IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 January 2023, accepted 19 February 2023, date of publication 3 March 2023, date of current version 15 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252004

RESEARCH ARTICLE

Dynamic Construction of Outlier Detector Ensembles With Bisecting K-Means Clustering

RASHA RAMADAN Z. KOKO[1,2,3], **INAS A. YASSINE**[2], **MANAL ABDEL WAHED**[2], **JUNE K. MADETE**[3], **AND MUHAMMAD A. RUSHDI**[2,4]

[1]National Public Health Laboratory, Bioinformatics Unit, Ministry of Health, Khartoum 187, Sudan
[2]Department of Biomedical Engineering and Systems, Cairo University, Giza 12613, Egypt
[3]Department of Electrical and Electronic Engineering, Kenyatta University, Nairobi 43844, Kenya
[4]School of Information Technology, New Giza University, Giza 12256, Egypt

Corresponding author: Rasha Ramadan Z. Koko (rasha.koko72@eng1.cu.edu.eg)

ABSTRACT Outlier detection (OD) is a key problem, for which numerous solutions have been proposed. To deal with the difficulties associated with outlier detection across various domains and data characteristics, ensembles of outlier detectors have recently been employed to improve the performance of individual outlier detectors. In this paper, we follow an ensemble outlier detection approach in which good outlier detectors are selected through an enhanced clustering-based dynamic selection (CBDS) method. In this method, a bisecting K-means clustering algorithm is employed to partition the input data into clusters where every cluster defines a local region of competence. Among the initial pool of detectors, the outputs of the detectors with the most competent local performance were combined through four possible schemes to produce the final OD results. Experimental evaluation and comparison of our method were carried out against four variants of locally selective combination in parallel (LSCP) outlier ensembles. The CBDS-based schemes compare well with the LSCP-based ones on 16 public benchmark datasets and incur considerably lower computational costs. The CBDS method consistently achieved superior average scores of the area under the curve (AUC) of the receiver operating characteristic (ROC), and particularly outperformed the LSCP method on nine of the 16 datasets in terms of the AUC score. In addition, while the CBDS and LSCP methods have similar computational costs on small datasets, the CBDS method achieves significant time savings compared with the LSCP method on large datasets.

INDEX TERMS Bisecting K-means, dynamic detector selection, outlier detection, outlier ensemble.

I. INTRODUCTION

Outliers (or anomalies) are defined as data samples that are significantly different from other observations in their category or group. Usually, outliers exhibit extreme deviation in value and/or in nature compared to surrounding objects. The occurrence of outlier data samples is very rare, where such outliers typically represent less than 5% of the overall data samples [1]. Detecting such outliers is critical in many applications for several reasons. Firstly, these outliers may adversely affect the results of further statistical analysis, and hence the outliers should be removed to boost the data quality [2]. Secondly, the outliers themselves may represent novel observations of special significance (e.g., rare diseases, abnormal behavior, or genetic mutations). Thus, outlier detection methods have been extensively investigated in a wide range of applications such as data mining [3], network intrusion detection [4], credit-card fraud detection, [5], product tampering detection [6], and industrial fault diagnosis [7]. Also, outlier detection methods have been explored for healthcare applications such as early disease prediction [8], brain tumor detection [9], identification of cancerous masses in mammograms [10], and detection of rare genetic expressions [11].

Outlier patterns can be generally categorized into global or local outliers. On the one hand, the global outliers show extreme deviations from all of the available normal data patterns. Indeed, a global outlier occurs somewhere in the

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

feature space well outside the normal data range [12]. On the other hand, a local outlier can be within the normal global data range, but, still, the pattern of such an outlier can be quite contextually different from its local neighbors.

Unlike binary classification tasks, outlier detection is typically an unsupervised task since it is applied to unlabeled data where ground truth is not available. Therefore, it is quite difficult to assess the accuracy of the detection methods. Moreover, these methods can become unreliable with high false-positive and false-negative rates [13]. To alleviate these limitations of the outlier detection methods, ensemble outlier detection (EOD) methods have been recently introduced [13]. In an EOD method, several outlier detectors are aggregated together to enhance the overall outlier detection performance. The outputs of the individual EOD detectors can be fused through different schemes in order to produce a unified outlier score [15]. A key desirable property of an outlier detector ensemble is to have strong diversity among the base detectors. Such diversity can be achieved by introducing different base models, using different training subsets, or different data subspaces. Recent EOD methods considered these factors in order to achieve satisfactory bias-variance trade-off [13].

Moreover, methods for combining detector outputs within an ensemble can account for all detectors or just select a subset of detectors. A recent design approach for outlier detection ensembles is to exploit a large number of potential detectors, identify and discard weak ones, and dynamically select the most competent detectors for producing the final ensemble output based on an evaluation of specific local regions. This dynamic selection of detectors was originally developed for dynamic classifier selection (DCS) in supervised learning [16], [17]. For the DCS mechanism, the local performance of each base classifier is evaluated at a local region of the feature space where a query sample is located, and the best top classifiers are selected and employed in making the final classification decision for that data sample. This DCS mechanism has been recently extended to the construction of outlier detection ensembles. To compensate for the absence of ground-truth data for outlier detector training, novel approaches for simulating ground-truth labels have been introduced [18], [19].

Remarkably, Zhao et al. [20] proposed an enhanced method for combining base outlier detectors within an unsupervised outlier ensemble. Following the DCS approach, the authors introduced the method of locally selective combination in parallel (LSCP) outlier ensembles. Specifically, a local region is defined around a test data sample based on the consensus of the nearest neighbors of this instance in randomly selected feature subspaces. The base detectors with the best performance in this local region are selected and their outputs are fused to generate the final output of the LSCP ensemble. However, the LSCP method is one of the feature subspace techniques which may generate repeated data samples. This problem causes ensemble performance degradation in addition to high running times [13]. Also, although the k-nearest-neighbor (kNN) scheme contributed prominently to enhancing the dynamic selection of the outlier detectors in the LSCP method, the kNN scheme generally leads to high computational complexity and degraded performance in high-dimensional feature spaces. These drawbacks demonstrate the need for exploring more robust computationally efficient techniques for dynamic detector selection and fusion within outlier detector ensembles.

Our work is motivated by the still-unmet need for real-time high-performance outlier detection methods, especially with large data sets and extremely low occurrences of outlier samples. In this paper, we propose a clustering-based dynamic selection (CBDS) outlier detector ensemble method. In this method, the bisecting K-means clustering algorithm is applied to partition the input data into clusters (subsets). Each cluster delineates a local region of competence. An initial pool of detectors is trained over all these different clusters, and the outputs of the most competent detectors are combined through four possible schemes to produce the final outlier detection results. Employing the bisecting K-means clustering algorithm results in reduced computational complexity of the subsequent outlier detection operations. Also, training base detectors over different data clusters reduces the variance. While a test sample is compared against k nearest neighbors in the LSCP method, a test sample in the CBDS method is compared against a much smaller number of cluster centroids. This results in a much lower computational cost compared to the LSCP method. As well, the bisecting K-means clustering approach produces clusters of similar sizes and this enhances the cluster balance.

In a nutshell, our paper proposes a new method for outlier detection using ensembles of outlier detectors. In comparison to the state-of-the-art LSCP method [20], our method is distinguished by the use of a modern variant of K-means clustering (namely, the bisecting K-means clustering method). In fact, our method compares a test sample with samples within the closest cluster, while the LSCP method compares a test sample with similar ones through a kNN scheme. Thus, our method offers more robustness and shows better performance (compared to the LSCP method) on a wide array of datasets with a significant reduction in computational cost.

