IJCI V9-01 (2022) 60–73
International Journal of Computers and Information (IJCI)
Available online at *https://ijci.journals.ekb.eg/*

A Comparative Study for Outlier Detection Strategies Based On Traditional Machine Learning For IoT Data Analysis.

Amina Elmahalawy, Hayam Mousa, Khalid M. Amin

Information Technology dept., Faculty of Computers and Information, Menoufia University, Shebin Elkom 32511, Egypt.
amina.elmahalawi1@ci.menofia.edu.eg, hayam.mousa@ci.menofia.edu.eg, k.amin@ci.menofia.edu.eg.

Abstract

Internets of Things (IoT) systems are increasing very fast. They have different types of wireless sensor networks (WSN) behind them. These networks have many applications that are a portion of our life such as healthcare, agricultural, mechanical, and military systems applications. Therefore, a massive amount of data was collected. Outlier detection is one of the essential fundamental problems in these applications. It helps to discover erroneous, imperfect, and noisy nodes. There are various techniques used to detect this outlier. Machine learning algorithm-based approaches are exceptionally much valuable and successful among them. This paper is concerned with the study of outlier detection techniques. It categorizes them into different approaches, such as Statistical, Nearest_Neighbor, Clustering, Subspace, Ensemble-based, and other approaches. These approaches are examined in detail. This study is concerned with determining the best outlier detection method that can be used to detect the outlier in the IoT data analysis. In conclusion, the experimental results show that the Isolation Forest, HBOS, and CBLOF approaches give better performance in terms of precision, Area under the curve (AUC), and execution time than other algorithms.

Keywords: IoT; Outlier detection; Machine learning; Local Outlier Factor (LOF); Isolation Forest (IF); Histogram Based Outlier Score (HBOS); Feature Bagging (FB).

1. Introduction

The Internet of Things is considered one of the fastest innovations. It contains billions of objects or devices that use several sensors to collect different types of data. It is confronting numerous challenges and new emissions. These challenges include hardware, security, privacy, heterogeneity, virtualization, and data analysis. This study centers on its data analysis issues, precisely data quality [1]. Data quality issues include uncertainty problems, noise that contains errors or outliers, inconsistencies that contain inconsistencies in symbols or names, and incompleteness that contains missing values. Highlight the outlier detection problem as one of the most critical challenges in data management in IoT is the idea of this paper. Internet of Things is considered one of the most data sources. The collected data is the basis for providing new and intelligent services in it. IoT data will be the central part of big data by 2030, where the numbering of the connected sensors/devices will attain one trillion [2]. However, the data generated from the sensor and other objects sometimes contain outliers that degrade observations' quality and reliability. The outlier, in general, is a pattern that differs from other observations, does not correspond to expected normal behavior, or corresponds well to specify abnormal behavior. This issue plays a crucial role, and it is interested in many applications like fraud detection, intrusion detection, healthcare, crime detection, network intrusion. Identifying outliers in the data can affect the process of knowledge discovery. Having precise information is more severe for everybody. It is vital to have precise information about your employee. It is fantastic to have exact client information. So, you will get in touch with your clients if wanted. Having the most precise information helps in your marketing efforts. So that, data cleansing is more vital as it

improves the data quality. It will increase productivity and efficiency. All misinformation and noisy data are gone once you are cleaning your data. It is leaving you with the best quality information.
It confirms your employees don't need to go through ancient documents and permit them to form most of their work hours. Furthermore, it confirms you get the correct information. It helps in reducing unexpected costs. Having harmonious errors in your work also can harm your company's name. In this study, a survey of outlier detection is presented and its various techniques. The outlier definition, the various outlier types, causes, detection methods, and challenges are studied. This study is organized as follows; the background is presented in section 2. The related works and well-known approaches are discussed in section 3. The experimental measurement and results are discussed in section 4. Results analysis and decision are discussed in section 5. Recommendations are discussed in section 6. In the end, the conclusions are derived in section 7.

2. Background

2.1 Internet of things (IoT)

It is the network of hardware objects and other elements embedded in electronics, software, sensors, actuators, and communication that specialize in these things to communicate and exchange data [3].
IoT architecture consists of different layers as follow [4, 5]:

1. Perception layer:
   It is the hardware layer. It contains devices equipped with sensors and microprocessors. This layer produces, gathers information, and sensing the state of this object.
2. Network layer:
   This layer firstly allows objects to communicate, talk, and participants' data with each other over different wireless or wired networks, then the data is collected, aggregated, and sent to the middleware layer [4].
3. middleware layer:
   It is the service layer. It is used to observe and handle services that are wanted by applications and users.
4. Interface layer:
   It permits the interaction between clients and applications by allowing exchange, communication, and handling event processing between various objects regardless of different physical platforms and hardware [5]. The following Fig.1 illustrates IoT architecture [6].

