big data and cognitive computing

Article

Multi-Level Clustering-Based Outlier's Detection (MCOD) Using Self-Organizing Maps

Menglu Li [1], Rasha Kashef [1,*] and Ahmed Ibrahim [2]

[1] Electrical, Computer, and Biomedical Engineering Department, Ryerson University, Toronto, ON M5B 2K3, Canada; menglu.li@ryerson.ca
[2] Department of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada; a24ibrah@uwaterloo.ca
* Correspondence: rkashef@ryerson.ca

Received: 4 September 2020; Accepted: 20 September 2020; Published: 23 September 2020

Abstract: Outlier detection is critical in many business applications, as it recognizes unusual behaviours to prevent losses and optimize revenue. For example, illegitimate online transactions can be detected based on its pattern with outlier detection. The performance of existing outlier detection methods is limited by the pattern/behaviour of the dataset; these methods may not perform well without prior knowledge of the dataset. This paper proposes a multi-level outlier detection algorithm (MCOD) that uses multi-level unsupervised learning to cluster the data and discover outliers. The proposed detection method is tested on datasets in different fields with different sizes and dimensions. Experimental analysis has shown that the proposed MCOD algorithm has the ability to improving the outlier detection rate, as compared to the traditional anomaly detection methods. Enterprises and organizations can adopt the proposed MCOD algorithm to ensure a sustainable and efficient detection of frauds/outliers to increase profitability (and/or) to enhance business outcomes.

Keywords: machine learning; clustering; outliers' detection; SOM; outlier factor; accuracy

1. Introduction

Illegal actions in business usually lead to a significant amount of financial loss, especially with those organizations that handle a large amount of data or metrics. For example, with the development of online shopping, the number of online transactional frauds is increasing, which involves scammers pretending to be legitimate online sellers, or buyers paying with an unauthorized credit card. However, it is impractical to examine each metric of the dataset over the whole timeframe manually. Therefore, discovering those anomaly behaviours in data is very critical to reduce fraud and to increase profitability. Outliers, or anomalies in general, refer to extreme objects that different from other observations of the same dataset [1]. It can cause some issues for statistical applications or training of machine learning algorithms, because an outlier may represent a variation, experimental error, or a novelty. Therefore, outlier detection is a popular topic in data mining research and is commonly used in credit card fraud detection [2], medical diagnosis [3], intrusion detection in cloud computing [4], and the pre-processing of a dataset [5]. For example, in [6], an automatic framework to estimate the production frontier is proposed. The first step in the framework is to use data points to fit a cubic function. The points that are far away from the curve are potential outliers. After eliminating all potential outlier points, the rest of the points are used to estimate the production frontier. This framework is proven to produce meaningful outcomes in simulation experiments and real-life applications. Furthermore, the centre location problem is addressed in [7], in which the location of facilities that is best suited for existing customers is decided. In the problem-solving process, the far-away customers are considered as outliers, because excluding those distant customers may lead to a reasonable and economic centre location for the decision-makers. Therefore, a k-max function is proposed to limit the influence of far-away customers for an optimal outcome. The value of k is pre-specified, the $(k-1)$ of far-away customers are detected as outliers and are not considered to the further location decision. The location decision process with outlier detection is able to provide an economic-efficient solution for the majority of customers. Outlier detection has also shown a significant impact in time-series analytics as illustrated in [8], in which they proposed a non-parametric outlier detection (FOD) for time series data. The FOD is based on the frequency-domain and Fourier transform. Firstly, the Fourier transform of time series data is calculated. Then periodic peaks and their most repetitive interval in the frequency domain are detected and transformed back to the time domain. In the time domain, the global extremes are identified as periodic outliers. In general, outlier detection methods are classified as distance-based, distribution-based, density-based, deviation-based, angle-based, deep learning-based, and clustering-based, based on the definition of an outlier. Existing outlier detection methods require prior knowledge of dataset patterns to obtain a decent detection accuracy. For example, the distance-based outlier detection method assumes inliers are closed to each other, and the density-based detection method believes inliers have more neighbour data points than outliners. However, those assumptions may not be suitable for general datasets with various types, sparsity, configurations, or prior labelling. Clustering analysis handles unlabelled data, overcomes the sparsity of data, and works efficiently on datasets with various configurations. Therefore, this paper proposes a clustering-based outlier detection technique that achieves a better detection performance for various datasets of different types and structures, as compared to traditional detection methods. In this paper, we are proposing a Multi-level Clustering-based Outlier's Detection method called MCOD. The MCOD algorithm uses two stages to discover outliers. In the first stage, a clustering process is performed on the original data to generate summarizations (i.e., cluster prototypes) of the data. In the next stage, an outlier risk factor (ORF) is assigned to each data point x. The ORF is a measure of both the size of the cluster the data point belongs to and the distance between the object and its closest cluster. In the first stage, the proposed MCOD algorithm uses self-organizing maps (SOM) [9] as the base level of clustering, due to its efficiency in handling several types of classification problems while providing a useful, interactive, and intelligible summary of the data. The major disadvantage of the SOM is that it requires necessary and sufficient data to develop meaningful clusters. Furthermore, the clustering performance strongly depends on the initial weight vectors. Initializing the weight vector of SOM with the prior knowledge of datasets significantly helps to group the input data correctly. The proposed MCOD provides a multi-level clustering process that enhances the quality of the SOM and provides a significant outlier detection capability using a cascaded-level clustering and detection process. In this paper, the MCOD is applied to datasets in different fields, such as biomedical datasets and credit card fraud transactions. Experimental results show that the MCOD demonstrates its capability of improving the outlier detection rate in comparison with state-of-art methods. Utilizing the MCOD, business organizations can significantly detect instant frauds in data and subsequently increase profit or users' outcomes. The rest of this paper is organized as follows: Section 2 introduces the background of current outlier detection methods; Section 3 presents the proposed model of multi-level clustering-based outlier detection; Section 4 describes the experimental analysis; Section 5 outlines the conclusions and the future works.

2. Related Work and Background

This section provides the commonly used techniques to detect outliers, which are based on distance, distribution, density, deviation, angle, network connections, and clusters. For each technique, the methodology to detect outliers and its execution complexity are discussed.

2.1. Distance-Based Outlier Detection

This approach identifies an outlier based on the distance to its neighbours. If the locality of a data point is sparsely populated, then this point is an outlier [10]. A reasonable distance value and a reasonable number of neighbours are set to be the thresholds. If the distance between two points is within the distance threshold, these two points are considered as a neighbour. The number of neighbours is the criterion for defining an outlier. This detection scheme was formalized by [11]. Given a dataset X, an object $x \in X$ is an outlier if it meets the following condition:

$$\left|\left\{x' \in X \,\middle|\, dist(x,x') > \delta\right\}\right| \geq \alpha n. \tag{1}$$

where n presents the number of objects in the dataset, and $\alpha, \delta \in \mathbb{R}$ $(1 \geq \alpha \geq 0)$ are thresholds. To improve the drawback of this distance-based method, which includes the lack of a ranking of outliers, the method of k-NN distance-based outlier detection is proposed, which gives each object a score by measuring the distance of its kth-nearest neighbour (kth-NN) [12]. Outliers can be ranked and identified by its score. The work [13] proved that the distance-based outlier detection method is capable of providing a comparable accuracy with a low computation cost.