Our paper has specifically the following four contributions:

- A clustering-based dynamic selection (CBDS) outlier detector ensemble method is introduced with a novel clustering mechanism based on the bisecting K-means clustering algorithm.
- Two approaches have been proposed for promoting detector ensemble diversity at the model level (through employing base models with a wide range of hyper-parameter settings), and the data-sample level (through trying different subsets of training samples)
- The influence of high levels of data imbalance on the outlier detection performance has been experimentally investigated.
- The outlier detection performance for low false-alarm ranges has been analyzed and assessed.

The rest of this paper is organized as follows. Related work is further explored in Section II. The details of the proposed method are introduced in Section III. The experimental setup (including the datasets, tools, and performance metrics used) is described in Section IV. The experimental results are extensively described in Section V. Detailed discussions of different aspects of the obtained results are provided in Section VI. Finally, conclusions, practical implications, and future research directions are pointed out in Section VII.

II. RELATED WORK

A. CONVENTIONAL OUTLIER DETECTION METHODS

Over the past few decades, many outlier detection methods have been introduced including distance-based, density-based, clustering-based and probabilistic methods [21]. First of all, the distance-based methods identify outliers by calculating pairwise distances for the available data samples. A sample whose average distance from the majority of the other data samples exceeds a certain pre-specified threshold is labeled as an outlier [22], [23]. Angiulli et al. [24] followed this approach to identify outliers in a high-dimensional space. Tran et al. [25] developed an algorithm, named core point outlier detection (CPOD), to speed up the identification of inliers and reduce neighbor search spaces for outlier candidates using a multi-indexed distance method with the core point data structure technique.

Density-based methods detect local outliers. The notion behind the method is that objects in high-density regions are inliers while those in low-density regions are considered outliers [26]. Tripathi et al. [27] employed the estimated high and low densities through a nearest-neighborhood-function method for credit-card fraud detection. The degree of outlierness of an object is specified by the local outlier factor (LOF) of the object [28]. Xu et al. [29] proposed a LOF model with tuned hyperparameters for outlier detection.

In the clustering-based methods, an object either belongs to a large cluster or is just an outlier. Outlier objects are then collected in a small-size outlier cluster [30]. Loureiro et al. [31] used hierarchical clustering in a data cleaning application in order to select a small subset of suspicious data samples and isolate these samples in small outlier clusters. Al-Zoubi et al. [32] identified small clusters and considered them outlier clusters using an objective function based on the fuzzy C-means algorithm (FCM). Bhowate and Gadicha [33] applied the bisecting K-means algorithm to cluster data into K clusters, and prune any cluster sample whose distance from the cluster centroid is less than a pre-specified cluster radius. The unpruned samples are labeled as suspected outliers.

Furthermore, in probabilistic outlier detection methods, the underlying data distribution is inferred, and the outlierness probability is estimated for each data sample. The most commonly used model for univariate data is the normal distribution. The objects are considered outliers if they lay in extreme regions with probability densities below a particular threshold [34], [14]. However, the methods based on univariate probabilistic models are of limited applicability as most of the real-world outlier detection problems involve multi-dimensional data. Moreover, prior knowledge is needed for building reliable probabilistic models. Barghash et al. [35] followed a probabilistic approach for detecting outliers in gene expression datasets. Peter et al. [36] used simple statistical methods for detecting spatial data outliers using the Google Earth Engine. Cho et al. [37] developed the OutlierD software package for outlier detection based on linear, non-linear and non-parametric quantile regression techniques.

B. ENSEMBLE OUTLIER DETECTION METHODS

In general, none of the conventional outlier detection methods remarkably outperforms the others. This can be mainly ascribed to the variabilities in the data distribution, data dimensionality, and data association patterns. In fact, an outlier detection method might perform well in a particular problem with a specific type of data, while poor performance is observed with the same detector for different problems and data types. This observation agrees with the no-free-lunch theorem that asserts the fact that no optimal outlier detector can be obtained for all problems and data types [38]. This limitation of individual outlier detection methods has been recently alleviated by the EOD methods [13]. As mentioned above, an EOD method employs several outlier detectors, of the same type or different types. These detectors are constructed with different hyperparameters, and combined together to enhance the overall outlier detection performance. The diversity within the outlier detector ensembles makes them applicable for different data types and characteristics.

Numerous methods have been proposed for fusing the outputs of individual detectors within an ensemble. These methods can be generally categorized into sequential ensemble combination methods, independent ensemble combination methods, and hybrid ensemble combination methods [14]. In a sequential ensemble combination method, e.g., a boosting-based method, the output scores of the base detectors are interdependent and the final output score is based on the score of the last base detector [39], [40]. For a parallel ensemble combination method, the score of each detector is independently obtained, and then the collected scores are employed to produce the final output score [41]. For a hybrid ensemble combination, both sequential and independent ensemble combinations are compatibly used [42]. A summary of state-of-the-art methods for each category is given in Table 9.

As mentioned earlier, a key design aspect of an outlier detection ensemble is to generate a large pool of potential detectors, identify and retain the most competent ones for computing the final output, and discard the other detectors. Generally, ensembles can be classified into three types in terms of the detector selection approach: generic and global (GG) ensembles in which all detectors are combined to produce the final output, static and global (SG) ensembles in

TABLE 1. Data characteristics of 16 ODDS datasets used for evaluating the performance of ensemble outlier detectors.

Dataset	*# Samples*	*Dimension*	*# Outliers*	*% Outliers*
Arrhythmia	452	274	66	14.6
Breastw	683	9	239	34.9
Cardio	1831	21	176	9.61
Glass	214	9	9	4.2
Ionosphere	351	33	127	36
Mnist	7603	100	700	9.21
Musk	3062	166	97	3.17
Pima	768	8	268	34.9
Optdigits	5216	64	150	3
Satellite	6435	36	2036	31.64
Satimage-2	5803	36	71	1.22
Shuttle	49097	9	3511	7.15
Thyroid	3772	6	93	2.47
Vertebral	240	6	30	12.5
Vowels	1465	12	50	3.43
WBC	378	30	21	5.56

which detectors are selected during the training phase [17] and, most recently, dynamic local (DL) selection ensembles [16]. For the DL selection ensembles, data locality is considered, and detectors are selected after evaluation for predicting outliers in the test phase [43]. A dynamic selection approach is essentially followed by Zhu et al. [44] for data mining in noisy data streams. Also, Zhao et al. [20] employed dynamic selection in the LSCP method. Furthermore, Bii et al. [45] introduced an adaptive boosting method for outlier detection (ADAHO). In this method, the performance of each base detector is enhanced through an adaptive boosting scheme based on the local domain of competence. Then, the nearest-neighbor detectors of the highest correlation with the test instance are selected, and the detector outputs are combined through conventional fusion methods.

III. PROPOSED APRROACH

A. METHOD DESIGN

In the proposed CBDS outlier detection method, an ensemble of diversified and independent base detectors is initially generated. The training dataset is then partitioned into clusters to define local regions of competence. Then, for each training sample, a ground-truth label is constructed based on the aggregation of scores returned by the base detectors. The performance of the base detectors is evaluated in each cluster. For a given test data sample, competent detectors are dynamically selected and the scores of these detectors are combined to produce the final output score for the test sample. The proposed method is shown in Figure 1, and the algorithmic steps are summarized in Algorithm 1. The details of the proposed method are as follows.