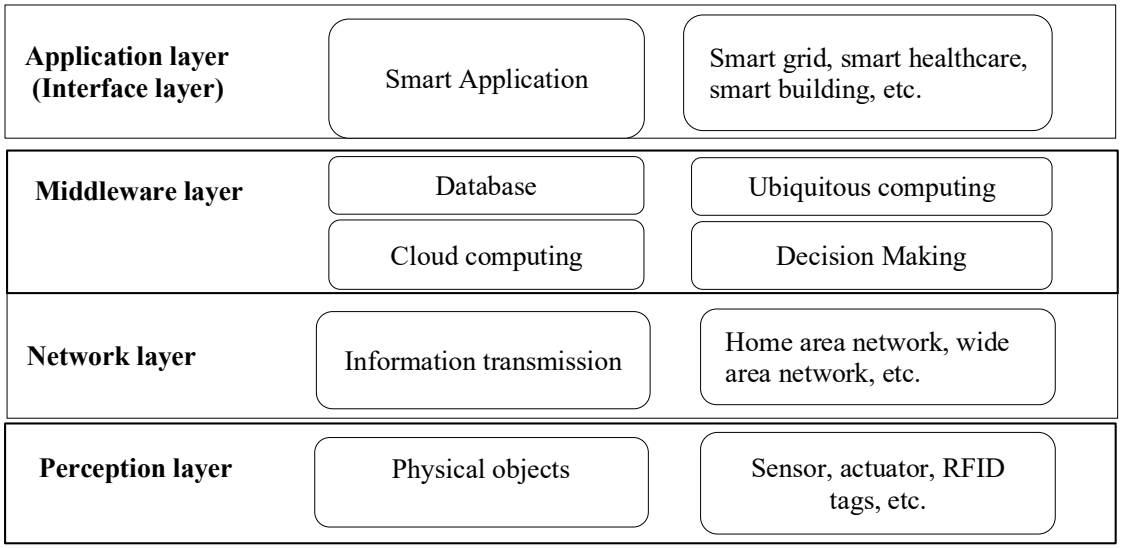

Fig. 1. IoT architecture [6]

In order to achieve the best exploitation of this environment, data quality must be ensured through some techniques. One of them is outlier detection. In this study, outlier detection is studied and discussed to estimate its efficacy to exploit its environment fully.

2.2 Outlier definition

Outliers are data that do not correspond to a well-defined concept of normal behaviour. It is called in various ways like anomalies, intrusions novelties, exceptions, frauds, etc. [7]. Outlier is categorized into three categories, global outlier, contextual and collective outlier. A data object should be a global outlier if it deviates from the rest of the dataset. In Fig. 2, an example of a global outlier using the distance-based method, the points in the region R is significantly deviating from the rest of the dataset and also point $O3$.

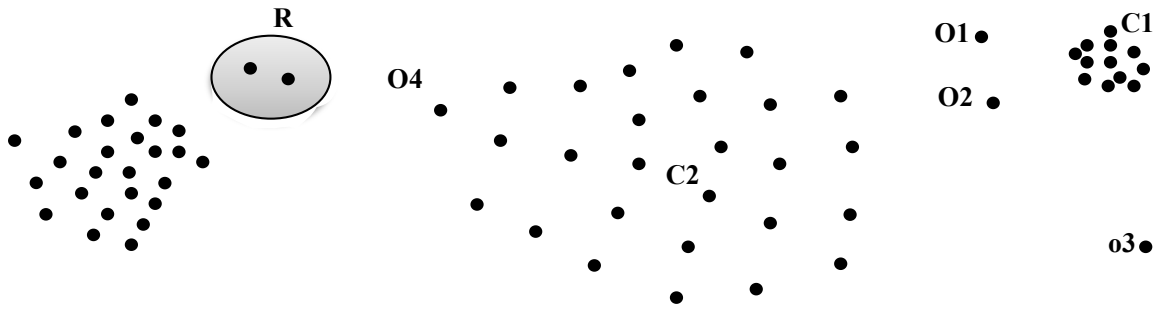

Fig. 2.Example of global and local outlier

Contextual outlier is a type of outlier that depends on location, time, etc., it also called a conditional outlier; for example, today in Toronto, the temperature is 28ºC, is it an outlier? Yes, it is abnormal in winter; it is normal if it is on a summer day. To determine the contextual attributes and context, it needs to know more setting information. Collective outlier -a subset of data objects collectively deviates significantly from the entire data set. In Fig. 1, an example of a Local outlier if its density deviates significantly from the local region in which it occurs. For example O1, O2 are local outliers according to cluster C1 using the density-based method.

This study focuses on point outliers. When a data point can be considered an outlier regarding the remainder of the data, this observation is termed a point outlier. Outliers exist in every actual data set. The causes for these outliers are an outcome of malicious activity, hardware failure, human error, Instrumentation error, setup error, sampling errors, data-entry error, modification in system behavior, and environmental changes. In the next section, outlier detection techniques to detect this outlier are presented.

2.3 Outlier detection techniques

Many techniques are designed to detect outliers. In [8], discuss the various recent strategies for outlier detection. The outlier detection techniques can be grouped into six categories [9, 10], as shown in Fig. 3.

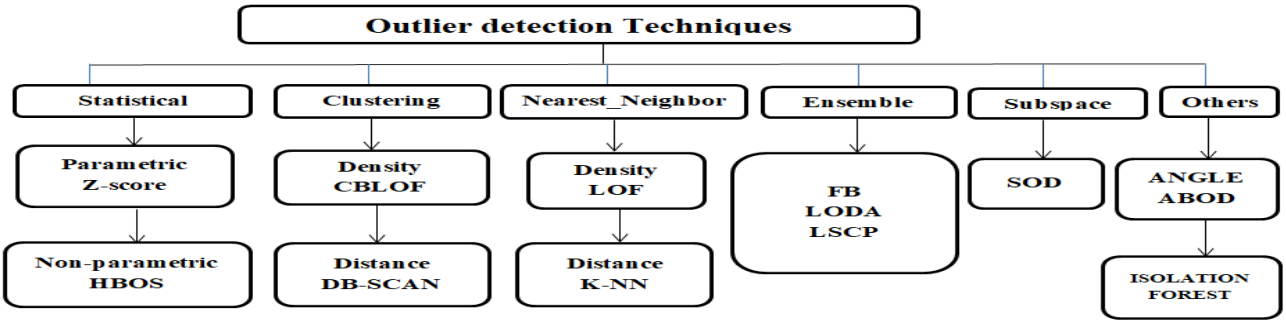

Fig. 3.a taxonomy for outlier detection algorithms

The following table 1 summarizes a brief description of the outlier detection methods concerned with the advantages and disadvantages of each method.