2.2. Distribution-Based Outlier Detection

The distribution-based method is known as statistical-based outlier detection, which assumes, that in a normal dataset without outliners, all data follow a stochastic model. This approach requires prior knowledge of the datasets, such as distribution, mean, and variance. A data point is classified as an outliner if it deviates from the target distribution. For a dataset X, the target distribution usually is a normal distribution $N\left(\mu, \sigma^2\right)$ and a standard deviation α are chosen as a threshold. Let $L(X,\alpha)$ be the lower bound of the standard deviation and $R(X,\alpha)$ be the upper bound. Areas lower than $L(X,\alpha)$ or higher than $R(X,\alpha)$ are considered as outlier region, which is expressed as

$$out\left(\alpha, \mu, \sigma^2\right) = \{-\infty, L(X,\alpha)] \cup [R(X,\alpha), \infty\} \tag{2}$$

An object is detected as an outlier as lying in the outlier region $out\left(\alpha, \mu, \sigma^2\right)$ [14]. For the multivariate case, the *Mahalanobis* distance is a popular-used criterion to detect outliers. Let $\overline{x}$ denote the mean vector of dataset X, and V be the covariance matrix. Then, the *Mahalanobis* distance M_i for each object i is calculated, which is given by

$$M_i = \left(\sum_i (x_i - \overline{x})^T V^{-1} (x_i - \overline{x})\right)^{\frac{1}{2}} \tag{3}$$

The object with a larger *Mahalanobis* distance M_i, x_i is classified as the outlier [15]. However, there are some drawbacks to this approach. For example, the data distribution is not pre-known in practice. Furthermore, it is difficult to estimate the actual distribution of the dataset for high dimensional data points.

2.3. Density-Based Outlier Detection

The density-based outlier detection method analysis the difference between the density of an object and the density of its neighbours. This method assumes the density of a normal object is similar to the density of its neighbours; therefore, if an object has a density that significantly different from its neighbours, this object will be considered as an outlier. The local outlier factor (LOF) is one of the most well-known unsupervised outlier detection methods, which functions similarly to the k-NN detection method [16]. Let $dist_k(x)$ be the distance between object x and its k-nearest neighbours x', and $N^k(x)$ be the set of k nearest neighbours $(kNNs)$ of the object x, the reachability distance and the local reachability density are defined in Equations (4) and (5), respectively. The LOF of object x is defined as in Equation (6).

$$reachdist_k(x,x') = max[dist_k(x),\ dist(x,x')] \tag{4}$$

$$lrd_k(x) = \frac{\left\|N^k(x)\right\|}{\sum_{x' \in N^k(x)} reachdist_k(x, x')} \tag{5}$$

$$LOF_k(x) = \frac{\sum_{x' \in N^k(x)} \frac{lrd_k(x')}{lrd_k(x)}}{\left\|N^k(x)\right\|} \tag{6}$$

The LOF indicates the average of the ratio of *local reachability density* of an object x and its k-nearest neighbours x'. An object with a high LOF value is identified as a local outlier. The *SimplifiedLOF* [17] differs from the standard LOF where the reachability distance (Equation (6)) is replaced by the k-NN distance, resulting in a simpler density estimate defined by the following:

$$dens(x) = \frac{1}{MinPts{\cdot}distance(x)} \tag{7}$$

The *SimplifiedLOF* has often been used implicitly and often unintentionally where the reachability had not been explicitly defined. The density estimate stems predominately from LOF when in the $reach\_dist_{MinPts}(x, y)$ equation, $MinPts - distance(x)$ is substituted for $MinPts - distance(y)$: since $y \in N_{MinPts}(x)$, $distance(x, y) \leq MinPts - distance(x)$.

2.4. Deviation-Based Outlier Detection

Based on the deviation-based outlier detection method, an object is classified as an outlier if it cannot fit into the main characteristics of the dataset. This approach simulates a mechanism in which human beings capture discordant objects from a series of similar objects. The sequential exception technique is one of the most popular [18]. For a dataset X, define a $Smoothing\ factor$ as the threshold. The threshold, $SF(I)$, indicates how much the deviation can be reduced by removing the object I from dataset X. An object x is identified as an outlier if it satisfies the following condition in Equation (8), The sequential exception technique involves a high computational cost of $O(2^n)$ for n objects.

$$SF(x) \geq SF(I) \tag{8}$$

2.5. Angle-Based Outlier Detection

In the angle-based outlier detection [19] approach, the variance in the angles between the outlier candidate and all other pairs of points are assessed as the angle-based outlier factor (ABOF) value of it. The Standard Angle-based Outlier Detection (ABOD) approach requires to calculate ABOF of each data points. The ABOF is defined as:

Definition 1. *Given a database* $\mathcal{D} \subseteq \mathbb{R}^d$, *a point* $\vec{A} \in \mathcal{D}$, *and a norm* $\|\cdot\| : \mathbb{R}^d \rightarrow R_0^+$, *where* $R_0^+ = \{x | x \in \mathbb{R} \wedge x \geq 0\}$. *The scalar product is denoted by* $\langle ., . \rangle : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. *For two points* $\vec{B}, \vec{C} \in \mathcal{D}$, $\overline{BC}$ *denotes the difference vector* $\vec{C} - \vec{B}$. *The angle-based outlier factor, ABOF* ($\vec{A}$), *is the variance over the angles between the difference vectors of* $\vec{A}$ *to all pairs of points in* $\mathcal{D}$ *weighted by the distance of the points:*

$$ABOF\left(\vec{A}\right) = VAR_{\vec{B}, \vec{C} \in \mathcal{D}}\left(\frac{\left\langle \overline{AB}, \overline{AC} \right\rangle}{\left\|\overline{AB}\right\|^2 \cdot \left\|\overline{AC}\right\|^2}\right) \tag{9}$$

$$ABOF\left(\vec{A}\right) = \frac{\sum_{\vec{B}\in\mathcal{D}}\sum_{\vec{C}\in\mathcal{D}}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}\cdot\left(\frac{\left\langle\overline{AB},\overline{AC}\right\rangle}{\left\|\overline{AB}\right\|^2\cdot\left\|\overline{AC}\right\|^2}\right)^2}{\sum_{\vec{B}\in\mathcal{D}}\sum_{\vec{C}\in\mathcal{D}}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}} - \left(\frac{\sum_{\vec{B}\in\mathcal{D}}\sum_{\vec{C}\in\mathcal{D}}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}\cdot\frac{\left\langle\overline{AB},\overline{AC}\right\rangle}{\left\|\overline{AB}\right\|^2\cdot\left\|\overline{AC}\right\|^2}}{\sum_{\vec{B}\in\mathcal{D}}\sum_{\vec{C}\in\mathcal{D}}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}}\right)^2 \tag{10}$$

It should be noted that $\vec{A}, \vec{B}, \vec{C}$ are all mutually different. After calculating the ABOF score, the standard ABOD ranks data points according to the outcome of the ABOF values. Given the inherent complexity of the ABOD, the primary issue with the aforementioned approach for outlier detection is the efficiency of the model. Using n to denote the number of points in the database, the time complexity of the ABOD is $O\left(n^3\right)$, which is taxing on the system performing the analysis. As weight is used in the calculation of ABOF, distant data points from a particular $\vec{A}$ is supposed to be of less importance than points neighbouring $\vec{A}$. Therefore, the formula for the FastABOD's corresponding ABOF was proposed to use pairs of points only in the set of N_{MinPts}. The following is the definition of the FastABOD [20].

Definition 2. *Given a database $\mathcal{D} \subseteq \mathbb{R}^d$, a point $\vec{A} \in \mathcal{D}$, and a norm $\|\cdot\| : \mathbb{R}^d \to R_0^+$, where $R_0^+ = \{x | x \in \mathbb{R} \wedge x \geq 0\}$. The scalar product is denoted by $\langle .,. \rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. For two points $\vec{B}, \vec{C} \in \mathcal{D}$, $\overline{BC}$ denotes the difference vector $\vec{C} - \vec{B}$. $N_{MinPts}\left(\vec{A}\right) \subseteq \mathcal{D}$ denotes the set of the N_{MinPts} nearest points of $\vec{A}$. The approximate angle-based outlier factor, $approxABOF_{N_{MinPts}}\left(\vec{A}\right)$, is the variance over the angles between the difference vectors of $\vec{A}$ to all pairs of points in $\mathcal{D}$ weighted by the distance of the points:*

$$ABOF\left(\vec{A}\right) = VAR_{\vec{B},\vec{C}\in N_{MinPts}\left(\vec{A}\right)}\left(\frac{\left\langle\overline{AB},\overline{AC}\right\rangle}{\left\|\overline{AB}\right\|^2\cdot\left\|\overline{AC}\right\|^2}\right) \tag{11}$$

$$ABOF\left(\vec{A}\right) = \frac{\sum_{\vec{B}\in N_{MinPts}(\vec{A})}\sum_{\vec{C}\in N_{MinPts}(\vec{A})}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}\cdot\left(\frac{\left\langle\overline{AB},\overline{AC}\right\rangle}{\left\|\overline{AB}\right\|^2\cdot\left\|\overline{AC}\right\|^2}\right)^2}{\sum_{\vec{B}\in N_{MinPts}(\vec{A})}\sum_{\vec{C}\in N_{MinPts}(\vec{A})}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}} - \left(\frac{\sum_{\vec{B}\in N_{MinPts}(\vec{A})}\sum_{\vec{C}\in N_{MinPts}(\vec{A})}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}\cdot\frac{\left\langle\overline{AB},\overline{AC}\right\rangle}{\left\|\overline{AB}\right\|^2\cdot\left\|\overline{AC}\right\|^2}}{\sum_{\vec{B}\in N_{MinPts}(\vec{A})}\sum_{\vec{C}\in N_{MinPts}(\vec{A})}\frac{1}{\left\|\overline{AB}\right\|\cdot\left\|\overline{AC}\right\|}}\right)^2 \tag{12}$$

