A Clustering Approach for Outliers Detection in a Big Point-of-Sales Database

Yoseph, Fahed; Heikkilä, Markku

Published in:
2019 International Conference on Machine Learning and Data Engineering (iCMLDE)

DOI:
10.1109/iCMLDE49015.2019.00023

Published: 01/01/2019

Document Version
Accepted author manuscript

Document License
Publisher rights policy

A Clustering Approach for Outliers Detection in a Big Point-of-Sales Database

Fahed Yoseph, Markku Heikkilä

Åbo Akademi University,

Turku, Finland

***Abstract* - Finding outliers, rare events from a collection of patterns, has become an emerging issue in the area of machine learning concerned with detecting and eventually removing anomalous objects in data. A key challenge with outliers/anomalies detection is because they are not a well-formulated issue. Outliers are defined as the extreme values that deviate from the overall patterns in data; they may indicate experimental errors, variability in measurement, or a novelty. Detecting outliers in large databases can lead to the discovery of hidden knowledge. However, identifying and removing outliers often helps to ensure that the observations represent the problem correctly. Though there are several techniques for detecting outliers/anomalies in a given database, thus, no single technique is proven to be the standard universal choice. Depending on the nature of the target application, different implementations require the use of different outlier detection methods. The clustering method is a very powerful method in the field of machine learning and defines outliers in terms of their distance to the cluster centers. In this study, we propose a clustering-based approach to identifying outliers in a retail point-of-sales dataset. To select the best clustering algorithm for the purpose, two algorithms are applied, K-means for hard, crisp clustering, and (FCM) Fuzzy C-means for soft clustering. The experimental results show that the K-means algorithm outperforms the (FCM) Fuzzy C-means algorithm in terms of outlier detection efficiency, and it is an effective outlier detection solution.**

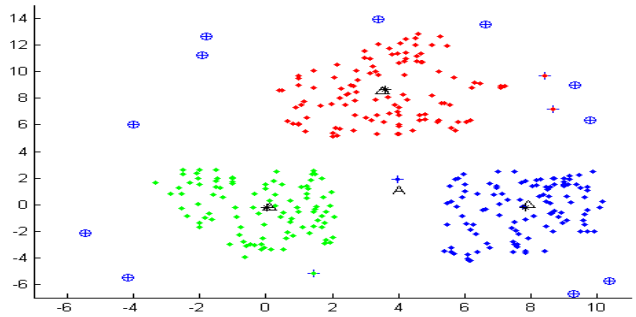
Keywords - Outliers Detection, Noise, Clustering Method, Point-of-sales

I.ANOMALY DETECTION

Detection of outliers or anomalies is known with many names, such as novelty detection, anomaly detection, change point detection, or intrusion detection. It is considered as one of the core problems in machine learning, gaining considerable attention in many industrial and financial applications, like retailing and banking [1]. The emerging expansion and growth of data have made scientists rethink the way anomalies should be approached. Outliers are known as the abnormal data objects having different behavior than the normal objects in the dataset, often caused by execution or measurement error [1]. Hawkins [1] defines outlier in the database as an observation that deviates significantly from the majority of patterns and stirs suspicion that a different mechanism generated the outlier.

Generally, outliers/anomalies detection singles out those objects, which are significantly deviating from the rest of the dataset to the extent that they seem to be generated by another process [15]. Outlier tends to bias statistical estimators, and they are the result of many artificial intelligence methods like the standard deviation, the mean value, or the position of the prototypes of k-means algorithms is affected [2][8]. Several data-mining methods have been used to identify the deviation of a data point from other data points [16]. However, data-mining methods have so far only been used in small datasets. Christy [35] notes that statistical estimates like the standard deviation and arithmetic mean might be affected by the spread of the data-points that lie far from the middle of the data distribution.

Researchers have argued that some of the statistical methods like Gaussian theory, are too time consuming in detecting outliers/anomalies in large datasets [35]. Supervised and unsupervised learning methods are the two distinct fundamental machine learning approaches to the problem of anomalies/outliers detection. The Supervised learning method builds a model on some norm, such as labeled outliers types, and detect deviations in observed (labeled) data from the normal model. The advantage of a supervised anomaly/outlier detection method is that it detects new types of activity as deviations from normal usage [28]. In contrast, an unsupervised method learns hidden structures from unlabeled data, and identifies outliers/anomalies without the use of prior knowledge of the data and requires that the analyst or decision-maker has the capacity to interpret the obtained results correctly and use them to make the right decisions. Outlier detection methods are often used for finding sudden or unexpected changes in data in the early stages of the analysis process [28]. Unsupervised learning, such as clustering algorithms, uses unlabeled input data and allows the algorithm to act on that information without guidance (training). In clustering, the task is to assign a set of



data objects into groups (clusters) so that the data objects in the cluster are in some sense more like each other, and different from data objects in other groups (clusters) [27].

Fig. 1. Sample of Outliers detection using the clustering method

I. RELATED STUDIES

Borgelt [26] stated that data-mining main objective is the discovery of valuable knowledge from a large database, identifying hidden patterns, and applying discovery algorithms to develop a data-mining model [22], [24]. Data-mining models help business executives hypothesize about the data and use it to make guided and better-informed decisions [25], [34]. Verification and discovery are the two primary goals process. With the verification process, the end-user's hypothesis about the data is verified, while the discovery

process aims at automating the process of finding unknown patterns [23], [25].