Table 1.A brief description of the outlier detection methods.

types	Description method	Advantages	Disadvantages
Statistical-Based Techniques	It is based on statistical methods that assume a distribution model or probability of fit to a given data set.	1-If the main data distribution propositions are correct; statistical approaches extend a statistically justifiable solution for outlier detection. 2-The outlier score computed from statistical approaches is connected with a confidence interval, which may be used as additional information while deciding regarding any test instance. 3-can operate in training data without labelling data if the distribution estimation is robust to an anomaly.	1-The key cons of statistical approaches are that they depend on the supposition that the data is coming from a particular distribution. 2- Choosing the best statistic is not an outright task. Especially, making hypothesis tests for complex distributions required to fit high dimensional data sets is counterintuitive. 3-Histogram-based methods are comparatively easy to implement, but a key fault of such approaches for multivariate data is that they cannot pick up the interactions between various attributes.
Clustering-Based Techniques	Clustering is used to group similar data points. Clustering-Based Techniques rely on the supposition that normal data instances belong to clusters in the data while outliers do not belong to any cluster as CBLOF[14]	1- Can operate in an unsupervised mode without any labelled data. 2-These approaches can be adapted to other complex data types. 3-The testing phase is fast	1-the performance depends on the clustering algorithm's effectiveness in capturing the cluster structure of normal instances 2-Many techniques detect anomalies as a by-product of clustering and hence are not optimized for anomaly detection
Nearest_Neighbor based detection	Identifying anomalies by using neighbourhood information. Examples include *k*NN[10], LOF[12], etc.	1-Independent of the data distributions 2- Intuitively understood and easily interpreted	1- Sensitive to parameters 2-Relatively poor performance
Ensemble-based detection	Integrating various outlier detection results to achieve a consensus. Examples are FB [18], LSCP [19], LODA[21] and so on.	1- High accuracy 2- Less sensitive	1- Inefficient 2- Determine the accurate meta-detectors is difficult
Subspace-based detection	Finding anomalies by sifting through various feature subsets. Examples SOD[8], etc.	1- High efficiency 2- Very effective in several cases	1- Determining the relevant feature subspaces for outliers is nontrivial and difficult

1. Statistical-Based Techniques: An earlier approach to deal with an outlier detection problem, techniques in labeling outliers. It is based on the relation with the distribution model. These strategies are divided into two basic categories:
   -Parametric methods: It is used when the data point has a hypothesis of the distribution model. The two familiar models for outlier detection are the Gaussian Mixture and the Regression model [9].
   -Nonparametric methods: It is used when the data point is not normally distributed, and there is no past knowledge about the distribution, so it is called the distribution-free algorithm. Some standards must be implemented to determine if the observation is an outlier or inliers in the dataset. The common techniques in this model are histograms and kernel density [9].
2. Clustering-Based Techniques: It is used to determine similar groups in the feature space of data input. Many various clustering algorithms can be used.
3. Distance-Based Techniques: It is based on computing the distance between observations. It is viewed as an outlier; based on its distance from its nearest neighbors.
4. Density-Based Techniques: In these strategies, the outlier is found in a region with low density; however, the normal point is found in very dense neighborhood regions.
5. Angle-Based Techniques: The main idea for angle-based approaches is the angle variance between pairs of the remaining objects [11].
6. Subspace-Based Techniques and Ensemble-Based Techniques: The main idea of these techniques based to integrate the results from different techniques to get additional strong models with high performance to detect any anomalies found in the data more efficiently.

3. Related work and well-known approaches

In this section, the foremost reliable and well-known techniques are presented for detecting the outlier.

3.1 Local Outlier Factor (LOF)Technique:

It is a standard outlier detection approach [12]. It is based on calculating the local density of a given data point with its neighbours.

LOF Score to be computed, view the following steps:

1. The KNN (k-nearest-neighbors) must be calculated for each record x. We find more than *k* neighbors if distance tie of the k_{th} neighbor.
2. The local density is rated by calculating the local reachability density (LRD) using these *k* nearest neighbors N_k [12].

$$\mathrm{LRD_K} = 1/(\frac{\sum_{O \in N_k(x)} d_{k(x,o)}}{|N_k(x)|}) \quad (1)$$

The d_k is reachability distance. In clusters with high dense will used the Euclidean distance.

3. LOF score is computing by matching the LRD of a record with the LRDs of its k –Neighbors [12]:

$$\mathrm{LOF}(x) = \frac{\sum_{o \in N_k(x)} \frac{LRD_k(O)}{LRD_k(x)}}{|N_k(x)|} \quad (2)$$

The LOF score is a proportion of densities; mostly, when LOF > 1, it is set as an outlier, but that is not always true. For example, let's say we know that we only have one outlier in the data, then we take the maximum LOF value among all the LOF values, and the point corresponding to the maximum LOF value is considered as an outlier. The Local Outlier Factor algorithm is shown below [12].

Algorithm 1

Input: Positive integer k, dataset D
Output: Anomaly scores for all points in D
suppose: k_distance(D, P) – a matrix that contains the k_distance neighbors and their k_distances
Reach_dist_k(P) –Local Reachability Density of each $P \epsilon D$
START
Local Outlier Factor ⟵ *null*
FOR each point p
KNN neighbours ⟵ *k_distance(D,k)* **step 1**
Local reachability distance ⟵ *reach_dist_k(KNNeighbors,k)* **step2**
FOR each p in KNN neighbors
Templof[i] ⟵ *sum (lrd[oϵN(p)])/lrd[i]/|N(P)|* **step3**
local outlier factor ⟵ *maximum(lof,templof)*
RETURN top(lof)
END

3.2 Histogram-Based Outlier Score Technique (HBOS)

It is a statistical-based technique [13]. It computes an outlier score by creating a histogram with a fixed or a dynamic bin width. It has two modes, static mode bandwidth, and dynamic mode bandwidth. In the static mode, each bin has the toke bin width evenly disseminate through the value extent. The bin width can change in the dynamic mode, but it is reasonable to assign a minimum number of examples in a bin. The outlier score is to be computed; firstly, the histograms are smooth to one in peak. Then, the score is inverted so that normal examples have low scores and outliers have a high score [13].

$$HBOS(V) = \sum_{i=0}^{d} \log \frac{1}{hist_i(v)} \quad (3)$$

In Equation(3), Where $hist_i$ Is the height of the feature i corresponding to the bins it is located at, and v is the node $v \in G$ The default value for the number of bins k is set to 10 [13].