The resulting time complexity is $O\left(n^2 + n{\cdot}N_{MinPts}^2\right)$. Furthermore, it is noted that as long as the number N_{MinPts} is selected small enough with respect to n, the FastABOD algorithm provides a marked acceleration.

2.6. Deep Learning-Based Outlier Detection

Deep learning methods, such as artificial neural networks (ANN), are recently used for fraud detection. The research of [21] on multiple structures of ANN models to detect fraud in the credit card transactions contained three types of layers: the input layer, hidden layers, and the output layer. They test the detection accuracy performance by one, two, and three hidden layers with one, ten, one hundred, and one thousand nodes using various activation functions, like the Relu, sigmoid, tanh, and identity functions. As a result, the highest precision rate of 96% is found when the model consists of two hidden layers with 1000 nodes and the Relu activation function. Furthermore, the sigmoid function gives the best sensitivity value than other activation functions. The work [22] presented four different deep learning detection methods and compares their performance on an identical dataset. The four methods were artificial neural networks (ANN), recurrent neural networks (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU). RNN, which is a variant of the ANN model, can model large sequential data because the RNN has links not only between layers but also between neurons in the same layer. The LSTM model involves a memory cell to store the state of the neuron. The function of the GRU model is to make each recurrent unit in RNN capture dependencies of different time scales. Their research indicates that the fraud detection accuracy of the LSTM and GRU is higher than the ANN model, which is a performance baseline. The research also shows that the larger network performs better than small networks. Instead of considering deep learning methods only, [23] compared several machine learning methods and deep learning methods for credit card fraud detection. The machine learning methods they used were k-nearest neighbour (KNN), random forest, and support vector machines (SVM), while the deep learning methods were convolutional neural networks (CNN), restricted Boltzmann machine (RBM), and deep belief networks (DBN). They apply all these methods on three different size datasets and use the area under the Receiver Operating Characteristic (ROC) curve, denoted by the AUC to evaluate the detection performance. The work [23] concluded that the best method for detecting larger datasets is by using SVM, potentially combined with CNN to attain more reliable performance. Comparing the deep learning methods only, CNN always provides higher accuracy and fewer false alarms than other methods, such as RBM and DBN.

2.7. Clustering-Based Outlier Detection

Cluster analysis is a grouping process that divides data objects into multiple collections in an unsupervised process. There are no predefined classes; instead, data objects are grouped based on their characteristics. As the clustering result, data objects that are in the same cluster are similar to each other, and they are dissimilar to the objects in other clusters. A way to define the clustering result whether it is high quality in clusters should have high intra-class similarity and low inter-class similarity. Clustering has a quite broad field of applications, such as medicine, social science, and market research. Therefore, several clustering approaches are studied to produce high-quality clusters. In clustering-based outliers detection methods, such as FindCBLOF [24], outliers form small clusters and are of a far distance of objects in large clusters [24,25]. The work [26] proposed a clustering method that assigns and updates a weight of relevance to each attribute of objects to decrease the impact of noise on the clustering result. For a dataset X, each object x which contains a set of m attributes are divided into multiple data chunk. Given a chunk size n, the number of clusters k and the weight vector W can be calculated as:

$$w_j = \begin{cases} 0, if\ D_j = 0 \\ \dfrac{1}{\sum_{j=1}^{h} \left[\frac{D_j}{D_t}\right]^{\frac{1}{\beta-1}}}, if\ D_j \neq 0 \end{cases}, \ where\ D_j = \sum_{l=1}^{k} \sum_{i=1}^{n} u_{i,l} d\left(x_{i,j}, z_{l,j}\right) \tag{13}$$

$$\sum_{j=1}^{m} w_j = 1\ and\ 0 \leq w_j \leq 1 \tag{14}$$

where w_j indicates the weight of j^{th} attribute of object x. $u_{i,l}$ equals a value of 1 if the i^{th} object is in the l^{th} cluster; otherwise, it equals 0. β is a user-defined parameter, $d()$ is the distance measure and z_l is the center of the l^{th} cluster. After updating the weight vector using Equations (13) and (14) to form the k clusters, outliers are detected if they are far away from its cluster centre than other objects. This method performs a high outlier detection rate and a low false alarm rate with a low time consumption [26]. Clustering-based outliers' detection has shown a great impact in the area of outlier's detection, as (1) it performs two dual tasks at the same time including clustering the data and detecting outliers, and (2) it does not need ground truth to train the model (i.e., it does not need prior knowledge about the data to build a detection model) [27].

3. Multi-Level Clustering-Based Outlier Detection

In this paper, we propose a multi-level clustering-based outliers detection method called MCOD. The MCOD algorithm uses two stages to discover outliers. In the first stage, a clustering process is performed on the original data to generate summarizations (i.e., cluster prototypes) of the data. Two sets of clusters are generated, a large population set (LPS), and a small population set (SPS), such that a cluster $S_i \in$ the LPS if its sizes | S_i | exceeds α *|X|, where X is the entire dataset, and α is a predetermined threshold; otherwise, S_i is a member of the SPS. In the next stage, an outlier risk factor (ORF) is assigned to each data point x; the ORF (x) is a measure of both the size of the cluster the data point x belongs to and the distance between the object and its closest cluster (if the object lies in a small cluster). For two data points, x and y in the d-dimensional space, the measure of closeness the two data points by:

$$Distance\ (x, y) = 1 - Sim(x, y) \tag{15}$$

For datasets with spherical shapes, we used the Euclidian distance as a measure of dissimilarity (i.e., large distance), such that *Distance(x,y) = Euclidian (x,y)*, and for datasets with high sparsity and dimensionality, we used cosine similarity as a measure of homogeneity (i.e., high similarity and *Sim(x,y) = Cosine(x,y)*). Given a numeric parameter α, a cluster S_i, and a prototype z_i, the ORF of an object $x \in S_i$ represented by z_i is defined as in Equation (16). The final outliers are returned as objects with high ORF values.

$$ORF(x) = \begin{cases} |S_i| * \min\left(Distance\left(x, z_j\right)\right), x \in S_i, S_i \in SPS\ and\ S_j \in LPS \\ |S_i| * (Distance(x, , z_i)), x \in S_i,\ S_i \in LPS \end{cases} \tag{16}$$

It can be shown that the efficiency of detecting outliers is constrained to the quality of the adopted clustering technique. In [25] and [28], it has been experimentally proven that better clustering solutions reveal better detection of outliers. Self-organizing maps (SOM) cluster the datasets into groups with high homogeneity and better overall clustering quality, as compared to traditional unsupervised clustering algorithms [9]; thus, in this paper, we focus on using the SOM as the base level of clustering in our proposed algorithm. More information on SOM is provided next.