Outliers come in different flavors, depending on the environment: collective outliers, point outliers, or contextual outliers, but generally, there are two types, univariate outlier, and multivariate outlier. A vast number of supervised, and unsupervised, and semi-supervised clustering algorithms have been used for outliers/anomalies detection. These clustering algorithms are further classified into cluster based, classification-based, nearest-neighbor–based, density-based, information theory-based, signal processing-based v and visualization-based [25], [29].

Understanding the outlier type will significantly affect the way to approach anomalies. A univariate outlier is a data point with an extreme value of the variable. Univariate outliers are often found when the distribution of values is viewed in a single space. A multivariate outlier, on the other hand, is a combination of unusual values on at least two independent or dependent variables found in n-dimensional spaces. From the viewpoint of clustering, the outlier is a by product of the clustering algorithms and are identified as data objects not located within clusters of the dataset, but instead, can be defined as noise [5].

Many data-mining algorithms observe patterns and relationships. One of the most important purposes in finding outliers in large databases is to highlight critical information as non-outliers. A faulty signal can result in the loss of critical information value. On the other hand, one entity's noise could be another entity's signal [31]. Many times, such as in the case of fraud detection, the outliers themselves are of particular interest, when outliers may indicate fraudulent activity [30],[32].

One of the approaches to find critical outliers information is from Zhang, Hutter, and Jin [15], who proposed (LDOF) the Local Clustering Distance-based Outlier/anomaly to measure the outliers data points in a scattered dataset. (LDOF) the approach is to the relativity of the location of the data objects to its neighbors to identify the deviation degree of data objects from its neighborhood. The top-n LDOF facilitates finding parameter settings in real-world applications by reporting only objects (data points) with the highest (top-n) LDOF values as outliers. In scattered datasets, the top-n LDOF outperforms other similar approaches such as top-n KNN and top-n LOF. D'Urso et al. [9] proposed outlier detection with a framework of partitioning data with the Fuzzy C-means clustering algorithm.

The clustering algorithm is considered one of the most prominent and reliable methods in statistics and machine learning because it is applicable to many natural phenomena to help understand and visualize the shape of data [10], [4]. According to Lefait and Kechadi [3], and Honda [10], clustering algorithm aims to create groups (clusters) of similar data objects in a way that the data objects belonging to the same cluster are very similar to each and different or very different from those belonging to different groups (clusters) are dissimilar. MacQueen [12] proposed k-means clustering (*KM*) as simple and fast learning algorithm to develop appropriate forecasting and decision planning, with the objective to find useful structures and patterns from the database or to systematically guide researchers and scholars to choose appropriate unsupervised machine learning method. Fuzzy C-means (*FCM*) is the method often used in machine learning, combining partitioning and imprecise, fuzzy memberships of data points to one or more clusters [41], [7]. In contrast to KM, where partitioning to clusters is hard, and a data point only belongs to one cluster, FCM applies soft partitions allowing memberships in several clusters. According to Bezdek, Ehrlich, and Full [36], fuzzy partitions let outliers be detected in the intersection of clusters because each data point is "grouped unequivocally with its intracluster neighbors."

Also, other clustering methods like DBSCAN [19], CURE [20], and BIRCH [21] may detect outliers. However, they are proposed to optimize clustering methods, but not outlier/anomaly detection.

IV. The Proposed Model

One of the main objectives of any clustering algorithm method is to discover patterns in high dimensional databases, cluster data objects with patterns similarity together, which will lead to the reduction of complexity. Our proposed method is to use clustering to find outliers and anomalies in large point-of-sales (POS) dataset. Python, as one of the most powerful general-purpose programming languages, offers extensive coverage of libraries for data analysis, artificial intelligence, and scientific computing. In this study, we use common Python libraries (sklearn, Pandas, Numpy, Scipy, Plot Lib), KM module from scipy.cluster.vq. and Matplot to develop a cluster model for outliers detection.

Our POS dataset contains over 5 million transactions provided by a retailer in Kuwait. We analyze the dataset with two clustering algorithms, KM, and FCM, to find outliers by means of unsupervised learning. The algorithms are discussed in detail next.

```
#Load the required packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#Plot styling
import seaborn as sns; sns.set()  # for plot styling
%matplotlib inline
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
#Read the Excel (.xls) file
dataset=pd.read_excel('C:/Users/Tahir/Desktop/ML_Project1/POS.xls')
#Explore the dataset
dataset.head()#top 5 columns
len(dataset) # of rows
#descriptive statistics of the dataset
dataset.describe().transpose()
print("Data importing")
```