3.3 K-Nearest neighbor based method

It is a nonparametric technique for classifying data samples [14]. First, it Calculates the approximation distances among various points on the input vectors, then specifies the unlabelled point to the class of its K-nearest neighbours. The k parameter is serious in the process of the k-NN classifier, and different (k) values can cause different performances. If the number of neighbours used for prediction k is big, it consumes a lot of classification time and affects the prediction accuracy. It is very easy to understand this technique with few predictor variables, but it has large storage requirements. It is also critical to choose the similarity function used to compare instances. There is no detected method to select k, except through cross-validation. Therefore, the computation technique is more expensive. The K-nearest neighbour (k-NN) algorithm is shown below [14].

Algorithm 2

Input: Positive integer k, dataset D
Output: Anomaly scores for all points in D
index ⟵ Build R StarTree (D)
results ⟵ []
for *Point* $p \in D$ **do**
    distance ⟵ FindKthNearestNeighbourDistance$(k, p, index)$
    Add *distance* for p to *results*
End
results ⟵ Sort Descending (*results*)
return *results*

3.4 Cluster-based Local Outlier Factor (CBLOF)

CBLOF Calculates the outlier score based on combing the cluster-based method with the local outlier factor technique [15]. To compute the outlier score by the distance of each example to its cluster center multiplied by the examples belonging to its cluster. It is calculated using the following steps:

1. Get cluster Ci for $i = 0 \ldots k$ from input graph G using any clustering algorithm,
   Where $C = C1, C2, \ldots, Ck$, such that $|C1| \geq |C2| \geq \ldots \geq |Ck|$, and k parameter is the total the number of clusters.
2. The Cluster C to Large cluster and Small cluster based on two conditions:
   (a) $|C1,| + |C2| + \ldots + |Cb| \geq |G|\ \alpha$
   (b) $\frac{|Cb|}{|Cb+1|} \geq \beta$

   Where parameters are:
   α= the percentage of normal instances represented in graph G
   β= the rate of the size of the small cluster to the size of the large cluster
   b = boundary of the Large and Small cluster
3. For data points where $v \in G$ belongs to cluster C_i, and C_i belongs to the small cluster. $v's$ CBLOF score is equal to the size of C_i multiplied by the minimum distance between v and C_j, for $j = 1 \ldots b$. For nodes $u \in G$ belonging to cluster C_i, where C_i belongs to the large cluster, $u's$ CBLOF score is equal to the size of Ci multiplied by the distance between u and C_i.

It is important to know that the performance of CBLOF is highly dependent on the method of clustering used. Some clustering methods are not fit well on anomaly detection tasks [15].

Isolation Forest(IForest)

Isolation Forest (IForest) [16] does not construct a normal node profile before identifying each node. It explicitly isolates anomaly instances by constructing isolation trees. The algorithm is defined as:
Let the input data $G = \{v_1, v_2, \ldots \ldots, v_n\}$ and each node $v_i \in G$ *with* attribute $Q = \{q_{1vi}, q_{2vi}, \ldots \ldots, q_{nvi}\}$ and h be the limiting height for the decision tree.
The anomaly score can be calculated by the following step [19]:

1. Build decision trees by frequently splitting G by randomly selecting attribute $q \in Q$ and divide value $p = (\min(q), \max(q))$ until the tree height equals h.
2. Calculate the expected path length $E(h(x))$ in the IFroest, where $h(x)$ represents the single path length for the sample v in one decision tree
3. Then the anomaly score can be calculated as $s = 2^{(-E(h(x)/c(n))}$ where $c(n)$ describes the average path length of a failing search in the Binary Search Tree.

The main idea is that outlier vertices are less frequent than normal vertices. Thus, outliers are separated from the early partitioning, and they will have a shorter average path length.

There are only two variables that need to be defined in this algorithm:

1. Determine the number of decision trees 2. Determine the size of the tree

3.5 Angle Based Outlier Detection(ABOD)

The idea of ABOF [11] is to detect the outlier based on the variance of the angles between the differences vectors of A to all pairs of points in D weighted by the distance of the point.

3.6 Subspace based methods

SOD approach addresses the problem of outlier identification in various subclasses of a high-dimensional data space. However, this approach is unable to detect an outlier in the starting data extent [17]. So, the Subspace Outlier method searches the subspace parallel to the axis to determine how far the object is from the neighbors in this subspace for each data [20].

3.7 Ensemble-based methods

Feature bagging [18] is an ensemble learning attempt to scale back the correlation between estimators by training them on random samples of features rather than the whole feature set. This algorithm can also be combined with a single classifier, support vector machine, and nearest neighbor. An LSCP technique (Locally Selective Combination in Parallel Outlier) named local region near a test instance has been found in [18]. This includes determining the local area to be the closest training data collection. LODA (lightweight online detector of anomalies) is an ensemble approach; These methods use a range of subspaces or base learners[21].

4. Experimental Measurements and Results

For our analysis, PYOD python outlier detection was used [22], which is a comprehensive and scalable toolkit for outlier detection. It contains various APIs and advanced models.

4.1 Dataset characteristic and Parameter used

In this experiment, some benchmark datasets have used for comparison. Dataset is easily available at Outlier Detection Dataset (ODDS:http://odds.cs.stonybrook.edu/#table 2). Each dataset is partitioned into two parts. 60% of the dataset is specified for training, and the rest 40% for testing, as shown in Table 2.