3.1. SOM Clustering

SOM clustering is used to produce a two-dimensional map to represent a high dimensional input training dataset. In this case, the number of neurons on the 2-D map represents the number of clusters. Each N-dimensional input data point x in the dataset is connected to the map with the neurons of the size of M through weights w_{ij}, where $i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, M$. For M clusters, there are a M number of w_i vectors. In the beginning, those weight vectors are initialized as random values. During the training process, each w_i vector gets updated to describe the input patterns associated with those clusters. The square of the Euclidean distance is used to measure the relationship between data point x and each weight vector w_i. The neuron j with a weight vector w_{ij}, where $i = 1, 2, \ldots, N$ that closely matches the data point x is chosen as the cluster that point x belongs to, which also means $\left\|x - w_{ij}\right\|^2$ is the smallest [9]. The major disadvantages of SOM are that it requires sufficient data to perform a good clustering, and the order of presentation of the training data impacts the final map solution [29]. The clustering performance strongly depends on the initial weight vectors. Initializing the weight vector of SOM with the prior knowledge of datasets significantly helps to group the input data correctly [30].

3.2. The MCOD (Ai-SOM) Outlier's Detection Algorithm

Since the SOM highly depends on the initialization, in the first stage of the MCOD algorithm, we propose a multi-level clustering algorithm approach that uses two levels of clustering in a cascade-level approach. In the first level, an $\mathrm{A_i}$ clustering method is applied to the dataset to divide the data points into *k* clusters. The A_i algorithm can be of any type of the commonly used clustering techniques, such as K-Means (KM) [31], Bisecting K-Means (BKM) [32], Partitioning Around Medoids (PAM) [33], and Fuzzy C-means (FCM) [34]. The appropriate A_i clustering method is selected based on the characteristics of the dataset. In the second level, the clustering result from the A_i method is used as the initial seeds to the SOM clustering method. The A_i algorithm is able to provide knowledge about the dataset and give a reasonable initial weight vector to the SOM process. Adding one prior clustering level to SOM can improve the cluster performance to the dataset so that the multi-level-based clustering outlier detection method is capable of classifying outliers for datasets with general configurations and properties. In the MCOD, the number of clusters, *k*, needs to be specified. The A_i algorithm groups the dataset into *k* clusters. The average of all data points in each cluster from the A_i algorithm is computed. Then, the clustering result from the A_i algorithm is passed to SOM. The number of neurons in the SOM is the same as the number of clusters in the A_i result, and the initial weight vector for these neurons is now initialized non-randomly. Instead, the average values of each clustering from the A_i algorithm are assigned to be the initial weight of each neuron in SOM. The multi-level clustering is shown in Figure 1. Using this multi-level cascaded strategy, the MCOD overcomes the initialization problem that the SOM suffers from, thus providing significant fraud detection capabilities for organizations and enterprises. The ORF for all data points is computed based on the clustering result after stage 1, using the provided initial weights and specified learning rate. A data point is classified as an outlier if it has a high ORF value. The MCOD algorithm is illustrated in Algorithm 1.

Algorithm 1 Multi-Level Clustering-Based Outlier's Detection (MCOD) (Ai-SOM)

```
    Input: Dataset X of $n$ records and $d$ dimension, Algorithm $A_i$, Number of clusters k, Learning rate η,
Alpha α
    Output: Top % outliers
    Begin
    Step1://Apply $A_i$ on X to obtain $k$ cluster
    Clusterj ← $A_i$ algorithm (X, k) where $j$ = 1, 2, 3, . . . , $k$
    Step2://Initialize a vector W with the size $k$
    for j ← 1, 2, 3, . . . , k do
        $\vec{w}$ j = mean (Clusterj)
    end for
    Step 3://Reshape W to a 2D matrix that matches the shape of SOM
    for x ← 1, 2, 3, . . . , sqrt(k)
        for y ← 1, 2, 3, . . . , sqrt(k)
            z = (x-1)·sqrt(k)+y
    Wx,y = $\vec{w}$ z
    end for
    end for
    Step 4: //Apply SOM and update W by using η to obtain updated k cluster
    UpdatedClusterj ← SOM (Clusterj, W, k, η) where j = 1, 2, 3, . . . , k
    Step 5: Find the ORF factor for each object x in the updated k cluster given α
    Step 6: Select top % data points with high value of ORF as outliers
End
```

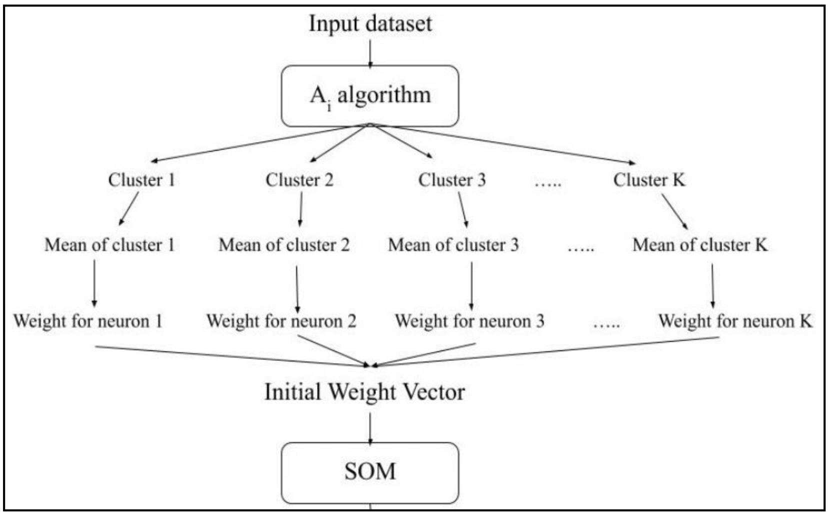

Figure 1. The first stage in the Multi-level Clustering Algorithm (MCOD).

4. Experiment Analysis

4.1. Datasets

Experiments were performed on artificial, biomedical, documents datasets, and credit card datasets with various characteristics and degree of outliers. A summary of the experimental datasets is shown in Table 1.

Table 1. Experimental Datasets.

Dataset	*n*	*K*	*d*
HBK	75	4	4
Wood	20	4	6
Cardio	2126	100	25
BC	699	100	9
RBC	13,731	100	15
European Credit	284,807	100	29

4.1.1. Artificial Datasets

The Hawkins, Bradu, and Kass (HBK) dataset [35] and the Wood dataset [36] are two artificial datasets with known true outliers used for our experimental analysis. The HBK is an artificially generated random dataset with 75 observations in four dimensions. The dataset contains 14 outliers. The Wood dataset consists of 20 observations, with data points 4, 6, 8, and 19 being outliers.

4.1.2. Biomedical Datasets

Two biomedical datasets were used: the Cardiotocography dataset [37] and the Breast Cancer dataset [38]. The cardiotocography dataset contains 2126 cases with 25 variables. Both pathologic and suspect cases are classified as outliers, while the normal cases formed the inliers. The percentage of outliers in this dataset is 22.5%. The total number of instances in the Breast Cancer dataset is 699, with 34.5% outliers. Each instance of this dataset has nine attributes.

4.1.3. Credit Card Datasets

The Royal Bank of Canada (RBC) dataset is a real dataset provided by the RBC bank [39]. The dataset contains 13,731 credit card transactions. Each transaction includes 15 variables, which are the result of a Principal Component Analysis (PCA) transformation. All transactions are labelled as fraud or non-fraud. There are 415 fraud transactions, which account for 3.02% of all datasets. The secondary dataset, European Credits, contains credit card transactions in September 2013 by the European credit cardholders, which has 492 instances of fraud out of 284,807 transactions [40]. The percentage of fraud transactions is 0.172% in this dataset. This dataset has 29 numerical variables transformed by PCA.