	count	mean	std	min	25%	50%	75%	max
HOUSEHOLD_KEY	10001.0	1.329600e+03	7.441180e+02	7.000000e+00	6.870000e+02	1.366000e+03	2.009000e+03	2.493000e+03
BASKET_ID	10001.0	2.714554e+10	7.973639e+07	2.698485e+10	2.709323e+10	2.715354e+10	2.721309e+10	2.728248e+10
DAY	10001.0	1.433597e+01	6.574946e+00	1.000000e+00	1.000000e+01	1.500000e+01	2.000000e+01	2.500000e+01
PRODUCT_ID	10001.0	1.453294e+06	1.674609e+06	3.315000e+04	9.071280e+05	9.952420e+05	1.083683e+06	9.885360e+06
SALES	10001.0	1.057286e+02	1.666687e+02	0.000000e+00	3.897500e+01	7.565000e+01	1.041250e+02	5.100000e+03
QTY	10001.0	1.304370e+00	8.348353e-01	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	2.000000e+01
STORE_ID	10001.0	2.864174e+03	8.180849e+03	2.700000e+01	3.240000e+02	3.670000e+02	4.100000e+02	3.200400e+04
RETAIL_DISC	10001.0	1.500000e-01	1.793100e-14	1.500000e-01	1.500000e-01	1.500000e-01	1.500000e-01	1.500000e-01
TRANS_TIME	10001.0	1.582079e+03	4.285271e+02	0.000000e+00	1.309000e+03	1.614000e+03	1.917000e+03	2.358000e+03
WEEK_NO	10001.0	2.727527e+00	9.949126e-01	1.000000e+00	2.000000e+00	3.000000e+00	4.000000e+00	4.000000e+00
COUPON_DISC	10001.0	-1.010799e-02	1.939101e-01	-1.800000e+01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
COUPON_MATCH_DISC	10001.0	-1.865813e-03	2.794241e-02	-5.000000e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
PRODUCT_CATEGORY_ID	10001.0	2.138986e+00	8.082282e-01	1.000000e+00	2.000000e+00	2.000000e+00	2.000000e+00	8.000000e+00
PRO_PRICE	10001.0	9.534594e+01	1.400720e+02	0.000000e+00	3.750000e+01	8.900000e+01	8.900000e+01	6.000000e+03

Fig. 2. Importing the required Python libraries and describing the experimented POS dataset

First, after importing all the required libraries, we read the sales dataset into Pandas data frame as a "dataset." Fig. 2 shows descriptive statistics about the structure of the dataset. This information is helpful for both setting up the analysis process and interpretation of the results.

3.1 Outliers Detection with K-means and Fuzzy C-means

3.2 K-means Algorithm (KM)

KM is a non-hierarchical, partitional, distance-based clustering algorithm and is primarily suitable for large databases commonly used in the field of marketing [15]. The algorithm has been extensively used in market (customer) segmentation and sales patterns recognition because of, simplicity in implementation, its stable performance, and fast execution [6], [16]. *k* represents the number of clusters (groups) chosen [13]. KM algorithm's main objective is to partition *n* objects into *k* clusters so that the inter-cluster similarity as the distance between observations is minimum and intra-cluster similarity is maximum. KM follows four standards steps listed below.

Step 1: Initialize the position of the clusters using random cluster centers.

Step 2: Each observation is assigned to the cluster with the shortest within-cluster sum of squares (WCSS). Where the sum of squares is the (squared Euclidean distance), this is intuitively the nearest mean. The Euclidean distance of observation, *x* in three dimensional coordinates, such as rececncy, frequency and monetary components of the classical RFM model in marketing, to a center, *c* is calculated as

$$d(x,c)=\sqrt{(x_r-c_r)^2+(x_f-c_f)^2+(x_m-c_m)^2}\,, \quad (1)$$

Where x_r, x_f, and x_m are normalized (RFM) recency, frequency and monetary values (scores) of observation *x*, and c_r, c_f, and c_m are corresponding coordinates of the cluster center *c*.

Step 3: KD updates centers, calculates new centers to the cluster as the n-dimensional centroid point amid *k* n-dimensional points,

$$CP(x_1,x_2,\ldots,x_k)=\left(\frac{\sum_{i=1}^{k} x1st}{k}\ \frac{\sum_{i=1}^{k} x2nd}{k},\ldots,\frac{\sum_{i=1}^{k} xnth_i}{k}\right). \quad (2)$$

Step 4: Steps 2 and 3 are repeated until the centers have converged, i.e., do not change. The centers have now become centroids of the *k* clusters. Convergence criteria is found by minimizing the sum of squared error (*SSE*) measures. To find the *SSE*, for each observation, *x*, the sum of the squared errors to the nearest cluster is calculated as in the equation

$$SSE=\sum_{i=1}^{k}\sum_{x\in C_j} dist^2\,(m_i, x_i), \quad (3)$$

where *x* represents a data point in group (cluster) c_i, and m_i is the representative point for group (cluster) c_i. The m_i corresponds to the center of the group (cluster) c_i.

3.2.1 Detecting Outliers Using K-means

Outliers adversely affect the quality of the clusters, particularly for KM analysis [14]. The best option to remove outliers using the KM is to identify outliers in the context of clusters and remove them. In Angiullis [17] model for outlier detection with KM the Steps 1-3 (above) have the following additions:

Step 1: In any KM iteration, after the data elements have been assigned to respective clusters, calculates the outlier score of all data points, where *k* is predefined.

Step 2: Define outlier score as the ratio of the distance of a data point to the centroid divided by the mean or median distance of all cluster members to the centroid of the cluster

Step 3: Either remove the data element with the highest outlier score or remove all the data elements with outlier scores above some threshold.

There are two general methods of detecting outliers in clustering (distance-based and cluster-based) [37], [38]. In this study, we use the cluster-based method to find outliers in POS data.

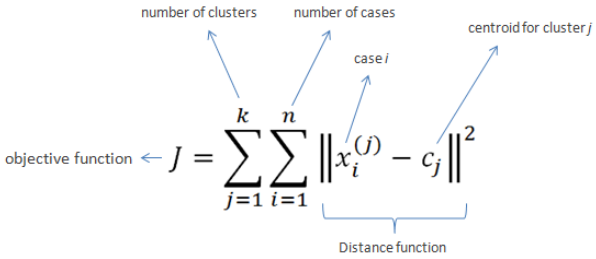


Figure 3. Components of the objective function of KM algorithm

In cluster-based outlier detection (*COD*), an outlier is found when a data point

1. does not belong to any cluster,

2. belongs to a very small cluster or

3. is forced to belong to a cluster where it is very different from other members.