1) SCORING OF TRAINING SAMPLES DATA OUTLIERS WITH BASE DETECTORS

In order to obtain enhanced outlier detection performance, a large pool of detectors with different characteristics could be employed to form a detector ensemble. This design approach was extensively considered. For example, van Stein et al. [46] searched for local outliers within feature subspaces. Also, Zhao and Hryniewicki [47] investigated the effects of data locality on outlier detection, and used heterogeneous combinations of outlier ensembles. Following this design approach, we propose an outlier detector ensemble that contains a pool of LOF detectors initialized with different parameter settings. Specifically, let R denote the dataset within which outliers are searched for, where the dataset contains N d-dimensional points and d denotes the number of features. Split R into a training set X_{train} with n points and a test set X_{test} with m points (i.e., $N = n + m$). The ensemble generation is carried out as follows

Step 1: Generate a pool of base detectors H_L, where L is the number of detectors.

Step 2: Initialize all base detectors with different parameter settings.

Step 3: Train all base detectors and use the detectors to predict vectors of outliers scores in X_{train}.

Step 4: Normalize each predicted score vector by scaling in the interval [0,1].

Step 5: Form an outlier score matrix $Q(X_{train})$ as follows:

$$Q(X_{train}) = [H_1(X_{train}), H_2(X_{train}), \ldots, H_L(X_{train})]\epsilon\ \mathbb{R}^{n\times L} \tag{1}$$

2) CONSTRUCTION OF LOCAL REGIONS OF COMPETENCE

Constructing the local regions of competence runs in parallel to the ensemble detector generation stage. In the LSCP and ADAHO methods, a region of competence is defined in the neighborhood of a test sample using the K-nearest neighbor (kNN) training samples [13], [47] whereas LSCP uses a feature bagging technique to sample subsets of data. In our work, we follow a different approach where the training patterns are clustered using the bisecting K-means algorithm [48] in order to define a local region of competence for each cluster center (not for each test pattern). The bisecting K-means algorithm initially divides the available training patterns into two clusters in order to minimize the sum of squared errors (SSE) between cluster centers and their associated patterns. This process is iteratively repeated to split each cluster into two based on the SSE criteria [49] until a prespecified cluster count K is reached. This SSE-based bisection process ensures that the resulting centroids are the best representatives of their respective clusters. In our work, the number of clusters K is defined for each dataset based on the elbow method [50]. Specifically, the bisecting K-means algorithm is carried out as follows:

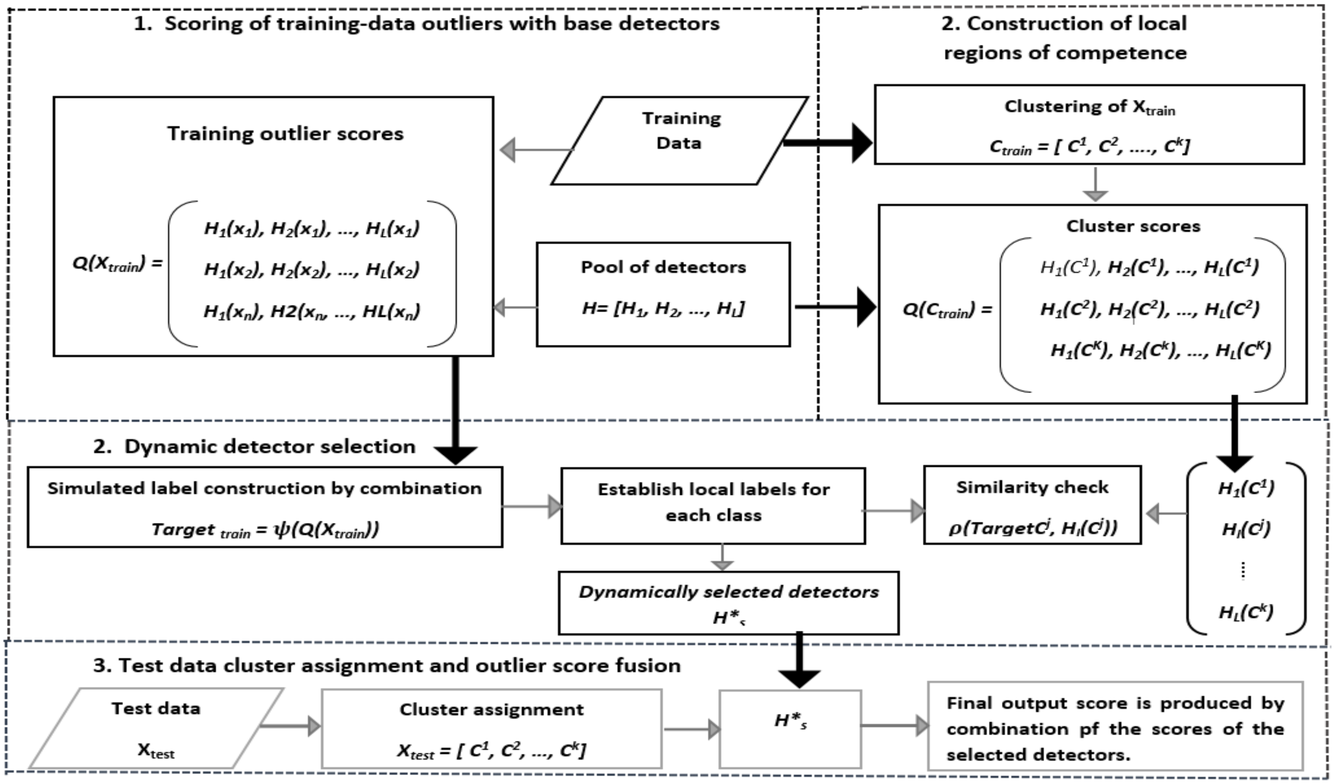
1. Scoring of training-data outliers with base detectors
2. Construction of local regions of competence
Training outlier scores
Training Data
Clustering of X_train
C_train = [C^1, C^2, ..., C^k]
Q(X_train) =
H_1(x_1), H_2(x_1), ..., H_L(x_1)
H_1(x_2), H_2(x_2), ..., H_L(x_2)
H_1(x_n), H2(x_n, ..., HL(x_n)
Pool of detectors
H= [H_1, H_2, ..., H_L]
Cluster scores
Q(C_train) =
H_1(C^1), H_2(C^1), ..., H_L(C^1)
H_1(C^2), H_2(C^2), ..., H_L(C^2)
H_1(C^K), H_2(C^K), ..., H_L(C^K)
2. Dynamic detector selection
Simulated label construction by combination
Target train = ψ(Q(X_train))
Establish local labels for each class
Similarity check
ρ(TargetC^j, H_i(C^j))
H_1(C^1)
H_i(C^j)
H_L(C^k)
Dynamically selected detectors
H*_s
3. Test data cluster assignment and outlier score fusion
Test data
X_test
Cluster assignment
X_test = [C^1, C^2, ..., C^k]
H*_s
Final output score is produced by combination pf the scores of the selected detectors.

FIGURE 1. A flowchart of the proposed CBDS method where the blocks with dashed borders correspond to different stages.

Step 1: Initialize the clusters of the training data X_{train} into two clusters and compute their centroids as

$$u^j = \frac{1}{n}\sum_{x\in C^j} x. \tag{2}$$

Step 2: Based on the SSE criteria, divide a given cluster into two clusters if the SSE after division is smaller than that before division. Otherwise, leave the cluster undivided.

Step 3: Repeat Step 2 for each cluster until K clusters are formed.