4.2. Adopted Outliers Detection Algorithms

The detection accuracy of the MCOD(A_i-SOM) is compared with that of the traditional density-based Local Outlier Factor (LOF) detection method [16] as well as the clustering-based detection approach FindCBLOF(A_i) [24] approach, where $A_i \in$ {KM, BKM, PAM, FCM, SOM}, using the *k*-means (KM) [31], Bisecting k-means (BKM) [32] or Partitioning Around Medoids (PAM) [33], Fuzzy k-means (FCM) [34], and Self-Organizing Map (SOM) [9]. The FindCBLOF(KM) uses the K-means clustering method to divide the dataset D into *k* clusters, which can be represented as $C = \{C_1, C_2, C_3, C_4, \ldots, C_k\}$ such that $C_i \cap C_j = \emptyset$ and $C_i \cup C_j = D$. The FindCBLOF(BKM) uses bisecting K-means (BKM) to split the dataset into *k* clusters. BKM firstly considers the dataset as one whole cluster, then divides one cluster into two sub-clusters at each bisecting step using K-means. The criterion of choosing cluster to bisect is based on the number of data points in the clusters. The cluster with the greatest number of data points is selected to be partitioned using K-means into two clusters. The bisecting process stops until the number of clusters is reached. The FindCBLOF(PAM) uses the PAM technique to divide the dataset into *k* number of clusters. The difference between PAM and K-means is that PAM uses medoids of a cluster instead of the mean, and the cluster centre is a data point inside the cluster. The FindCBLOF(FCM) uses Fuzzy *c*-means (FCM), in which, each data point has a degree of belonging to each cluster, and the cluster centre is the mean of all data points in this cluster weighted by their degrees of membership. Finally, the FindCBLOF(SOM) uses the self-organizing map (SOM) to cluster the dataset. For each FindCBLOF(A_i) approach, where $A_i \in$ {KM, BKM, PAM, FCM, SOM}, every data point is assigned with a clustering-based local outlier factor (CBLOF), and those with the largest value of CBLOF are considered as outliers depending on the choice of contamination. Table 2 shows the parameter settings for each of the above algorithms where *k* is the number of clusters and MaxITER is the maximum number of iterations.

Table 2. Parameter Settings for the Adopted Algorithms.

Algorithm	Parameters
KM	k = 4 or 100, MaxITER = 300
BKM	k = 4 or 100, Maximum number of trials to select cluster to bisect = 20
PAM	k = 4 or 100, MaxITER = 20
FCM	k = 4 or 100, MaxITER = 100
SOM	Map size = 2 × 2 or 10 × 10, Sigma = 0.5, Learning rate = 0.5 Neighbourhood function = Gaussian
LOF	Number of neighbours = 10
CBLOF	Alpha = 0.9, Beta = 5 Add weight to outlier score calculation = True

4.3. Evaluation Criteria

The performance of the detection algorithms performance is measured by the number of detected outliers compared to true outliers, area under the precision-recall curve (AUPRC), and the accuracy of label prediction. The AUPRC indicates the model's capability of distinguishing between fraud and non-fraud. The execution time of each detection method is also measured and compared.

4.4. Individual Clustering Results

As stated in [25] and [41], better clustering solutions reveal better detection of outliers; thus, to test the performance of the proposed MCOD algorithm, using a clustering method A_i, we tested the performance of each of the clustering methods separately, using silhouette score [42], as shown in Table 3. The silhouette score of each instance measures how similar an object is to its own cluster compared to other clusters by calculating the intra-cluster distance and the distance to its nearest cluster. The silhouette score of a dataset is the mean of each instance's silhouette score. The score range is between −1 to 1. A score that is closed to 1 indicates a good clustering performance. Scores near zero means overlapping clusters. If the silhouette score is a negative value, it generally indicates that an instance has been assigned to the wrong cluster, or the clusters are too similar. It can be shown in Table 3 that SOM and KM has the best performance measured by high silhouette Score for the HBK, Wood, Cardio, BC, and RBC datasets.

4.5. Experiment 1: Artificial Datasets

Tables 4 and 5 show the number of detected true outliers for the FindCBLOF, LOF, and the proposed MCOD clustering using KM, BKM, PAM, FCM, SOM algorithms on the HBK and Wood datasets. For these two artificially constructed datasets, the true labelled outliners are provided [43]; thus, all outliners detected by the LOF, FindCBLOF and MCOD algorithms are tested against the true outliers. The detection models generated by the MCOD clearly identify the true outlier records compared to the LOF and the FindCBLOF methods. We can also observe that MCOD(KM-SOM) outperforms the MCOD(FCM-SOM), MCOD(BKM-SOM), and MCOD(PAM-SOM) as the KM itself has better clustering quality (measured by high values of the Silhouette score as shown in Table 3).

Table 3. Clustering quality (silhouette score).

Dataset	KM	BKM	FCM	PAM	SOM
HBK	0.8757	−0.2003	0.3973	0.3251	0.9210
Wood	0.4364	−0.0836	0.4360	0.2097	0.4364
Cardio	0.2498	−0.3784	0.1544	0.1347	0.2344
BC	0.2227	−0.6473	0.0297	0.1953	0.3430
RBC	0.8928	0.3414	0.6982	0.7399	0.8719
European Credit	0.1841	−0.1803	0.0796	0.1425	0.1344

Table 4. The number of true detected outliers (LOF and FindCBLOF).

Dataset	LOF	FindCBLOF (KM)	FindCBLOF (BKM)	FindCBLOF (PAM)	FindCBLOF (FCM)	FindCBLOF (SOM)
HBK	3	5	7	9	5	11
Wood	4	1	1	2	1	3

Table 5. The number of true detected outliers (MCOD).

Dataset	MCOD (KM-SOM)	MCOD (BKM-SOM)	MCOD (PAM-SOM)	MCOD (FCM-SOM)
HBK	14	12	13	13
Wood	4	3	3	4

4.6. Experiment 2: Medical Datasets with True Outliers

The FindCBLOF, LOF, and MCOD were applied to the Cardiotocography and Breast Cancer medical datasets. In this experiment the TopRatio ranges from 10% to 30%. From each detection method, data points with a high ORF value within the percentage of the TopRatio were considered outliers, and they were compared with the true outliers. Tables 6 and 7 show the number of true outliers detected by the LOF and the FindCBLOF for the Cardiotocography and Breast Cancer medical datasets, respectively. Tables 8 and 9 present the detection quality of the MCOD algorithm for the Cardiotocography and Breast Cancer medical datasets, respectively.

Table 6. Number of Detected Outliers for the Cardio Dataset (FindCBLOF).

TopRatio	LOF	FindCBLOF (KM)	FindCBLOF (BKM)	FindCBLOF (FCM)	FindCBLOF (PAM)	FindCBLOF (SOM)
10%	31	71	23	68	72	44
15%	50	98	35	87	90	60
20%	75	124	53	114	112	83
25%	85	158	76	132	123	108
30%	92	171	99	162	155	134

It can be shown that, for both the Cardiotocography and the Breast Cancer dataset, the number of detected true outliers using the MCOD method was more than those detected by the LOF and FindCBLOF. For the Cardiotocography dataset, the FindCBLOF(A_i), where $A_i \in$ {KM, BKM, PAM, FCM, SOM}, the technique was able to detect 22% of true outliers on average, while the MCOD(A_i-SOM) techniques achieved 42%. The result from the Breast Cancer dataset shows the same performance. The MCOD increases the average detection rate of true outliers from 48% by using FindCBLOF to up to 81%.

Table 7. Number of Detected Outliers for the Breast Cancer Dataset (FindCBLOF).

TopRatio	LOF	FindCBLOF (KM)	FindCBLOF (BKM)	FindCBLOF (FCM)	FindCBLOF (PAM)	FindCBLOF (SOM)
10%	11	45	19	28	32	56
15%	24	52	23	39	63	74
20%	45	69	36	50	89	99
25%	59	81	44	68	97	129
30%	68	93	50	76	131	159

Table 8. Number of Detected Outliers for the Cardio Dataset (MCOD).

TopRatio	MCOD (KM-SOM)	MCOD (BKM-SOM)	MCOD (PAM-SOM)	MCOD (FM-SOM)
10%	82	85	82	79
15%	123	123	108	110
20%	155	144	137	136
25%	185	169	162	164
30%	214	197	182	184

Table 9. Number of Detected Outliers for the Breast Cancer Dataset (MCOD).