COD techniques have been on the basis that outliers do not belong to any cluster since there are very few of them, and they lie far away from the normal instances. The KM algorithm for COD approach involves two-phase only [33] to divide the database into a pre-determined *k* number of clusters and calculated accuracy and outline index.

First phase: Find the size of the cluster, the smallest cluster is considered as a potential outliers cluster. Anomalies or outliers are always appearing in very small numbers or sometimes are forced to belong to a particular cluster.

Second phase: Once the possible outliers clusters are detected, start removing these clusters, or data points far away from big clusters. Next, run the KM algorithm and again calculate the accuracy and the outlier score.

Our first approach with KM algorithm-based COD is to cluster the dataset with $k = 4$. We assume that outliers are more visible the more clusters are used.

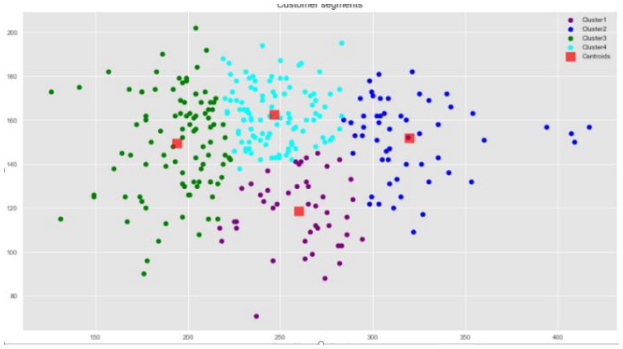


Fig. 4. K-means algorithm distribution of four clusters with two features.

Figure 4 depicts a scatter plot visualization of four distributed clusters generated by the KM algorithm-based COD. The four red squares represent cluster centers. It is apparent that outliers (an observation point that is distant

from other observations within the cluster) can be easily pointed out where the number of data points from all four clusters deviate largely from the proximity of the data points to the other data points in the dataset. However, the outlier scores vary from one cluster to another.

We have now the first cluster iteration and corresponding cluster centers. Next, we update the model with four clusters and fit the new points into the model for identifying outliers.

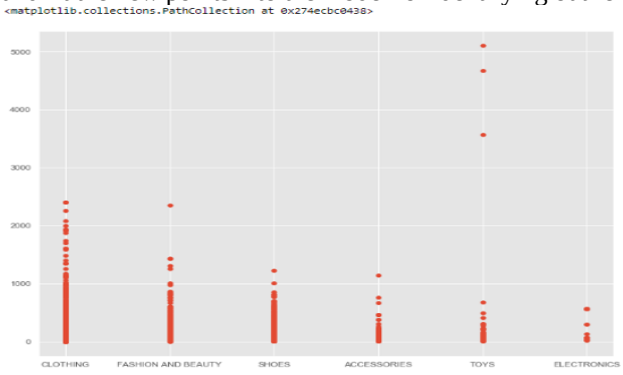


Fig. 6. KM outlier detection

Figure 6 depicts all retailing product categories (Toys, Fashion, and Beauty, Accessories product categories) to visualize potential outliers in each category. For example, when we look into the sales of TOYS in POS dataset, we can confirm that maximum sales prices lie beneath the 400 KD. That means prices, higher than 400 KD, can be considered potential outliers in the dataset. Similarly, we can look into other product categories is POS dataset to find outliers.

3.3 FCM Clustering Method

FCM is an unsupervised soft clustering method, initially developed in 1973 by Dunn [8]. In the FCM algorithm one data point to belong to two or more groups (clusters) with a fuzzy membership, using concepts from the field of fuzzy set theory and fuzzy logic. Assigning each data point *i* a membership to any of the *j = 1,..,k* clusters with membership value $\mu_j = [0,1]$ determines the degree of belonging to the cluster *j* [10], [17].

Accordingly, memberships of the data point *i* to *k* clusters are shown with vector u_{ij}. Allowing for membership in several clusters serves our purpose of finding outliers, as data point *i* with several $\mu_{ij} > 0$ may indicate outliers in the spaces between cluster centers. This adds a powerful detection capability compared to traditional hard-threshold clustering, such as KM, where every point is assigned a crisp, exact label *1* as the membership to the assigned cluster and 0 to all the rest of *k-1* clusters. More generally, a data point close to the cluster center has a high-degree of non-exclusive membership in that cluster and generally has lower memberships in other clusters, while for data point farther away from the cluster center the non-exclusive membership in the cluster in question is lower but may well have memberships in other clusters as well. However, the sum of the membership values of a data point to all clusters must be 1.

The FCM has been applied to a vast domain of applications [11], and it is frequently used in pattern recognition [18].

3.3.1 Detecting Outliers Using (FCM) Fuzzy C-means

The FCM algorithm starts with the same step as the KM, a randomly initials the cluster centers. Then FCM algorithm assigns all data points a random membership in each group (cluster). By iteratively updating the center of the cluster and the membership grades for every data point in the cluster, the algorithm then moves the centers of the cluster to more appropriate locations within a dataset and, for every data point, the FCM algorithm finds the degree of membership in each group (cluster). these types of iterations minimize the objective function which represents the distance from any given data point to the cluster center weighted by the membership of that data point in the cluster.