This experiment is performed ten times independently with random splits, and the average value of alliterations is considered the final output. Our experiment was implemented in Python 3.7, Jupyter Notebook, and was conducted on Windows 10 64-bit version with core i5 processor and 8 GB RAM.

To perfectly evaluate our contribution, various datasets in different domains are presented. They contain the patient's data such as Cardio and Pima; handwritten digits data like Pendigits; also satellite image data like Satellite and Satimage-2, and other multivariate data. Since this study is concerned with IoT data analysis. The experiments used an existing IoT dataset. The description of the satellite dataset consists of features collected from satellite observations as multi-spectral image data.

Table 2 gives the dataset characteristic concerned with the numbers of the sample, dimension, and outlier ratio for each dataset. The comparative methods here , a variety of methods were chosen including outlier detectors like Local Outlier Factor (LOF) [12], cluster-based Local Outlier Factor (CBLOF) [15], Fast Angle-Based Outlier Detection (FABOD) [11], and Subspace Outlier Detection (SOD) [17].Histogram-based outlier detection (HBOS)[13], Ensemble outlier detection methods as Isolation Forest (iForest) [16], outlier ensembles &

combination frameworks like Locally Selective Combination of Parallel Outlier Ensembles (LSCP) [19], and Feature bagging [18].

The above outlier detection methods parameters are set as follows: the k neighbour number is 10 in LOF, FABOD, and SOD. The sub-sampling size and tree number are 200 and 100 in iForest. Finally, the amount of contamination is set to 0.1.

Table 2. DATASET characteristics

Data	#Samples	# Dimensions	Outlier Perc
Cardio	1831	21	9.6122
Musk	3062	166	3.1679
Pendigits	6870	16	2.2707
Pima	768	8	34.8958
Satellite	6435	36	31.6395
satimage-2	5803	36	1.2235

4.2 Metric Parameter Evaluation

This section presents some metrics used to evaluate the performance of outlier detection methods. They are described as follows:

4.1.1 AUC-ROC curve

The area under the curve (AUC) is the probability curve and the *Receiver operating characteristic* (ROC). The degree of separability is the most important evaluation metric for outlier detection. It is said that the higher the AUC better the model in distinguishing anomalies as anomalies. The ROC curve is a popular performance evaluation. It is plotted with True positive rate (TPR), which is called recall against the false positive rate (FPR), where FPR is on the x-axis and TPR is on the y-axis [23].

True Positive Rate is defined as follows [23]:

$$\mathrm{TPR} = \mathrm{TP}/\left(\mathrm{TP} + \mathrm{FN}\right) \tag{4}$$

False Positive Rate is defined as follows [23]:

$$FPR = \mathrm{FP}/\left(\mathrm{FP}+\mathrm{TN}\right) \tag{5}$$

The excellent model has a reading near to 1. It means that it has a better measure of classification. The poor model has a reading near to 0. It means that the worst measure of classification. When AUC approximate to 1 is called an ideal model, and a value close to 0 means it reciprocates the result. It means that the model is predicting an inlier as an outlier and an outlier as an inlier. Table 3 presents the obtained results for evaluated ROC characteristics using different algorithms with different datasets.

Table 3. ROC comparison between different algorithms with different datasets

Algorithms / Datasets	ABOD	CBLOF	FB	HBOS	Iforest	KNN	LOF	LODA	LSCP	SOD
Cardio	0.684	0.8525	0.59	0.8584	0.9331	0.7842	0.5887	0.9107	0.7042	0.6825
Musk	0.2922	1	0.6465	0.9999	0.9883	0.8732	0.6668	0.9977	0.6166	0.8066
Pendigits	0.7197	0.9667	0.5189	0.9181	0.9468	0.774	0.5277	0.8371	0.583	0.744
Pima	0.6574	0.6413	0.6112	0.6844	0.6615	0.6912	0.6338	0.6664	0.6511	0.618
Satellite	0.57	0.7218	0.5389	0.7247	0.6643	0.6784	0.5396	0.5967	0.5485	0.6289
satimage-2	0.769	1	0.6224	0.9831	0.9961	0.9424	0.6054	0.9842	0.6153	0.7272

Fig. 4 presents the obtained results for evaluated ROC characteristic using different algorithms with different datasets.