TopRatio	CBLOF (KM-SOM)	MCOD (BKM-SOM)	MCOD (PAM-SOM)	MCOD (FCM-SOM)
10%	63	33	47	42
15%	85	55	74	68
20%	116	83	97	97
25%	144	105	126	118
30%	173	128	154	145

4.7. Experiment 3: Real Credit Card Datasets

Both the LOF, FindCBLOF, and MCOD detection methods were applied on two real credit card datasets. Both datasets are labelled with true and false outliers. Multiple contaminations of outliers were selected, and the detection labels from each method and contamination level were compared with the true labels. The number of correctly and incorrectly detected labels was counted and recorded. For measuring detection performance, the value of area under precision and recall curve (AUPRC) and detection accuracy were calculated. The AUPRC provides a trade-off between precision and recall at various thresholds, i.e., contamination level, in our case. The AUPRC represents high recall and precision, where high precision shows a low false-positive rate, and high recall shows a low false-negative rate. Figures 2 and 3 show the detection performance for the RBC dataset. It can be shown that for the RBC dataset, the MCOD algorithm improved the AUPRC value by 10% to 15% and increased the detection accuracy by up to 5.6%. Figures 4 and 5 show the detection performance for the European Credit dataset. It can be shown that the MCOD algorithm improved the AUPRC value by 22%, as compared to the LOF and the FindCBLOF methods. Furthermore, the detection accuracy of the fraud transactions in data increased by up to 10%.

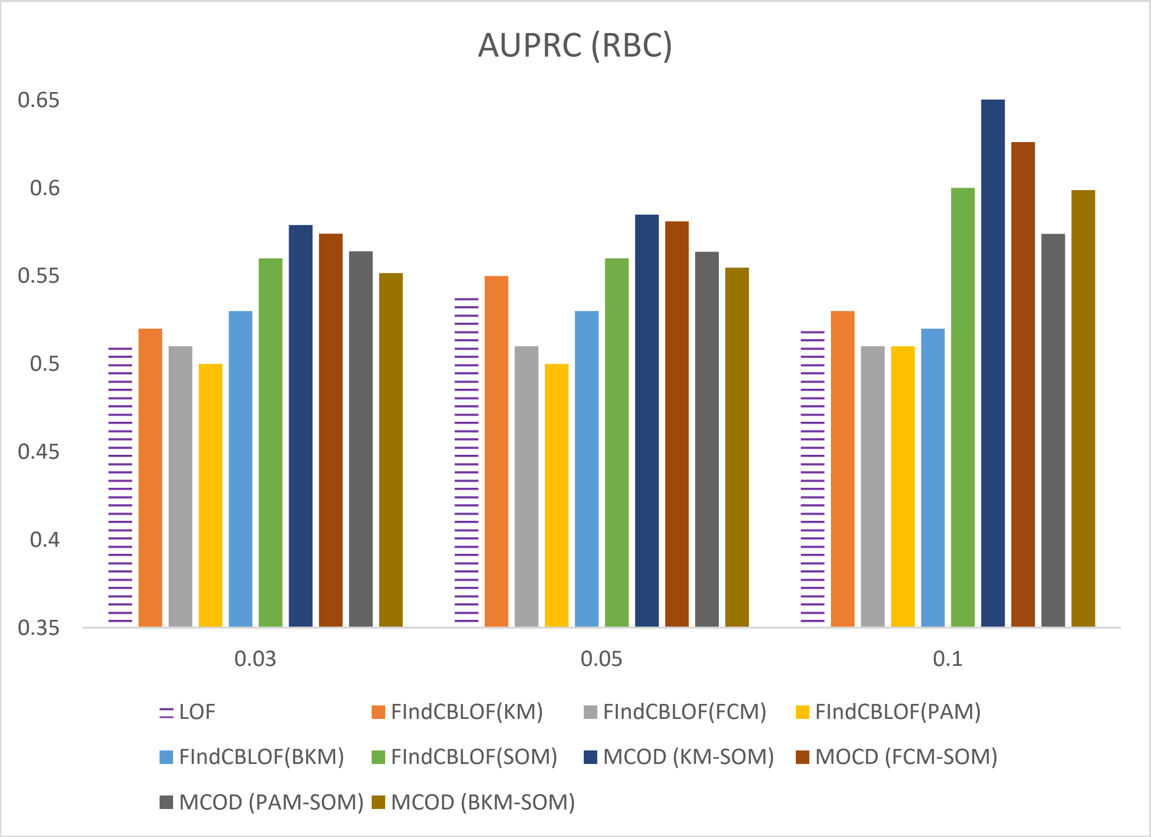

Figure 2. AUPRC (RBC Dataset).

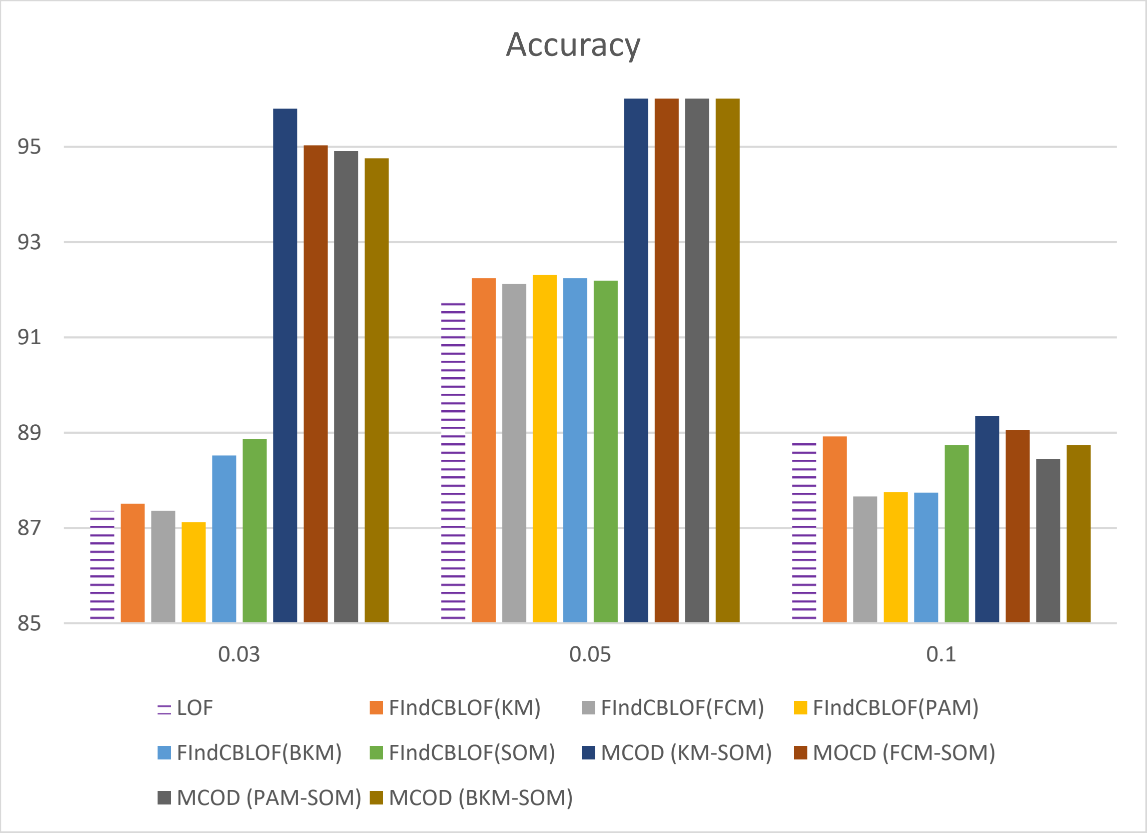

Figure 3. Detection Accuracy (RBC Dataset).