Fuzziness in the FCM algorithm is represented by the ability of any data point to have membership value to each cluster by means of fuzzy membership value, a number in the interval [0,1].

The objective function of the FCM minimizes the formula

$$J_m = \sum_{i-1}^{N} \sum_{j-1}^{C} u_{ij}^{m} \left\| X_i - C_j \right\|^2, 1 \le n < \infty \qquad (4)$$

The *N* represents the number of data points; *C* represents the number of groups (clusters). *m* represents the fuzzy partition matrix exponent for controlling the degree of fuzzy overlap, with *m > 1*.

Fuzzy overlap refers to how fuzzy the boundaries between groups (clusters) are, that is, the number of data point that has significant membership in more than one cluster. x_i represents the ith data point. c_j represents the center of the jth cluster. Next; the cluster center of cluster *j* is

$$\mathrm{c_j} = \frac{\sum_{i=1}^{D} \mu_{ij}^{m} x_j}{\sum_{i=1}^{D} \mu_{ij}^{m}} \qquad (5)$$

where μ_{ij} represents the membership degree of x_i in the jth (group) cluster.

The FCM algorithm follows steps (1-5)

1. Then the FCM randomly initializes the group (cluster) membership values, μ_{ij}.
2. Calculate the cluster centers.

$$\boldsymbol{\mu_{ji}} = \frac{1}{\sum_{k=1}^{N} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_j - c_k \right\|} \right)^{\frac{2}{(m-1)}}} \qquad (6)$$

3. Update u_{ij} according to the following
4. The (FCM) calculates the objective function, J_m.
5. steps (2–4) are repeated until J_m improves after a specified maximum number of iterations, or by less than the specified minimum thresholds.

In our example, we have clustered the dataset into three clusters. Figure. 3 shows our first distribution of three randomly generated clusters using FCM algorithm.

While dealing with transactional data anomalies, one key aspect is to further examine the outliers in various contexts, taking into consideration that transactional data are always domain-specific.

Next, the FCM is executed to generate a random number of clusters.

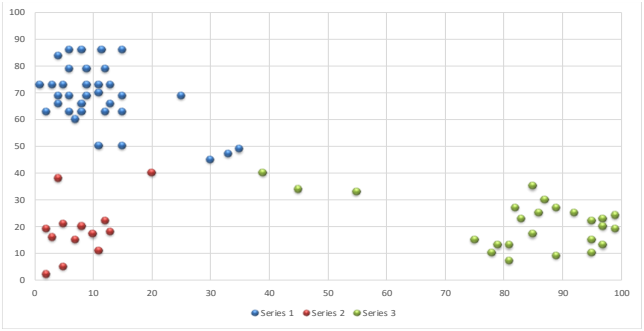


Fig. 5. Distribution of three clusters using FCM Algorithm.

Figure 5 depicts a scatter plot visualization with three clusters randomly generated by the FCM algorithm. The data values have fallen inside the normal cluster bounds, but there are indeed abnormal points (far from the cluster) when these points are compared to the rest of the data points. We can see that the blue-colored cluster is relatively having more outliers compared to the green and orange colored clusters. We can see there is a number of points deviating from the rest of the points (data). Therefore, the immediate and logical way to infer that the dataset contains outliers (anomalies), is to look at the minimum and maximum data values of the given data.

We have identified with each new data point to which cluster belongs. Next, we develop an outlier detection model with the three clusters and fit new points into the model for identifying outliers in our POS dataset using FCM algorithm.

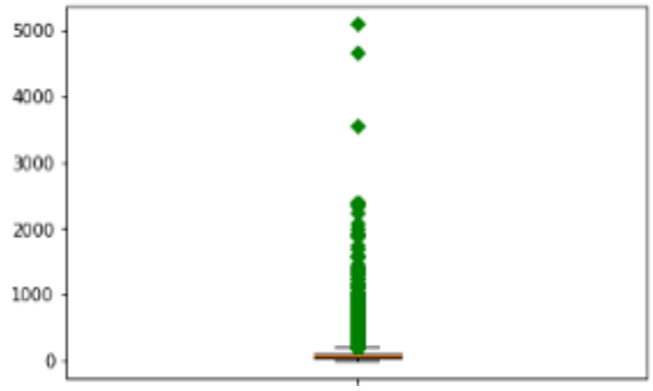


Fig. 6. Potential outliers in data

In the boxplot of figure 4, green dots above 3000 represent potential outliers in POS dataset with prices that are higher than 3000 KD and can be considered as noise/outliers. In this case, 3000 KD is a scaling mistake, which is 300 price value in the SALES column, and more than that price can be considered as outliers in the dataset.

V. Comparative Analysis and Result (Performance and Accuracy)

In this study, we use a real POS dataset obtained from a local retailer, Fig 7. illustrate a sample of the experimented POS dataset. In this section, we apply KM and FCM clustering algorithms to compare their performance by calculating the time complexity analysis in the identification of anomalies/outliers in clustering analysis.

Two clustering algorithms might have the same time complexity, for example, O(n^2), but one algorithm may take twice or more running time as the other algorithm.

Time complexity is a method of expressing the performance of an algorithm by quantifying the amount of time taken by the algorithm to run as a function of the length of the input [39]. It is an asymptotic behavior description of running time as input size approaches infinity. Thus, the time taken by the algorithm to compute an issue of size n is in the set of functions denoted by O(n) [40].