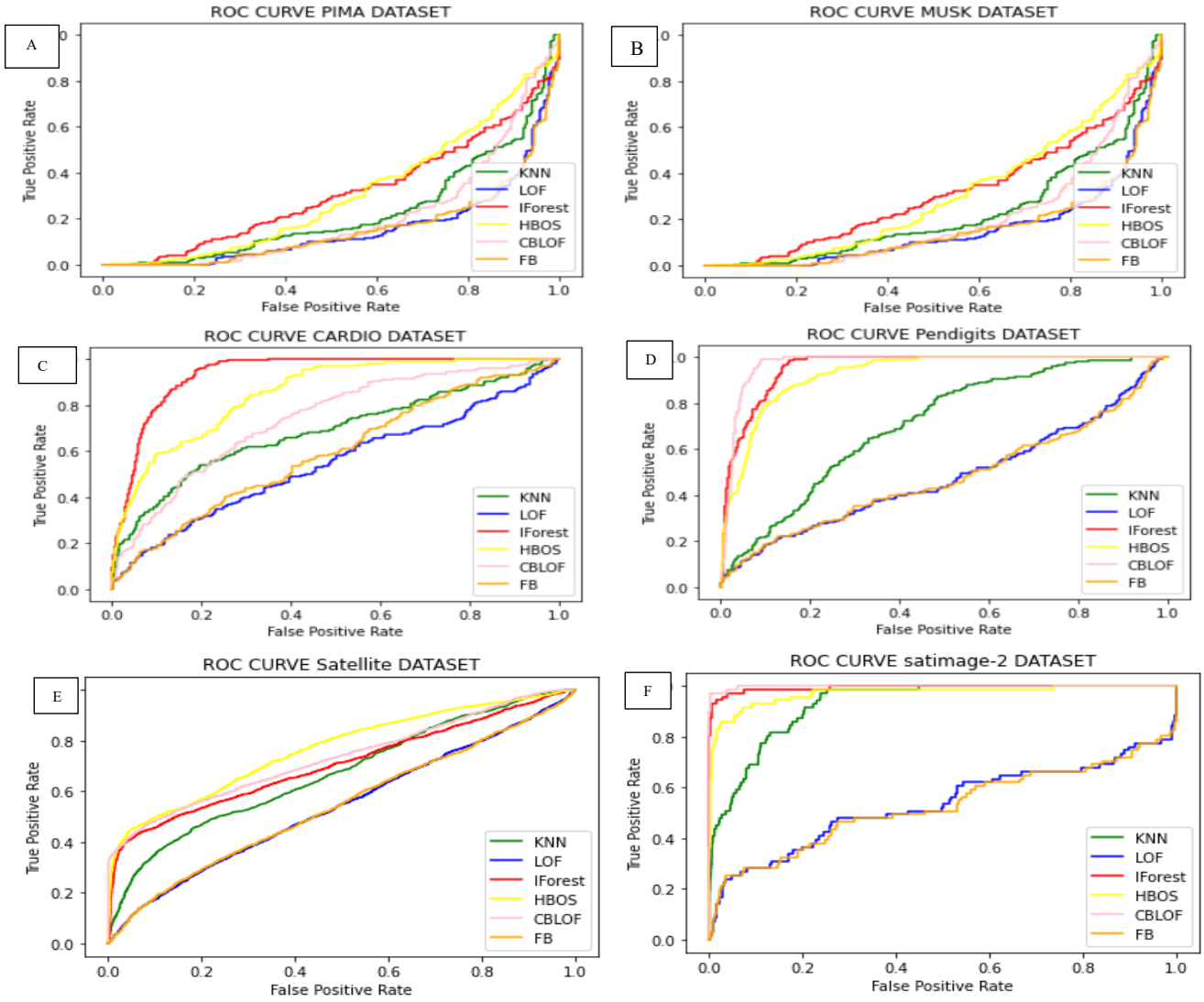

Fig. 4 Comparison ROC for different algorithms with different datasets[A:Pima dataset ,B:musk dataset, C:cardio dataset, D: Pendigits dataset, E: Satellite dataset, F: satimage-2 dataset]

4.1.2 Precision@ rank N:

Precision: is a proportion of numbering true outliers (m) over the numbering of outlier candidates (t).

$$\text{precision} = m/t \tag{6}$$

Average precision is the mean of precision scores through the ranks of all outlier points [24]. It is used in state evaluation only one value of n. The results obtained as shown in Table 3 and Fig. 5. The average precision of the compared algorithms is varied, as the ratio of outliers on these data sets is different. The precision is more sensitive to the value n on these datasets with a small outlier percentage and less sensitive on these datasets with a high outlier ratio. Table 4 presents the obtained results for evaluated average precision using different algorithms with different datasets.

Table 4. Average precision performance

Algorithms / Datasets	ABOD	CBLOF	FB	HBOS	Iforest	KNN	LOF	LODA	LSCP	SOD
cardio	0.3875	0.6625	0.3125	0.525	0.6125	0.425	0.3125	0.5625	0.275	0.35
musk	0.0488	1	0.439	0.9512	0.7073	0.4878	0.3171	0.878	0.2927	0.1463
pendigits	0.0893	0.3036	0.0714	0.2679	0.2857	0.1429	0.0714	0.1786	0.0893	0.1071
pima	0.4727	0.4818	0.4455	0.5273	0.4818	0.5182	0.4455	0.4909	0.4909	0.4455
satellite	0.384	0.5521	0.373	0.5374	0.5472	0.4871	0.373	0.4945	0.3853	0.4491
satimage-2	0.1364	0.9545	0	0.5909	0.8182	0.4091	0	0.6818	0	0.3636

Fig. 5 presents the obtained results for evaluated average precision using different algorithms with different datasets.

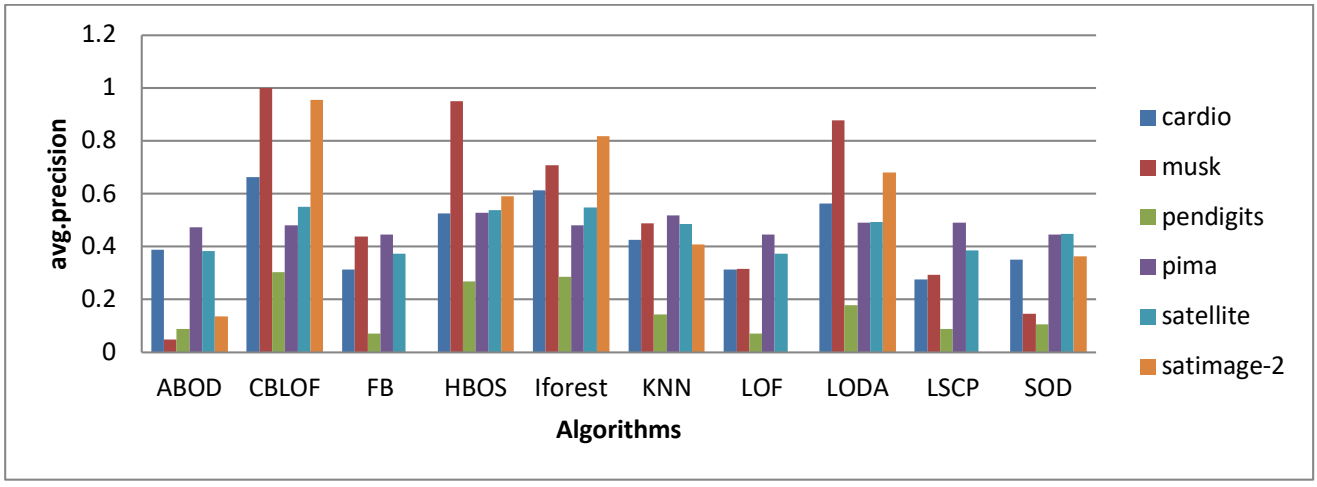

Fig. 5. Average precision Comparison between different algorithms with different datasets

4.1.3 Execution time:

Time consumption on experiment result of outlier detection algorithm on different size of dataset gives us the clear performance of the algorithm. When an increasing rate of the execution time shows that a particular algorithm is not much appropriate for a large dataset with high dimensionality. As shown in Table 4 and Fig. 5, the time elapsed in seconds (average of 10 independent trials) the results obtained for execution time are presented.