Step 4: For each cluster center u_i, denote the set of associated training patterns by C^j, where $j \in \{1, 2, .., K\}$. The collection of all clusters $C_{train} = \{C^1, C^2, .., C^K\}$ partition the original training data X_{train}. Let $U = [u_1, u_2, .., u_K]$ represent the set of the K centroids.

Step 5: Apply the pool of the base detectors to the clusters to produce a matrix $Q(C_{train})$ which lists the scores of the L detectors for each cluster as follows

$$Q\left(C_{train}\right) = \left[H_1\left(C^1_{train}\right), \ldots, H_L\left(C^K_{train}\right)\right] \in \mathbb{R}^{K\times L}. \tag{3}$$

3) DYNAMIC DETECTOR SELECTION

In the previous stage, scores were computed for all detectors in a given cluster. However, not all of the detectors are at the same level of competency in detecting the outliers. Incompetent detectors may negatively affect the overall ensemble-based detection performance. Thus, in this stage, we quantitatively evaluate the competency of the detectors associated with each cluster in order to rank the detectors by competency and select only the most competent ones. Hence, each cluster shall have its specific set of competent detectors which could be used to predict outliers in test data and produce a combined outlier score for each test sample. However, in unsupervised machine learning techniques (like the one in hand), there are no ground-truth labels that can be used for evaluating the accuracy of each detector [51]. Consequently, for the purposes of dynamic detector selection, we have to firstly simulate such labels from the training data. Thus, for each cluster, the most competent detectors are identified and retained. The steps of this dynamic detector selection process are as follows:

a: GROUND-TRUTH SIMULATION

Ground-truth outlier score simulation was proposed by Rayana and Akoglu [18]. Specifically, the outlier score matrix $Q(X_{train})$ for the training data X_{train} is used to generate a set T_{train} of corresponding simulated ground-truth outlier scores for the training samples. The ground-truth scores can be obtained using two methods for aggregating the outlier scores of base detectors:

Averaging (Avg): For a given training sample, the simulated ground-truth score is calculated as the arithmetic average of the scores obtained using the base detectors. For the whole training set, the vector of the simulated ground-truth scores is given by

$$T_{train-avg} = \frac{1}{L}\sum_{l=1}^{L} H_l(X_{train}) \in \mathbb{R}^{n\times 1} \tag{4}$$

where H_l denotes the l^{th} base detector, $l \in \{1, 2, \ldots, L\}$.

Maximization (Max): For a given training sample, the simulated ground-truth score is calculated as the maximum of the scores obtained using the base detectors. For the whole training set, the vector of the simulated ground-truth scores is thus given by finding the maximum scores across all base detectors

$$T_{train-max} = Max\left\{H_1\left(X_{train}\right), \ldots, H_L\left(X_{train}\right)\right\} \in \mathbb{R}^{n\times 1}. \tag{5}$$

In general, we denote the ground-truth generation operator by ψ

$$T_{train} = \psi\left(Q\left(X_{train}\right)\right) \in \mathbb{R}^{n\times 1} \tag{6}$$

where the operator ψ denotes either the average or maximization operators as shown in Eqs. (4), (5), respectively.

b: DETECTOR SELECTION

After ground-truth generation, the competent detectors are identified and selected as follows.

Step 1: For each cluster $C^j, j \in \{1, 2, .., K\}$, let T_{C^j} denote the set of the ground-truth outlier scores of the training samples associated with C^j,

$$T_{C^j} = \left\{T_{train}(x_i)\,|\,x_i \in C^j_{train}\right\} \in \mathbb{R}^{n_j\times 1}, \tag{7}$$

where $\sum_j n_j = n$.

Step 2: For each cluster C^j, compute the Pearson correlation between the set of the ground-truth scores T_{C^j} and scores obtained by each of the L detectors:

$$\rho\left(T_{C^j,} H_l\left(C^j_{train}\right)\right) \tag{8}$$

where $j \in \{1, 2, .., K\}$ and $l \in \{1, 2, \ldots, \mathrm{L}\}$.

For a given cluster, the most competent base detectors are the ones with the highest correlation values. Indeed, detector competency is evaluated using the Pearson correlation similarity measure (instead of a direct accuracy measure) because of the lack of reliable ways to establish binary labels in unsupervised learning [47].

Step 3: For each cluster C^j, the top L_c detectors with the highest correlation values are retained as the competent detectors of this cluster, where $L_c < L$. We denote this set of competent detectors of C^j by $H^{j*} = \{H_l | l$ is the index of one of the top L_c detectors for $C^j\}$. Note that selecting one detector for one cluster can be highly error-prone even if it is the most competent one [47] [15].

4) TEST DATA CLUSTER ASSIGNMENT AND OUTLIER SCORE FUSION

Given a test sample X_t which belongs to X_{test}, we carry out the following steps:

Step 1: Calculate the distance of the test sample X_t to each centroid u_j using the Euclidean distance function d such that

$$d\left(x_j, u_i\right) = \sqrt{\left(x_j, u_i\right)^2}. \tag{9}$$

Step 2: Assign the test sample X_t to the closest cluster C^{OPT} in terms of the computed Euclidean distances:

$$OPT = \underset{j\in\{1,2,\ldots,K\}}{\operatorname{argmin}}\ d(X_t, u_j). \tag{10}$$

Note that the time complexity for assigning each test sample to its best cluster is $O(Km)$ where K is the number clusters, and m is the number of test samples.

Step 3: The L_c competent detectors associated with the closest cluster C^{OPT} are used to score the test pattern X_t, obtaining an outlier score vector

$$Q\left(X_t\right) = \left[H_j\left(X_t\right)\right]_{H_j \in H^{j*}} \in R^{L_c\times 1}. \tag{11}$$

Step 4: The outlier scores calculated using Eq. (11) are fused using one of different possible score-based fusion methods to obtain the final outlier score of the test sample X_t. We prefer these methods to rank-based fusion methods, as the former provide clear quantitative measures of the degree of outlierness [15]. In our work, we explore some of the following score-based fusion methods:

- **Averaging (Avg):** Average the scores of the competent base detectors as the final outlier score of a test sample.
- **Maximization (Max):** Return the maximum outlier score across the competent detectors for a test sample.
- **Maximum of Average (MOA):** The competent detectors are divided into subgroups of nearly equal sizes. For each detector subgroup, the maximum score is found. Then, the final score is computed as the average of the subgroup scores.
- **Average of Maximum (AOM):** The competent detectors are divided into subgroups of nearly equal sizes. For each detector subgroup, the average score is found. Then, the final score is computed as the maximum of the subgroup scores.

Using the above-mentioned fusion methods, we explore four variants of the proposed CBDS method: CBDS-Avg, CBDS-Max, CBDS-MOA, and CBDS-AOM. The steps of the proposed CBDS algorithm are summarized in Algorithm 1.