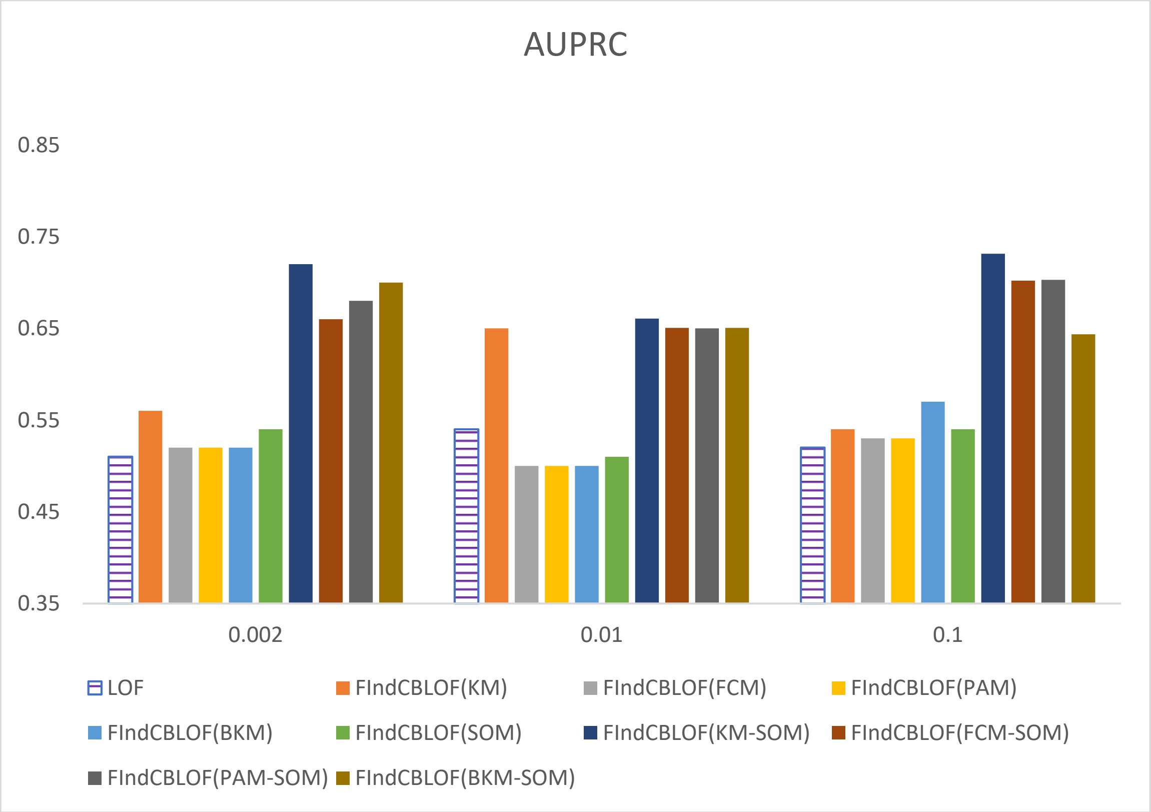

Figure 4. AUPRC (European Credit Dataset).

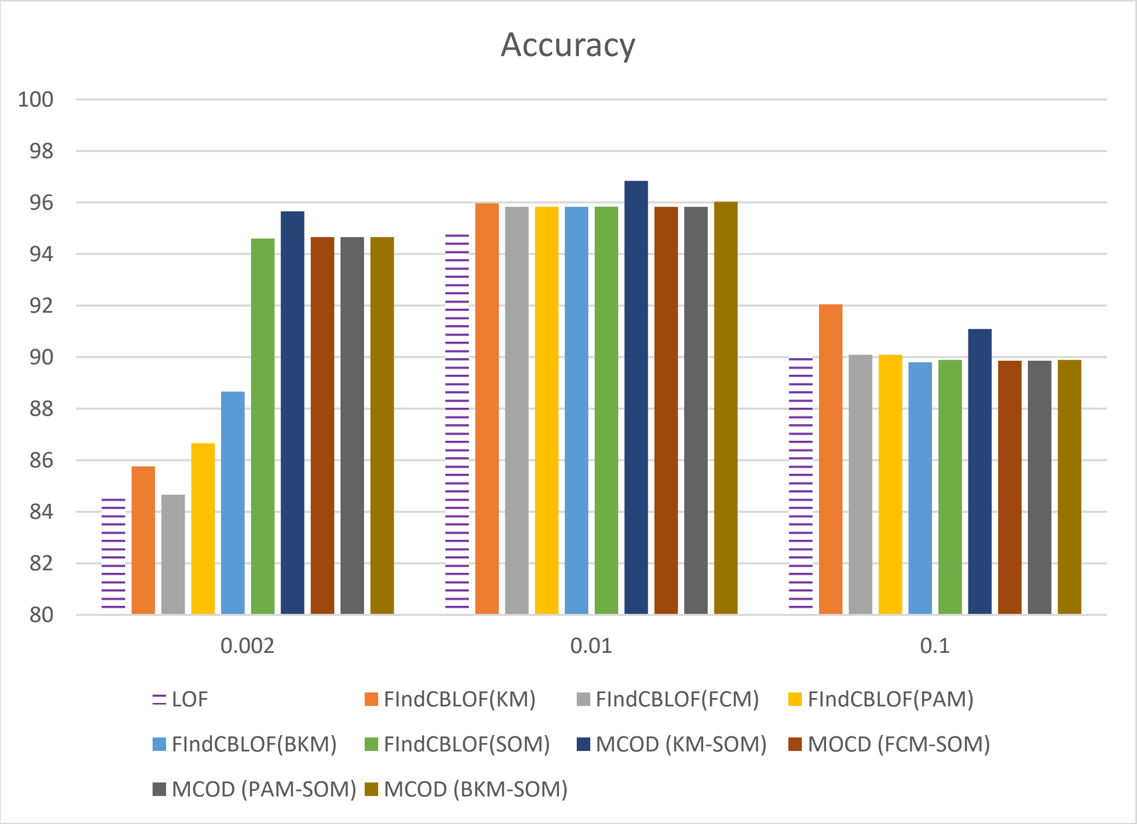

Figure 5. Detection Accuracy (European Credit Dataset).

5. Conclusions and Future Directions

Outliers detection plays a significant role in many business applications, including finance, healthcare systems, face recognition, and many others. In this paper, a multi-level clustering-based outlier detection technique (MCOD) is proposed. This approach is based on using two stages to finally assign an outlier risk factor (ORF) to each data point and recognizing the set of final outliers by the high value of ORF. The MCBOD algorithm relies on the fact that the clustering method in the first layer can provide the knowledge about the dataset as the initial seeds to the second layer to achieve better detection of outliers in the dataset. The MCOD is applied to artificial datasets, biomedical datasets, and credit card datasets with different degrees of outliers and instances of fraud. The undertaken experimental results indicate that the MCOD attains better outlier detection performance than the traditional outlier detection techniques measured by the detection accuracy and the area under the PR curve for various datasets with different configurations and types. As outliers and fraud detection play an important role in many business applications, including credit card fraud detection, anomalies records in healthcare, and many others, the MCOD can be used as an automatic tool to find and discover those outliers with high efficacy and better detection accuracy. Various enterprises and organizations can adopt the MCOD algorithm to reduce the impact of frauds or outliers in data and increase profit/outcomes. Future directions include adopting different clustering algorithms and datasets with large sizes, dimensions, and different levels of outliers. Performing sensitivity analysis on the parameters used is also recommended for future research work.