. Table 5.Execution time for different algorithms with different datasets

Algorithms / Datasets	ABOD	CBLOF	FB	HBOS	Iforest	KNN	LOF	LODA	LSCP	SOD
cardio	0.5541	0.1875	1.0319	0.016	0.7759	0.264	0.152	0.056	6.1113	1.8947
musk	2.5317	0.3256	13.1314	0.09	1.6359	1.9749	1.7916	0.06	59.2833	6.5795
pendigits	2.0708	0.2737	4.7124	0.01	0.9243	0.7912	0.699	0.06	29.7414	22.0847
pima	0.18	0.1313	0.1247	0	0.429	0.04	0.02	0.03	1.4368	0.417
satellite	2.5087	0.4245	8.5197	0.03	1.2253	1.3068	1.1498	0.06	43.3043	20.1228
satimage-2	2.165	0.3815	6.5931	0.03	0.9973	1.2337	0.9324	0.0605	37.9726	16.3932

Fig. 6 presents the obtained results for evaluated time execution using different algorithms with different datasets.

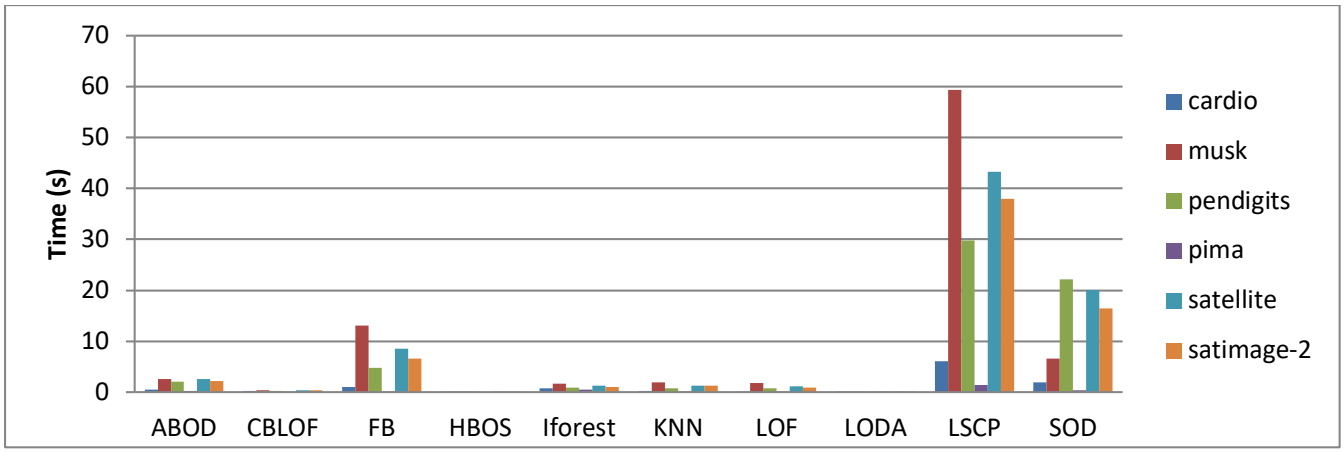

Fig. 6. Execution time for different algorithms using different datasets.

5. Results analysis and decision

This section from table 3 and fig. are represented the RECEIVER OPERATING CHARACTERISTIC (ROC) values that were used to evaluate the performance of outlier detection methods by computing the area under the curve. We note that the isolation forest technique is the fastest performing approach for its performance. Most outlier detection algorithms seek to build a profile for normal objects, then identify objects that do not conform to the normal profile as outliers. It identifies outliers by isolating them in the data. It requires less memory requirement compared to other outlier detection algorithms. Histogram-based methods (HBOS) for analysis of the single feature are more effective, but multi or high-dimensional data lose much of the effectiveness as they cannot analyze multiple features simultaneously.

The KNN method determines the right number of neighbors. determining the value of parameter k is more significant. Ensemble-based learning that combines more methods gets a better result. The approaches with a large range of subspaces or base learners perform as LODA (lightweight online detector of anomalies). So, it is more important to determine how to choose accurate subspaces and base learners. Angle Based outlier detection (ABOD) is less performance because it calculates the similarity of the objects by computing the cosine value of the angle. LOF (local outlier factor) requires computation for all objects in the dataset. CBLOF is good performance since combining distance-based algorithms with the clustering method can improve the model and result. Table 4 and Fig. 5 are represented the precision. The accuracy of the surface dimensionality estimation method describes how close repeated measurements are to each other. A completely accurate method will provide the same estimate every time it is used on the same surface, regardless of whether it is accurate. Some measurements are used to classify the relative accuracy of different methods. It can be observed that CBLOF, IForest, and HBOS, similar to AUC, had stable performance. From table 5 and Fig. 6 are represented the time of execution. It is the time consumption of the experiment result of the outlier detection algorithm. We observed that HBOS takes less execution time than other outlier detection methods.