IV. EXPERIMENTAL SETUP

A. DATASETS

The proposed approach was evaluated using 16 real-world multidimensional datasets of the annotated Outlier Detection Datasets (ODDS) [52]. The used datasets come from different fields, and have large variations in the number of samples, the number of features (or dimensionality), and the number (or percentage) of outliers. A summary of these data characteristics is shown in Table 1 for each of the 16 datasets. Further performance evaluation was also conducted on 3 highly-imbalanced datasets with minority class percentages below 0.2%. These datasets are *Train* [53], *Creditcard* [54], and

Algorithm 1 Outlier Detection With CBDS

```
Input: Pool of $H$ detectors; Training data $X_{train}$; Test data $X_{test}$
Output: Outlier scores of the test data $X_{test}$

    train all detectors in $H$ on the training data $X_{train}$
    get the outlier scores for all training patterns using the
        detectors in $H$ to form the outlier score matrix $Q(X_{train})$
    get the simulated ground-truth scores $T_{train}$
    partition $X_{train}$ into $K$ clusters $C_{train}$
    for each cluster $C^j$ in $C_{train}$ do
        compose the set of cluster-specific ground-truth
        scores $T_C^i$ from $T_{train}$
        apply the pool of detectors $H$ to $C^j$
        get the outlier score matrix $Q(H_L(C^j))$ for $C^j$
        # identify the competent detectors for $C^j$
        for score $H_l(C^j)$ in $Q(H_L(C^j))$ do
            compute the similarity of $(TC^j, H_l(C^j))$
            return the most competent detectors $H^{j*}$ for $C^j$
        end for
    end for
    for each test pattern $X_t$ in $X_{test}$ do
        compute the distance between $X_t$ and each centroid
        $u^j$ using Eq. (10)
        assign $X_t$ to the nearest cluster $C^{opt}$
        apply all the competent detectors $H^{j*}$ associated with $C^{opt}$
            to get multiple detector-specific outlier scores for $X_t$
    end for
    if Avg then                # Select a score fusion method
        return $MOA(H^{j*}(X_t))$
    Else
        return $AOM(H^{j*}(X_t))$
    end if
```

TABLE 2. Data characteristics of highly-imbalanced datasets used for evaluating the performance of outlier detectors.

Dataset	# samples	Dimension	# outliers	% outliers
Train	19084	16	10	0.052
creditcard	284,807	28	492	0.172
Fraud_data	1048576	11	1142	0.108

Fraud_data [55] with corresponding minority-class percentages of 0.052%, 0.172%, and 0.108%, respectively. The characteristics of these three datasets are summarized in Table 2.

B. EVALUATION METRICS

In this study, the outlier detection performance was evaluated using the receiver operating characteristic curve (ROC) and the area under the ROC curve (AUC). Further evaluation metrics (for specific setups) are the F1-score, precision, and recall. Moreover, we compared the computational complexities of the proposed CBDS and LSCP methods.

C. EXPERIMENTAL SETTINGS

Experiments were employed to evaluate the outlier detection performance outcomes of the proposed CBDS method, and compare these outcomes against those of the LSCP method. Comparisons are made in the training stage (where the simulated ground-truth is generated using the *Avg* and *Max* aggregation operators), and in the testing stage (where outliers are predicted in the test samples using the *MOA* and *AOM* aggregation operators). The results for both the training and testing stages are reported for the 16 ODDS datasets and the 3 highly-imbalanced datasets, as listed in Table 3 and Table 7, respectively. A pool of 20 LOF detectors was employed in the training stage, where each detector is assigned a distinct minimum number of points to form a dense region (*MinPts*). This *MinPts* parameter is picked from the set {5, 20, 30, 40,..., 190, 200} in proportion to the dataset size. The implementation of the proposed algorithm and comparisons with other methods were carried out in Python. In particular, the Python Outlier Detection toolkit (PyOD) [56] and the scikit-learn toolkit [57] were used to run experiments on a laptop with an Intel® Core^{TM} CPU with two 2.67-GHz processors and 8-GB RAM.

V. RESULTS

A. QUANTITATIVE OUTLIER DETECTION RESULTS

Table 3 shows the AUC scores of the CBDS and LSCP methods for the *Avg*, *Max*, *MOA,* and *AOM* aggregation operators. For each of the 16 ODDS datasets, we show the highest AUC score in bold. In general, as shown by the last row of Table 3, the average AUC scores of the CBDS schemes are consistently higher than the corresponding LSCP ones, where the average scores are taken over the 16 ODDS datasets.

As can be observed, both the CBDS-Avg and LSCP-Avg methods demonstrate relatively weak performance. The potential explanation for this is that the averaging outcomes can be adversely affected by contributions from relatively incompetent detectors. For the LSCP-Avg variant, no adequate bias-variance balance can be maintained, and this leads to deteriorated detection performance. Indeed, although the *Avg* fusion method theoretically leads to reduced variance and higher local competency, selecting a relatively small number of detectors may practically result in weaker variance reduction and higher bias because of the heuristic ground-truth simulation method. The CBDS-Max method returned the highest AUC scores on high-dimensional data (e.g. *Ionosphere*), and relatively low-dimensional data (e.g. *Satellite*), while the LSCP-Max scheme achieved the highest AUC on the low-dimensional *Breastw* dataset although schemes with the *Max* operator are generally more robust on high-dimensional data. For the AOM operator, the CBDS-AOM scheme outperformed the LSCP-AOM one on 5 datasets (*Cardio*, *Glass*, *Satimage-2*, *Optdigits*, and *Vertebral*). This result shows that the CBDS-AOM scheme balances bias and variance reduction through the second-stage fusion, and outperforms all other variants. We also notice that the highest CBDS-AOM scores are achieved on high-dimensional datasets (*Cardio*, *Satimage-2*, *Optdigits*) as well as relatively low-dimensional datasets (*Glass* and *Vertebral*), whereas the LSCP-AOM method had the highest AUC scores on four datasets (*Arrhythmia*, *Musk*, *Thyroid*, and *Vowels*), among which only the *Thyroid* one is a low-dimensional dataset. This demonstrates that the CBDS-AOM method has steady outlier

TABLE 3. The AUC scores of the CBDS and LSCP ensemble outlier detectors with four variants of the aggregation operators. For each dataset, the scores are obtained as the averages of ten runs, and the highest score is set in bold.

Dataset	CBDS				LSCP			
	Avg	Max	MOA	AOM	Avg	Max	MOA	AOM
Arrhythmia	0.7206	0.7103	0.7198	0.7232	0.7241	0.7226	0.7236	**0.7268**
Breastw	0.6203	0.7013	0.6217	0.6348	0.6561	**0.7123**	0.6752	0.6766
Cardio	0.7497	0.7708	0.7668	**0.7759**	0.7331	0.7552	0.7488	0.7446
Glass	0.7778	0.8772	0.8591	**0.8778**	0.7378	0.7317	0.7287	0.7348
Ionosphere	0.8889	**0.9762**	0.8916	0.9208	0.9094	0.9057	0.9088	0.9086
Mnist	0.8471	0.83	**0.8561**	0.8411	0.7906	0.7909	0.7912	0.791
musk	0.8787	0.8924	0.8987	0.9082	0.8395	0.8889	0.8953	**0.9208**
Optdigit	0.7322	0.7714	0.7846	**0.7971**	0.4593	0.4619	0.4605	0.46
Pima	0.5402	0.6151	0.6078	0.639	0.6744	0.678	**0.6783**	0.6747
Satellite	0.6542	**0.7198**	0.661	0.6851	0.5828	0.5851	0.5844	0.5835
satimage-2	0.8947	0.9094	0.9062	**0.9201**	0.8795	0.9157	0.901	0.8976
Shuttle	0.6163	0.6315	**0.6643**	0.6045	0.5386	0.5234	0.5297	0.5338
Thyroid	0.942	0.9494	0.9475	0.9443	0.9401	0.9481	0.9496	**0.9497**
Vertebral	0.3772	0.4304	0.4314	**0.4372**	0.4239	0.4305	0.4305	0.428
Vowels	0.9174	0.921	0.9231	0.9262	0.9199	0.9241	0.9238	**0.9285**
WBC	0.7429	0.7593	0.7593	0.6957	0.8811	0.8785	**0.8837**	0.8793
Average	**0.7436**	**0.7791**	**0.7687**	**0.7707**	**0.7306**	**0.7408**	**0.7384**	**0.7399**

TABLE 4. The p-values of the Nemenyi test for comparing the mean AUC values of the 4 CBDS variants (the statistically significant values are shown in bold).

