subspaces for density-based outlier ranking. In *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, p.1037-1048 , Apr. 2012.

[24]: R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Trans. Knowl*. Discov. Data 10, 1, Art. 5, July 2015.

[25]: F. T. Liu, K. M. Ting, and Z.-H. Zhou, Isolation forest, in *Proc. 8th IEEE Int. Conf. Data Mining*, p.413-422, Jul. 2008.

[26]: Y. Zhao & M. K. Hryniewicki, Dcso: Dynamic combination of detector scores for outlier ensembles. In *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2018.

[27]: X. Yang, L. J. Latecki, and D. Pokrajac, Outlier detection with globally optimal exemplar-based GMM. In *Proc. SIAM Int. Conf. on Mining (SDM)*, p.145-154, Apr. 2009.

[28]: A. Dempster, N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 1977

[29]: X. Tang, R. Yuan, and J. Chen, Outlier detection in energy disaggregation using subspace learning and Gaussian mixture model. *Int. J. Control Autom.*, Vol. 8, No. 8, p. 161-170, 2015.

[30]: X. He, & P. Niyogi, Locality Preserving Projections, *NIPS'03: Proceedings of the 16th International Conference on Neural Information Processing Systems*, p.153-160, Dec 2003.

[31]L. J. Latecki, A. Lazarevic, and D. Pokrajac, Outlier detection with kernel density functions, in *Proc. 5th Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, p.61-75, 2007.

[32]M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, p.226-231, Aug. 1996.

[33]M. Çelik, F. Dadaşer-Çelik and A. Ş. Dokuz, Anomaly detection in temperature data using DBSCAN

algorithm. *International Symposium on Innovations in Intelligent Systems and Applications*, p.91-95, 2011.

[34]: J. MacQueen, Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Math. Statist, Prob.*, Vol. 1, p.281-297, Jun. 1967.

[35]: S. Guha, R. Rastogi, and K. Shim, CURE: An efficient clustering algorithm for large databases. In *Proc. ACM SIGMOD Int. Conf. Manage. Data*, p.73-84, Jun. 1998.

[36]: R. Domingues, M. Filipponne, P. Michiardi, and J. Zouaoui, A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, Vol. 74, p.406-421, Feb. 2018.

[37]: C. D. Brown and H. T. Davis, Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, Vol. 80, Issue 1, p.24-38, 2006.

[38]: F. Tortorella, An Optimal Reject Rule for Binary Classifiers. In: Ferri F.J., Iñesta J.M., Amin A., Pudil P. (eds) Advances in Pattern Recognition. Lecture Notes in Computer Science, Vol. 1876. Springer, Heidelberg, 2000.

[39]: G. E. P. Box and M. E. Muller, A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics, Ann. Math. Statist*, p.610-611, Jun. 1958.

8. Appendix

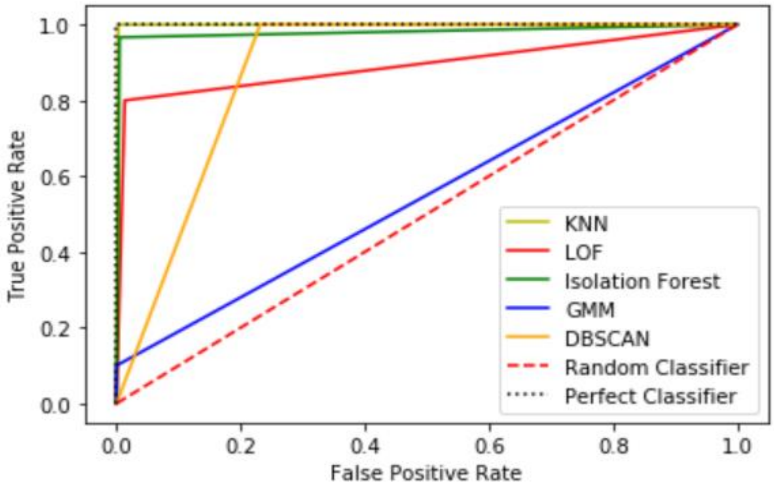


Figure 4: ROC curve for the best different models on the Synthetic Dataset

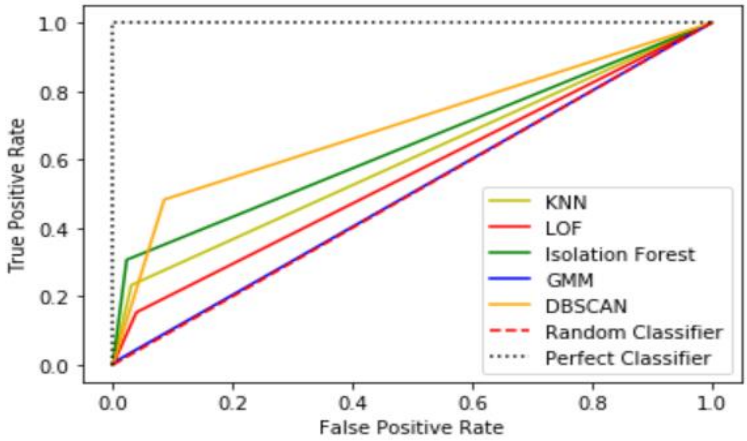


Figure 5: ROC curve for the best different models on the Cardio Dataset