Trax_id	PRODUCT_ID	PRODUCT_CATEGORY	SALES	QTY	STORE_ID	RETAIL_DISC
26985360571	855,325	CLOTHING	76	1	324	0.15
27008831408	855,468	FASHION AND BEAUTY	92	4	324	0.15
27008831439	856,515	CLOTHING	76	1	324	0.15
26985365821	870,826	CLOTHING	7	1	324	0.15
26985360571	1,043,590	SHOES	76	1	324	0.15
26996870743	1,015,048	ACCESSORIES	76	1	324	0.15

Fig 7. Sample of the POS dataset

The analysis uses five industry performance measurements namely time complexity O(ncdi), number of outliers (data points) detected by each algorithm, (time complexity) when the number of iterations is varying, time complexity when the number of clusters is varying, and the elapsed time required by each algorithm. Next, the results of our comparative analysis are shown in the below Tables and Figures. The time complexity of KM algorithm is O(ncdi), where O(ndc2i) is the time complexity of the Fuzzy c-means algorithm. Thus, keeping the number of data points constant, we can conclude that (n = 100, d = 3, i = 20) and the varying number of clusters.

Time Complexity answers to the question: How does the runtime (the time it takes to execute a piece of code) of the function grow with respect to the number of data points (the input data size) *n*, the number of clusters *c*, the number of dimensions *d*, and the number of iteration each algorithm takes *i*. Therefore, the time taken to compute the size *n* is in the set of functions denoted by *O(n)*. Outlier scores are used to count the number of outliers detected by the algorithm. The higher the scores are, the more abnormal the data point is. The score indicates the overall abnormality of the outlier in the dataset. The descriptive comparative analysis is applied to the dataset and is shown in Tables 1, 2, and 4; and in Figure 8.

Fig. 7 illustrates the point-of-sales (POS) database used in the experiment. The Trax_id indicates the unique transaction number. The Product_id is the unique product code in the POS database. Product_Category represents the category each product belongs to. Sale and Qty represent the total amount and the quantity of each invoice, respectively. Store_id represents the store location. Retail_DISC represents the possible discount given on the product.

TABLE 1. TIME COMPLEXITY COMPARATIVE ANALYSIS BETWEEN KM AND FCM ALGORITHMS

Algorithm	**Time Complexity**	**Elapsed Time (Seconds)**
KM	O(ncdi)	0.443755
FCM	O(ndc²i)	0.781679

In Table 1. The time complexity (*TC*) was examined with a matrix to evaluate KM and FCM algorithms. From the analysis results in Table 1, we can observe that the TC with KM, 0.443755, is low and relatively stable, while the TC with FCM is higher, 0.781679.

TABLE 2. TIME COMPLEXITY OF KM AND FCM ALGORITHMS WHEN THE NUMBER OF ITERATIONS IS VARYING

Cluster No.	Number of Iterations	KM (Time Complexity)	Fuzzy C-Means (Time Complexity)
C1	5	3000	6000
C2	10	6000	12000
C3	15	9000	18000
C4	20	12000	24000

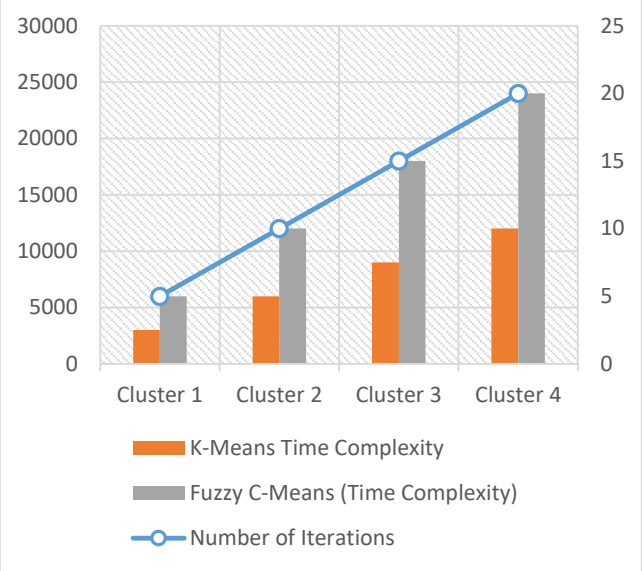


Fig 8: Time complexity of KM and FCM algorithms by a varying number of iterations.

Table 2 and Figure 7 show the TC of the algorithms with a various number of clusters. It takes more time for the FCM to calculate outliers than the KM conclusion, the TC decreases in all different iterations after applying the KM faster than with FCM.

TABLE 3: TIME COMPLEXITY OF KM AND (FCM) ALGORITHMS WHEN THE NUMBER OF CLUSTERS/ANOMALIES IS VARYING

Cluster No.	Number of Clusters	KM Time Complexity	FCM (Time Complexity)
C1	1	6000	6000
C2	2	12000	12000
C3	3	18000	54000
C4	4	24000	96000

Table 3 shows the total number of TC by each algorithm when the number of clusters is varying. The results conclude that the TC still decreases in all different iterations after applying the KM algorithm as compared to the FCM.

Both KM and FCM algorithms heavily rely on the number of neighbors *k*. To further illustrate the performance impact of *k*, we performed additional comparison experiments among the two algorithms in Table 4 that summarize the number of outliers/anomalies detected by KM and FCM algorithms.