	CBDS_ MOA	CBDS_ AOM	CBDS_ Avg	CBDS_ MAX
CBDS_MOA	1	0.6	**0.02**	0.9
CBDS_AOM	0.6	1	**0.001**	0.7
CBDS_Avg	**0.02**	**0.001**	1	**0.01**
CBDS_ MAX	0.9	0.7	**0.01**	1

TABLE 5. The p-values of the Nemenyi test for comparing the mean AUC values of the 4 LSCP variants (the statistically significant values are shown in bold).

	LSCP_Avg	LSCP_ MAX	LSCP_ MOA	LSCP_ AOM
LSCP-_Avg	1	0.3	0.1	0.1
LSCP-_ MAX	0.3	1	0.9	0.9
LSCP-_ MOA	0.1	0.9	1	0.9
LSCP-_ AOM	0.1	0.9	0.9	1

detection performance regardless of the data dimensionality while the LSCP methods may exhibit unstable performance on low-dimensional datasets. This can be ascribed to the fact that the LSCP methods use feature bagging to create subsets of high-dimensional data samples, and this leads to data sample repetition [51]. Also, the employed LOF detectors show degraded performance robustness with repeated data samples, especially low-dimensional ones. On the other hand, the CBDS methods alleviate these problems through creating highly-separated data clusters. Furthermore, while the CBDS and LSCP scores are relatively close for some datasets (*Arrhythmia*, *Vertebral*, *Vowels*, *WBC*), the differences are quite noticeable for some other datasets (*Optdigits*, *Glass*). It is also noticeable that the proposed CBDS-AOM scheme achieved the highest AUC of 0.4372 on the highly challenging *Vertebral* dataset. For the MOA aggregation operator, the CBDS-MOA achieved the highest AUC scores on the *Mnist* and *Shuttle* datasets, while the LSCP-MOA scheme achieved the highest scores on the *Pima* and *WBC* datasets. In general, the CBDS variants outperform the LSCP ones on 9 (out of 16) datasets. Indeed, all LSCP variants demonstrate remarkably poor performance, compared to the CBDS variants, on the *Optdigits* and *Shuttle* datasets with average AUC differences of 0.3109 and 0.0978, respectively. The CBDS variants show higher variation in AUC values compared to LSCP.

B. STATISTICAL SIGNIFICANCE TESTING

The Friedman test shows that there is a statistically significant difference among the mean AUC values of the four CBDS variants ($\chi^2 = 17.79; p = 4.85 \times 10^{-5} < 0.05$) We also performed the Nemenyi test for pairwise comparisons for testing the four variants of the CBDS. As shown in Table 4, the Nemenyi test revealed statistically significant differences between the average AUC value of CBDS_Avg (AUC=0.7436) and the corresponding values for each of CBDS_MOA ($AUC = 0.7687; p = 0.02$), CBDS_AOM ($AUC = 0.7707; p = 0.001$), and CBDS_MAX ($AUC = 0.7791; p = 0.01$). We notice that the average AUC value for the CBDS_Avg variant is less than the AUC values for

TABLE 6. The p-values of the Nemenyi test for comparing the mean AUC values of the 8 CBDS and LSCP variants (the statistically significant values are shown in bold).

	CBDS_MOA	CBDS_AOM	CBDS_Avg	CBDS_ MAX	LSCP_Avg	LSCP_ MAX	LSCP_ MOA	LSCP_ AOM
CBDS_MOA	1	0.9	0.3	0.9	0.3	0.9	0.9	0.9
CBDS_AOM	0.9	1	**0.01**	0.9	**0.01**	0.6	0.7	0.8
CBDS_Avg	0.3	**0.01**	1	0.1	0.9	0.6	0.5	0.5
CBDS_ MAX	0.9	0.9	0.1	1	0.1	0.9	0.9	0.9
LSCP_Avg	0.3	**0.01**	0.9	0.1	1	0.6	0.5	0.5
LSCP_ MAX	0.9	0.6	0.6	0.9	0.6	1	0.9	0.9
LSCP_ MOA	0.9	0.7	0.5	0.9	0.5	0.9	1	0.9
LSCP_ AOM	0.9	0.8	0.5	0.9	0.5	0.9	0.9	1

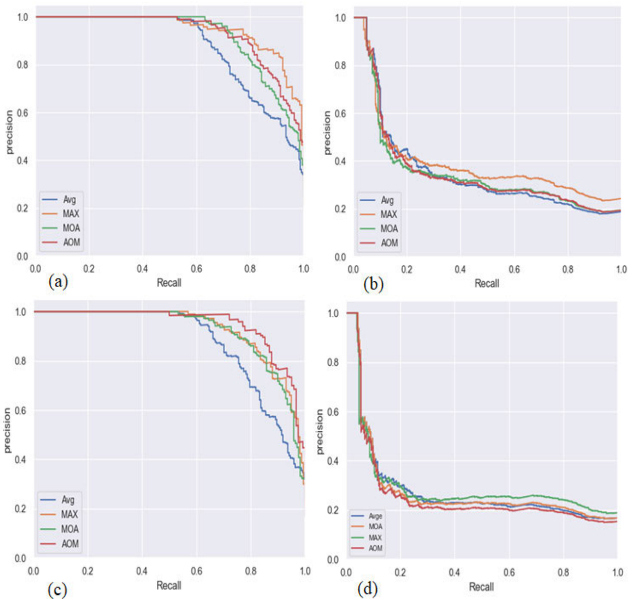

FIGURE 2. Precision-recall curves for the CBDS and LSCP variants: (a) CBDS with the Thyroid dataset, (b) CBDS with the Vertebral dataset, (c) LSCP with the Thyroid dataset, and (d) LSCP with the Vertebral dataset.

all other variants with statistically significant margins. The Friedman test revealed that there is no statistically significantly difference among the mean AUC values of the four LSCP variants ($\chi^2 = 6.396$; p=0.09).

Furthermore, the post-hoc Nemenyi test couldn't identify any pairs of the 4 variants with statistically significant differences (See Table 5). Moreover, we used the Friedman test to investigate whether a statistically significant difference exists among the 8 variants of the CBDS and LSCP methods. The test indicates that there is a statistically significant difference in terms of the mean AUC values ($\chi^2 = 20, p = 5.4 \times 10^{-3} < 0.05\chi^2 = 20$, p = 5.4 × 10^{-3} < 0.05). We also performed the Nemenyi test for pairwise comparisons among the 8 variants. Table 6 shows the resulting *p*-values for all variant pairs. From Table 6, significant statistical differences are present for two pairs of methods. First, the difference between the average AUC values of CBDS_AOM (AUC=0.7707) and CBDS_Avg (AUC=0.7436) is statistically significant with a *p*-value of 0.01. Also, the difference between the average AUC values of CBDS_AOM (AUC=0.7707) and CBDS_Avg (AUC=0.7306) is statistically significant with a *p*-value of 0.01. This latter result statistically confirms the superiority of the CBDS approach over the LSCP one.