Author Contributions: Conceptualization, M.L., R.K. and A.I.; methodology, M.L., R.K. and A.I.; software, M.L., R.K.; validation, M.L., R.K. and A.I.; formal analysis, M.L., R.K. and A.I.; resources, M.L., R.K. and A.I.; data curation, M.L., R.K. and A.I.; writing—original draft preparation, M.L., R.K. and A.I.; writing—review and editing, M.L., R.K. and A.I.; visualization, M.L. and R.K.; supervision, R.K.; project administration, R.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ryerson University, Faculty of Engineering and Architectural science.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kashef, R.; Gencarelli, M.; Ibrahim, A. Classification of Outlier's Detection Methods Based on Quantitative or Semantic Learning. In *Combating Security Challenges in the Age of Big Data. Advanced Sciences and Technologies for Security Applications*; Fadlullah, Z., Khan Pathan, A.S., Eds.; Springer: Cham, Switzerland, 2020.
2. Malini, N.; Pushpa, M. Analysis on credit card fraud identification techniques based on KNN and outlier detection. In Proceedings of the 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, India, 27–28 February 2017; pp. 255–258.
3. Rajeswari, N.; Nachammai, S.; Jemima, P.E.; Rajeswari, A.M. Unexpected Health Issues Prediction in Medical Data Using Apriori Rare Based Outlier Detection Method. In Proceedings of the 2019 International Conference on Vision towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 30–31 March 2019; pp. 1–6.
4. Kumar, M.; Mathur, R. Unsupervised outlier detection technique for intrusion detection in cloud computing. In Proceedings of the International Conference for Convergence for Technology-2014, Pune, India, 6–8 April 2014; pp. 1–4.
5. Zheng, L.; Hu, W.; Min, Y. Raw Wind Data Preprocessing: A Data-Mining Approach. *IEEE Trans. Sustain. Energy* **2014**, *6*, 11–19. [CrossRef]
6. Khezrimotlagh, D.; Cook, W.D.; Zhu, J. A nonparametric framework to detect outliers in estimating production frontiers. *Eur. J. Oper. Res.* **2020**, *286*, 375–388. [CrossRef]
7. Schnepper, T.; Klamroth, K.; Stiglmayr, M.; Puerto, J. Exact algorithms for handling outliers in center location problems on networks using k-max functions. *Eur. J. Oper. Res.* **2018**, *273*, 441–451. [CrossRef]
8. Erkuş, E.C.; Purutçuoğlu, V. Outlier detection and quasi-periodicity optimization algorithm: Frequency domain based outlier detection (FOD). *Eur. J. Oper. Res.* **2020**. [CrossRef]
9. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **1982**, *43*, 59–69. [CrossRef]
10. Aggawal, C. Proximity-Based Outlier Detection. In *Outlier Analysis*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 101–133.
11. Knox, E.M.; Raymond, T.N. Algorithms for mining distance-based outliers in large datasets. In Proceedings of the International Conference on Very Large Data Bases, San Franciso, CA, USA, 24–27 August 1998.
12. Dang, T.T.; Ngan, H.Y.; Liu, W. Distance-based k-nearest neighbors outlier detection method in large-scale traffic data. In Proceedings of the 2015 IEEE International Conference on Digital Signal Processing (DSP), Singapore, 21–24 July 2015; pp. 507–510.
13. Domingues, R.; Filippone, M.; Michiardi, P.; Zouaoui, J. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognit.* **2018**, *74*, 406–421. [CrossRef]
14. Davies, L.; Gather, U. The Identification of Multiple Outliers. *J. Am. Stat. Assoc.* **1993**, *88*, 782. [CrossRef]
15. Han, J.; Kamber, M.; Pei, J. Outlier Detection. In *Data Mining: Concepts and Techniques*; Elsevier Science: Burlington, NJ, USA, 2012; pp. 543–584.
16. Swersky, L.; Marques, H.O.; Sander, J.; Campello, R.J.G.B.; Zimek, A. On the Evaluation of Outlier Detection and One-Class Classification Methods. In Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, 17–19 October 2016; pp. 1–10.
17. Schubert, E.; Zimek, A.; Kriegel, H.-P. Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection. *Data Min. Knowl. Discov.* **2012**, *28*, 190–237. [CrossRef]
18. Kantardzic, M. *Data-Mining Concepts*; Wiley: Hoboken, NJ, USA, 2011; pp. 1–25.
19. Kriegel, H.-P.; Hubert, M.S.; Zimek, A. Angle-based outlier detection in high-dimensional data. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD 08, Las Vegas, NV, USA, 24–27 August 2008; p. 444. Available online: https://www.dbs.ifi.lmu.de/~{}zimek/publications/KDD2008/KDD08-ABOD.pdf (accessed on 23 September 2020).

20. Ye, H.; Kitagawa, H.; Xiao, J. Continuous Angle-based Outlier Detection on High-dimensional Data Streams. In Proceedings of the 19th International Database Engineering & Applications Symposium—IDEAS '15, Yokohama, Japan, 13–15 July 2015; pp. 162–167. [CrossRef]
21. Pillai, T.R.; Hashem, I.A.T.; Brohi, S.N.; Kaur, S.; Marjani, M. Credit Card Fraud Detection Using Deep Learning Technique. In Proceedings of the 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA), Bombay, India, 29–31 March 2018; pp. 1–6.
22. Roy, A.; Sun, J.; Mahoney, R.; Alonzi, L.; Adams, S.; Beling, P. Deep learning detecting fraud in credit card transactions. In Proceedings of the 2018 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 27 April 2018; pp. 129–134. Available online: https://ieeexplore.ieee.org/document/8374722 (accessed on 23 September 2020). [CrossRef]
23. Raghavan, P.; El Gayar, N. Fraud Detection using Machine Learning and Deep Learning. In Proceedings of the 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 11–12 December 2019; pp. 334–339.
24. He, Z.; Xu, X.; Deng, S. Discovering cluster-based local outliers. *Pattern Recognit. Lett.* **2003**, *24*, 1641–1650. [CrossRef]
25. Kashef, R.; Kamel, M.S. Towards Better Detection of Outliers, International Conference on BioInformatics and BioEngineering. *Biotechno* **2008**, *1*, 149–154. [CrossRef]
26. Yogita, L.; Toshniwal, D. A Framework for Outlier Detection in Evolving Data Streams by Weighting Attributes in Clustering. *Procedia Technol.* **2012**, *6*, 214–222. [CrossRef]
27. Wang, H.; Bah, M.J.; Hammad, M. Progress in Outlier Detection Techniques: A Survey. *IEEE Access* **2019**, *7*, 107964–108000. [CrossRef]
28. Guha, S.; Rastogi, R.; Shim, K. Rock: A robust clustering algorithm for categorical attributes. *Inf. Syst.* **2000**, *25*, 345–366. [CrossRef]
29. Ebbels, T.M. Non-linear Methods for the Analysis of Metabolic Profiles. In *The Handbook of Metabonomics and Metabolomics*; Elsevier BV: Amsterdam, The Netherlands, 2007; pp. 201–226.
30. Wehrens, R. Data Mapping: Linear Methods versus Nonlinear Techniques. *Compr. Chemom.* **2009**, *2*, 619–633.
31. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-Means Clustering Algorithm. *J. R. Stat. Soc. Ser. C* **1979**, *28*, 100. [CrossRef]
32. Savaresi, S.M.; Boley, D. On the performance of bisecting K-means and PDDP. In Proceedings of the 2001 SIAM International Conference on Data Mining, Chicago, IL, USA, 5–7 April 2001; pp. 1–14.
33. Barnett, V.; Lewis, T. *Outliers in Statistic Data*; John Wiley's: New York, NY, USA, 1994.
34. Dunn, J.C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J. Cybern.* **1973**, *3*, 32–57. [CrossRef]
35. Hawkins, D.M.; Bradu, D.; Kass, G.V. Location of Several Outliers in Multiple-Regression Data Using Elemental Sets. *Technometrics* **1984**, *26*, 197. [CrossRef]
36. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; Wiley: Hoboken, NJ, USA, 1987.
37. Aggarwal, C.C.; Sathe, S. Theoretical Foundations and Algorithms for Outlier Ensembles? *ACM SIGKDD Explor. Newsl.* **2015**, *17*, 24–47. [CrossRef]
38. West, M.; Blanchette, C.; Dressman, H.; Huang, E.; Ishida, S.; Spang, R.; Zuzan, H.; Olson, J.J.A.; Marks, J.R.; Nevins, J.R. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 11462–11467. [CrossRef] [PubMed]
39. Personal and Business Banking Services-RBC Royal Bank. Available online: http://www.rbcroyalbank.com/ (accessed on 12 January 2020).
40. Machine Learning Group. Credit Card Fraud Detection, Kaggle, 23 March 2018. Available online: https://www.kaggle.com/mlg-ulb/creditcardfraud/data (accessed on 12 January 2020).
41. Kashef, R. Ensemble-Based Anomaly Detection Using CooperativeLearning. In Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance, PMLR 71, Halifax, NS, Canada, 14 August 2018; pp. 43–55. Available online: http://proceedings.mlr.press/v71/kashef18a/kashef18a.pdf (accessed on 23 September 2020).

42. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
43. Williams, G.; Baxter, R.; He, H.; Hawkins, S.; Gu, L. A comparative study of RNN for outlier detection in data mining. In Proceedings of the 2002 IEEE International Conference on Data Mining, Maebashi City, Japan, 9–12 December 2002.