TABLE 4. NUMBER OF OUTLIER/ANOMALY DETECTED BY KM AND FCM

Number of Clusters	Number of Outliers Detected in KM	Number of Outliers Detected in FCM
C1	1170	907
C2	375	252
C3	197	195
C4	781	403

The values of the KM and FCM algorithms vary significantly in terms of their sensitivity to the outliers (Table 4). The total number of the detected outliers is calculated in each of the four clusters, and then manually revised by the client to investigate the degree and the seriousness of the outliers. Some of the discovered outliers were found to be a result of excessive repeated small transactions. From the result, it is found that the number of outliers detected using KM increases with the increase of the number of iteration, whereas with FCM the increase of the number of iteration had less impact on the outlier detection.

TABLE 5. TOTAL OF OUTLIER/ANOMALY DETECTED AND ERROR RATE BY KM AND (FCM)

Algorithm	Number of Outliers Detected	Error (%)	
KM Algorithm	2523	0.52%	13
FCM Algorithm	1757	2.68%	47

As shown in Table 5, the KM algorithm correctly identified 2523 outliers, as compare to 1757 outliers of the FCM. The KM algorithm is faster in the identification of outliers compared to FCM, because the algorithm uses the robust mean in computing. Also, KM has obvious advantages and far more efficient in minimizing errors, which are 0.26 compared to 0.97. However, the data objects filtered as errors should be further investigated by careful human experts analysts, if they should be flagged as outliers, or manually removed.

TABLE 6. ELAPSED TIME IN SECONDS

KM Algorithm	FCM Algorithm
2.27246	4.19491

Table 5 depicts the elapsed time in seconds required for the KM COD to discover the outliers inside the clusters. As seen, the time is a lot shorter than the FCM algorithm.

VI. CONCLUSION

The objectives of this paper are to improve the quality of outlier detection by comparing various clustering algorithms. Outlier patterns in the traditional clustering algorithms are either incorporated by larger patterns or neglected systematically. However, detection of these anomalies or outliers from a large dataset is often critical.

This study proposed an efficient outlier detection model based on clustering. The main advantage of our techniques is

the continuous monitoring of potential outliers-generating data process. KM COD and FCM clustering algorithms are used to partition the dataset into a number of clusters. Generally, in KM algorithm, each data point belongs to only one cluster, wherein FCM algorithm, one data point might belong to one cluster or more clusters, therefore, we found it difficult to identify outliers in POS dataset using FCM even with the help of domain experts to confirm the authenticity of the discovered potential outliers. We conclude that the numbers of outliers in a dataset are generally very few compared to the size of the experimented point-of-sales (POS) dataset used in this study. Maimon [40] stated even with a small proportion of anomalies /outliers; this can seriously distort the numerical summary of the database [40].

From the obtained analysis results, we conclude that the KM algorithm using cluster-based outliers detection is superior to the FCM algorithm in detecting outliers fast in large POS datasets. Although FCM produces results close to the results of KM. FCM requires more computation time than the KM because of the way the FCM algorithm measures calculations involvement. In computing outlier scores for each cluster, FCM is computationally expensive, whereas for KM computation complexity and cost are less expensive. The task of detecting outliers is often manual, time-consuming, and constrained by limited resources. Our outlier detection model addresses these issues by identifying data points with largest dissimilarities to the data, and with a higher probability of being outliers.

different clustering algorithms, such as dbscan, EM, and other prototype-based clustering algorithms, may be implemented to evaluate the performance differences in identifying outliers and noise in big Point of Sales database.

REFERENCES

[1]. Hawkins D (1980). Identification of outliers. Chapman and Hall, London.

[2]. Estivill-Castro V, Yang J (2004). Fast and robust general-purpose clustering algorithms. Data-mining and Knowledge Discovery 8: 127. –150, Kluwer Academic Publishers, Netherlands.

[3]. Lefait, G., & Kechadi, T. (2010). Customer segmentation architecture based on clustering techniques. In Digital Society, ICDS'10. Fourth International Conference on (pp. 243-248). IEEE.

[4]. Yoseph, F., & Heikkila, M. (2018). Segmenting retail customers with an enhanced RFM and a hybrid regression/clustering method. In International Conference on Machine Learning and Data Engineering (iCMLDE) (pp. 108-116). IEEE.

[5]. Carter, N. M., Stearns, T. M., Reynolds, P. D., & Miller, B. A. (1994). New venture strategies; Theory development with an empirical base Strategic Management Journal, 15(1), 21-41.

[6]. Kashwan, K. R. & C. Velu (2013). Customer Segmentation Using Clustering and Data-mining Techniques. International Journal of Computer Theory & Engineering 5(6): 856-861.

[7]. Gunaseelan, D., & Uma, P. (2012). An improved frequent pattern algorithm for mining association rules. International Journal of Information and Communication Technology Study, 2(5).

[8]. Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics. 3 (3): 32–57.

[9]. D'Urso, P., Prayag, G., Disegna, M., & Massari, R. (2013). Market segmentation using bagged fuzzy c–means (BFCM): Destination image of Western Europe among Chinese travelers.

[10]. Miyamoto, S., Ichihashi, H., Honda, K., & Ichihashi, H. (2008). Algorithms for fuzzy clustering (pp. 1394-1399). Heidelberg: Springer.

[11]. Nayak, J., Naik, B., & Behera, H. S. (2015). Fuzzy C-means (FCM) clustering algorithm: a decade review from 2000 to 2014. In Computational intelligence in data mining -volume 2 (pp. 133-149). Springer, New Delhi.