C. PRECISION-RECALL CURVES

The precision-recall (PR) curves were constructed for the *Thyroid* dataset (which exhibits the best performance) and the *Vertebral* dataset (which exhibits the worst performance). For the CBDS results on the *Thyroid* dataset (Figure 2(a)) and the *Vertebral* dataset (Figure 2(b)), the CBDS-Max variant outperforms the other CBDS variants. Similar curves based on the LSCP variants are shown in Figure 2(c), (d) for the *Thyroid* and *Vertebral* datasets, respectively. On the one hand, the PR curves for the LSCP variants on the *Thyroid* dataset (Figure 2(c)) look slightly better than those associated with the CBDS variants (Figure 2(a)). On the other hand, the degradation of the PR curves for the *Vertebral* dataset using the CBDS variants (Figure 2(b)) is visibly less severe than the corresponding degradation for the LSCP variants (Figure 2(d)).

D. ROC ANALYSIS UNDER LOW FALSE-ALARM PROBABILITY

In the case of outlier detection, false alarms imply labeling normal samples as outliers. This type of error ultimately affects the quality of the detection results and, therefore, the operating point of an outlier detector should generally be in the low false-alarm range [58]. For example, Figure 3(a) shows the ROC curves of the CBDS variants for the *Thyroid* dataset (for which the highest AUC values were scored by the CBDS variants). Figure 3(b) shows the same curves restricted to a range of low probability of false alarms (PFA) of [0, 0.1]. The ROC curve growth patterns for the proposed CBDS method show consistency among the full and restricted ranges. Also, Figure 3(c), (d) show the ROC curves of the LSCP variants on the *Thyroid* dataset for the whole and restricted low PFA ranges. Obviously, the ROC curves of the LSCP and CBDS methods are mostly similar in the whole false-positive range (with quite similar AUC values as those given in Table 3 for the *Thyroid* dataset). However, for the low PFA range, the CBDS-associated ROC curves (Figure 3(b))

TABLE 7. The AUC scores of the CBDS and LSCP ensemble outlier detectors with four variants of the aggregation operators. For each of the imbalanced datasets, the highest AUC score is shown in bold.

Dataset	*CBDS*				*LSCP*			
	Avg	**Max**	**MOA**	**AOM**	**Avg**	**Max**	**MOA**	**AOM**
Fraud_data	0.4329	0.4542	0.6232	0.6237	0.6384	0.6309	0.6377	**0.6453**
Train	0.321	**0.3718**	0.3621	0.3601	0.2826	0.2884	0.2947	0.2931
Creditcard	**0.5354**	0.5243	0.5219	0.5303	0.4352	0.4628	0.443	0.4288

TABLE 8. Outlier detection metrics of the CBDS and LSCP methods for three simulated datasets of gene expression data.

Class ratio	CBDS				LSCP			
	AUC	Precision	Recall	F-score	AUC	Precision	Recall	F-score
10:90	0.5978	0.5978	0.6484	0.6484	0.5958	0.5724	0.5261	0.4882
35:65	0.7578	0.6904	0.7620	0.7620	0.8366	0.58755	0.6126	0.5968
50:50	0.9263	0.8201	0.8211	0.8205	0.9374	0.6483	0.8899	0.6913

TABLE 9. Comparison between state-of-the-art outlier detection methods.

Authors	Algorithm	Combination method	Selection method	Number of datasets	Highest score	Evaluation metrics
Lazarevic and Kumar [42]	FB	parallel	GG	14	0.869	AUC
Micenková *et al.* [61]	BORE	sequential	SG	3	100	AUC
Rayana *et al.* [18]	SELECT	hybrid	SG	7	0.9245	Accuracy
Rayana *et al.* [19]	CARE	hybrid	SG	16	0.9498	Accuracy
Ouyanget al. [72]	EBOD	sequential	SG	10	0.918	AUC
Stein *et al.* [46]	GLOSS	parallel	GG	6	0.951	AUC
Bii *et al.* [45]	ADAHO	sequential	DL	10	0.9699	AUC
Zhao *et al.* [43]	XGBOD	hybrid	SG	7	0.9999	AUC
Zhao *et al.* [20]	LSCP	parallel	DL	20	0.9981	AUC
Proposed approach	CBDS	parallel	DL	20	0.9762	AUC

are obviously closer to the top-left corner (and thus have higher local AUC values) compared to the LSCP-associated ROC curves (Figure 3(d)).

E. DETECTION PERFORMANCE ANALYSIS WITH T-SNE VISUALIZATION

Figures 4, 5, and 6 visualize the performance of the CBDS and LSCP methods on the *Glass*, *Cardio*, and *Breastw* datasets, respectively. The t-distributed stochastic neighbor embedding (t-SNE) method [59] is employed to reduce the dimensionality of each of these datasets. The ground-truth normal and outlying points are denoted using yellow dots and orange squares, respectively. The points correctly predicted as outliers using the CBDS and LSCP methods are denoted by the blue cross and green delta shapes, respectively. For the *Glass* and *Cardio* datasets, the CBDS method identified most of the outlying points correctly, while the LSCP method exhibited relatively lower performance. The *Breastw* dataset, shown in Figure 6, illustrates that the LSCP approach has better performance than the CBDS one.

F. VISUAL PERFORMANCE ANALYSIS ON THE CLUSTER AND BASE DETECTOR LEVELS

We further examined the performance of the proposed CBDS method for selected samples of the formed clusters and base detectors. Figure 7 illustrates the performance of the base detectors in two different clusters of the *Breastw* dataset before the first fusion step in the training stage. Only the results of three base detectors (named D1, D2, and D3) are shown for simplicity. The three detectors have *MinPts* parameter values of 10, 20, and 30, respectively. On the one hand, the left-side cluster (Figure 7(a)) shows superior performance for the 3 detectors. Actually, detector D2 shows the best outlier detection performance, compared to the two

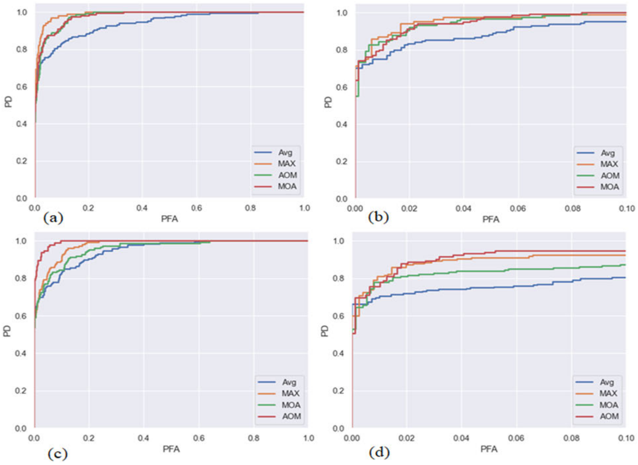

FIGURE 3. ROC plots for the CBDS and LSCP variants: (a) The ROC plots for the CBDS variants over the whole false-alarm range. (b) The ROC plots for the CBDS variants over the zone of low-probability false alarms [0, 0.1]. (c) The ROC plots for the LSCP variants over the whole false-alarm range. (d) The ROC plots for the LSCP variants over the zone of low-probability false alarms [0, 0.1].