6. Analysis and recommendation

In this study, different outlier detection approaches are described and analyzed. A comprehensive performance study does evaluation of most outlier detection algorithms. The outcome of various evaluation parameters are analyzed, i.e., AUC, precision, and execution time. The average value from all the above tables in Table 3, Table 4, and Table 5 are calculated and arranged the algorithms from best to worst, as shown in Table 6. When applying all methods to six benchmark datasets, the experimental results demonstrate that IForest and CBLOF perform high performance. HBOS is the fastest algorithm as it takes minimum time in execution. Cluster-based LOF (CBLOF) is better than the KNN approach. Feature bagging algorithm and Local outlier factor is the worst algorithm comparatively, but later is still better than LSCP in time complexity. Table 6 presents the obtained results for evaluated average (ROC, average precision, and execution time) metrics using different algorithms with different datasets.

Table 6. AVERAGE OF PERFORMANCE MEASUREMENT

Algorithms	Performance management		
	ROC curve	**Average precision**	**Execution time (s)**
IFOREST	0.86501	0.57545	0.99795
CBLOF	0.86371	0.65908	0.28735
HBOS	0.86143	0.56661	0.02933
LODA	0.83213	0.547716	0.0544
KNN	0.79056	0.41168	0.9351
SOD	0.7012	0.30806	11.248
LSCP	0.61978	0.25553	24.552
ABOD	0.61538	0.25311	1.3865
LOF	0.59366	0.25325	0.7908
FB	0.58798	0.27356	5.6855

7. Conclusion

In this study, outlier detection techniques are concerned. These Techniques allow many of the existing systems to have consistent performance (e.g. IoT, crowdsourcing, data aggregation, etc.). It allows such systems to detect and avoid erroneous and disrupted data. Specifically, machine learning approaches are studied, compared, and analyzed. This allows defining the ones that have the best accuracy for detecting

outliers; moreover to give researchers a point of view for the algorithms that can be combined to develop a new ensemble approach. Hybrid and ensemble approaches are being vastly used. They provide a better result and overcome the drawbacks of traditional techniques. In addition, deep learning-based techniques are recently used for detecting outliers. These approaches have also been approved to achieve better accuracy over traditional techniques. Therefore, in the future, a new approach that depends on both deep learning and ensemble approaches are going to be developed aiming to achieve accurate result compared with the existing approach.

References

[1] H. Wang, M. J. Bah, and M. Hammad, "Progress in Outlier Detection Techniques: A Survey," IEEE Access, vol. 7, pp. 107964–108000, 2019.

[2] A. Castillo and A. D. Thierer, "Projecting the Growth and Economic Impact of the Internet of Things," SSRN Electronic Journal, 2015.

[3] Ashton K."Internet of Things". https://www.rfidjournal.com/that-internet-of-things-thing (22 June 2009).

[4] S. Li, L. D. Xu, and S. Zhao, "5G Internet of Things: A survey," Journal of Industrial Information Integration, vol. 10, pp. 1–9, Jun. 2018.

[5] L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[6] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges," in 2012 10th International Conference on Frontiers of Information Technology (FIT 2012), Islamabad, Pakistan, Dec. 17–19, 2012. IEEE, 2012.

[7] M. Ahmed and A. Naser, "A novel approach for outlier detection and clustering improvement," in 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA 2013), Melbourne, VIC, Jun. 19–21, 2013. IEEE, 2013.

[8] M. Goldstein and S. Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data," PLOS ONE, vol. 11, no. 4, Apr. 2016, Art. no. e0152173.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection," in Encyclopedia of Machine Learning and Data Mining, Boston, MA: Springer US, pp. 1–15, 2016.

[10] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Computers & Security, vol. 28, no. 1-2, pp. 18–28, Feb. 2009.

[11] H.-P. Kriegel, M. S hubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in the 14th ACM SIGKDD international conference, Las Vegas, Nevada, USA, Aug. 24–27, 2008. New York, New York, USA: ACM Press, 2008.

[12] Breunig, Markus M., et al. "LOF: identifying density-based local outliers." Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000.

[13] Goldstein, Markus, and Andreas Dengel. "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm." KI-2012: Poster and Demo Track (2012): 59-63.

[14] V. M. Tellis and D. J. D'Souza, "Detecting Anomalies in Data Stream Using Efficient Techniques: A Review," in 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), Kannur, Mar. 23–24, 2018. IEEE, 2018.

[15] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," Pattern Recognition Letters, vol. 24, no. 9-10, pp. 1641–1650, Jun. 2003.

[16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in 2008 Eighth IEEE International Conference on Data Mining (ICDM), Pisa, Italy, Dec. 15–19, 2008. IEEE, 2008.

[17] Janssens, J. H. M., et al. "Stochastic outlier selection." Tilburg centre for Creative Computing, techreport 2012-001 (2012).

[18] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in Proceeding of the eleventh ACM SIGKDD international conference, Chicago, Illinois, USA, Aug. 21–24, 2005. New York, New York, USA: ACM Press, 2005.

[19] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki, and Z. Li, "LSCP: Locally Selective Combination in Parallel Outlier Ensembles," in Proceedings of the 2019 SIAM International Conference on Data Mining, Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 585–593, 2019.

[20] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data," in Advances in Knowledge Discovery and Data Mining, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 831–838 , 2009.

[21] T. Pevný, "Loda: Lightweight on-line detector of anomalies," Machine Learning, vol. 102, no. 2, pp. 275–304, Jul. 2015.

[22] Zhao, Yue, Zain Nasrullah, and Zheng Li. "Pyod: A python toolbox for scalable outlier detection." arXiv preprint arXiv: 1901.01588 (2019).

[23] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," Statistical Analysis and Data Mining, vol. 5, no. 5, pp. 363–387, Aug. 2012.

[24] N. Craswell, "R-precision, " in Encyclopaedia of Database Systems,L. Liu and M. Ozsu, Eds., Springer, Berlin, Germany, pp. 2453, 2009.