[12]. MacQueen (1967). Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297

[13]. Wang, H., Huo, D., Huang, J., Xu, Y., Yan, L., Sun, W., & Li, X. (2010). An approach for improving the K-means algorithm on market segmentation. In System Science and Engineering (ICSSE), International Conference on (pp. 368-372). IEEE.

[14]. Dipanjan, D., Satish, G., & Goutam, C. (2011). Comparison of Probabilistic-D and k-Means Clustering in Segment Profiles for B2B Markets. SAS Global Forum.

[15]. K. Zhang, M. Hutter, and H. Jin (2009). A new local distance-based outlier detection approach for scattered real-world data. In PAKDD '09: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data-mining, pages 813–822.

[16]. E. M. Knorr and R. T. Ng (1998). Algorithms for mining distance-based outliers in large datasets. In Proc. 24th Int, Conf. Very Large Databases, VLDB, pages 392–403.

[17]. F. Angiulli, S. Basta, and C. Pizzuti (2006). Distance-based detection and prediction of outliers. IEEE Transactions on Knowledge and Data Engineering, 18:145–160.

[18]. K.M. Bataineh, M.Naji, M.Saqer. (2011). A Comparison Study between Various Fuzzy Clustering Algorithms" vol.5 no.4 Aug.

[19]. M. Ester, H.-P. Kriegel, and X. Xu (1999). A database interface for clustering in large spatial databases. In Proceedings of 1stInternational Conference on Knowledge Discovery and Data-mining (KDD-95),

[20]. S. Guha, R. Rastogi, and K. Shim. CURE, (1994). An efficient clustering algorithm for large databases. SIGMOD Rec., 27(2):73–84, Sannella, M. J. Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., The University of Washington.

[21]. T. Zhang, R. Ramakrishnan, and M. Livny. Birch (1996). an efficient data clustering method for very large databases. SIGMOD Rec., 25(2):103–114.

[22]. Provost, F., & Fawcett, T. (2013). Data Science for Business: What you need to know about data-mining and data-analytic thinking. O'Reilly Media, Inc.

[23]. Ramageri, B. M., & Desai, B. L. (2013). Role of data-mining in the retail sector. International Journal on Computer Science and Engineering, 5(1), 47.

[24]. Swati, K., & Kumar, S. (2015). A comparative study of various data transformation techniques in data-mining. International Journal of Scientific Engineering and Technology, 4, 146-148.

[25]. Bodon, F., and Rónyai, L. (2003). Trie: an alternative data structure for data-mining algorithms, Mathematical and Computer Modelling, 38(7), 739-751.

[26]. Borgelt C, (2012). Frequent item set mining (Wiley Interdisciplinary Reviews Data-mining & Knowledge Discovery (6)), pp. 437–456.

[27]. García, S., Luengo, J., & Herrera, F. (2015). Data preprocessing in data-mining (pp. 59-139). New York: Springer.

[28]. Berry, M. J. A., & Linoff, G. S. (2004). Data-mining techniques second edition - for marketing, sales, and customer relationship management. Wiley.

[29]. Yu, D., Sheikholeslami, G., and Zang, (2002). A find out: finding outliers in very large datasets, in Knowledge and Information Systems, pp. 387 - 412.

[30]. Breunig, M.M., Kriegel, H.P., and Ng, R.T., LOF: Identifying density-based local outliers., ACM Conference Proceedings, pp. 93-104

[31]. Aggarwal, C. C., Yu, S. P. (2000). "An effective and efficient algorithm for high-dimensional outlier detection, The VLDB Journal, 2005, vol. 14, pp. 211–221.

[32]. Knorr, E.M., Ng, R. T., Tucakov, V., (2000). "Distance-based outliers: algorithms and applications," The VLDB Journal, vol. 8, pp. 237–253.

[33]. Barnett and Lewis, Barnett V., Lewis T., (1994). Outliers in Statistical Data. John Wiley.

[34]. J. Han and M. Kamber, (2006). Data-mining: Concepts and Techniques, pages 440-444. Morgan Kaufmann Publishers, second edition.

[35]. Christy, A., Gandhi, G. M., & Vaithyasubramanian, S. (2015). Cluster-based outlier detection algorithm for healthcare data. Procedia Computer Science, 50, 209-215.

[36]. Bezdek, J. C., Ehrlich, R. and Full, W. (1984). FCM: The Fuzzy C-Means Clustering Algorithm, Computers & Geosciences Vol. 10, No. 2-3, pp. 191-20.

[37]. Zhang, Q. (Ed.). (2010). Visual Analytics and Interactive Technologies: Data, Text, and Web Mining Applications: Data, Text and Web Mining Applications. IGI Global.

[38]. Pachgade, M. S., & Dhande, M. S. (2012). Outlier detection over data set using a cluster-based and distance-based approach. International Journal of Advanced Research in Computer Science and Software Engineering, 2(6).

[39]. Firdaus, S., & Uddin, M. A. (2015). A survey on clustering algorithms and complexity analysis. International Journal of Computer Science Issues (IJCSI), 12(2), 62.
[40]. Dong, J., & Qi, M. (2009). K-means optimization algorithm for solving the clustering problem. In Second International Workshop on Knowledge Discovery and Data-mining (pp. 52-55). IEEE.
[41]. Maimon, O., & Rokach, L. (2005). Data-mining and knowledge discovery handbook.