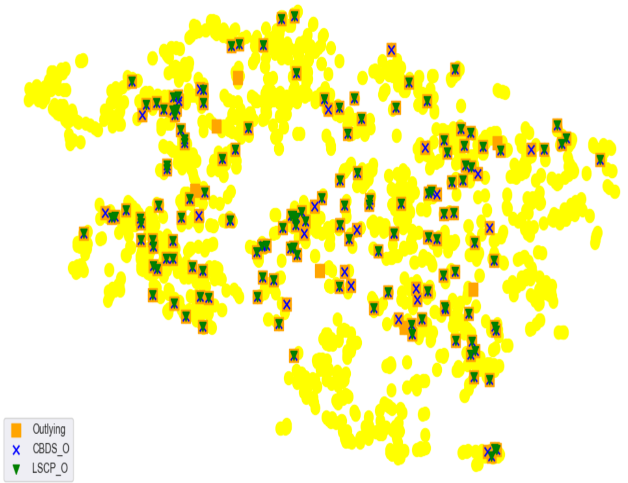

FIGURE 5. T-SNE visualization of the Cardio dataset. The normal and outlying points are shown by yellow dots and orange squares, respectively. Points identified as outliers by the LSCP and CBDS methods are labelled by blue cross and green delta shapes, respectively.

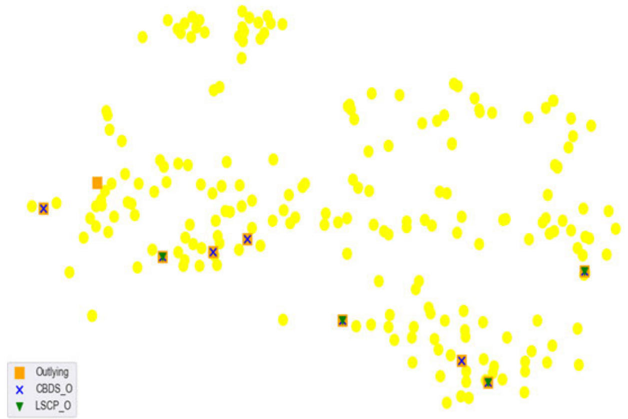

FIGURE 4. T-SNE visualization of the Glass dataset. The normal and outlying points are shown by yellow dots and orange squares, respectively. Points identified as outliers by the LSCP and CBDS methods are labelled by blue cross and green delta shapes, respectively.

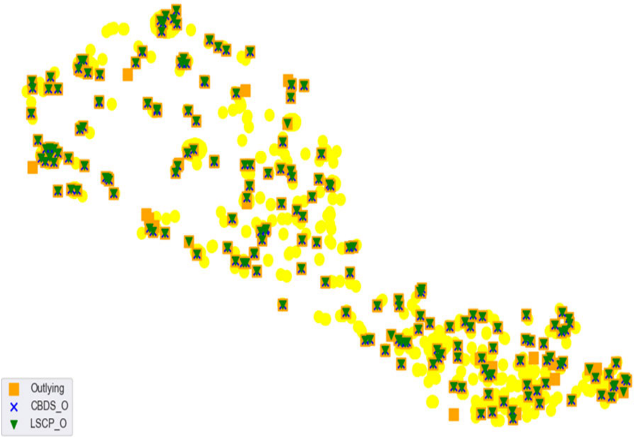

FIGURE 6. T_SNE visualization of Breastw dataset. The normal and outlying points are shown by yellow dots and orange squares, respectively. Points identified as outliers by the LSCP and CBDS methods are labelled by blue cross and green delta shapes, respectively.

other detectors, D1 and D3. On the other hand, the right-side cluster (Figure 7(b)), which is mostly dominated by outlier samples, shows highly degraded performance for the same detectors (with the same *MinPts* parameter values). Most of the outlier samples are clearly missed by the three detectors in this cluster. This obviously strengthens the conclusion that the detector competency can vary widely across clusters and datasets.

Figure 8 shows the outliers detected using the CBDS_AOM variant in each of the 4 clusters in the test stage for the *Cardio* dataset. For clarity of visualization, we show only the true outliers and the correctly detected ones for each cluster. Clearly, most of the ground-truth outliers were correctly detected in each formed cluster of this dataset.

G. DETECTION PERFORMANCE ON HIGHLY-IMBALANCED DATA

Approaches for testing outlier detection performance under data imbalance conditions typically have one of two different perspectives. On the one hand, some approaches treat the minority class of the imbalanced data as the outlier data [60]. On the other hand, other approaches assume that outliers can be present in any class of the imbalanced data (including the majority and minority classes). We performed outlier detection experiments with the two perspectives.

For the first perspective (where the minority-class samples are treated as the only outliers), we conducted experiments with low and extremely-low minority-class percentages (compared to the total dataset size). As shown in

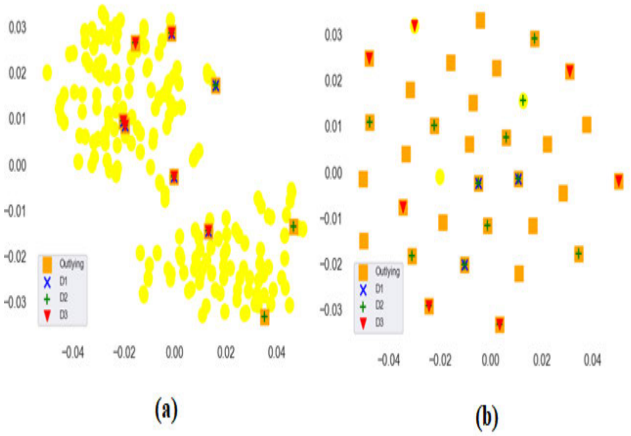

FIGURE 7. Performance of the base detectors within two different clusters of the Breastw dataset, where D1, D2, and D3 represent three LOF detectors with MinPts values of 10, 20, 30, respectively. (a) Superior performance of the three detectors on one cluster, (b) Degraded performance of the same detectors in the other cluster.

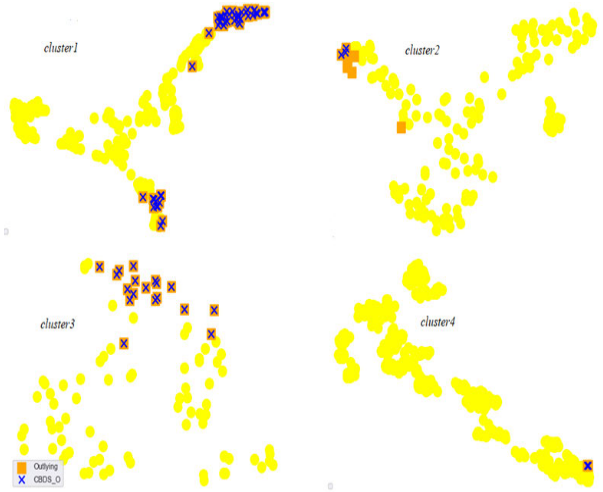

FIGURE 8. T-SNE visualization of the performance of the CBDS_ AOM variant for 4 clusters of the Cardio dataset in the test stage. The normal and outlying points are shown by yellow dots and orange squares, respectively. The outlier points detected by CBDS are labeled by blue crosses.

Section V.1, we conducted outlier detection experiments on 16 datasets with low minority-class percentages (ranging from 1.22% to 36%). We discussed the results of these experiments in detail in Section V.1, and demonstrated the superiority of our method over the LSCP variants. Furthermore, we conducted experiments with extremely low minority-class percentages.

Specifically, we conducted experiments on three datasets with minority class percentages below 0.2. Again, we carried out experiments on these three datasets under the assumption that the outliers are exactly the minority-class samples. Table 7 shows the results for these experiments. Obviously, while both the CBDS and LSCP methods incurred low scores, the performance of the CBDS method is slightly better (possibly due to the smaller cluster sizes in the CBDS method). Moreover, the competent LOF detectors of the CBDS method search for outliers independently in data clusters whereas the LSCP method suffers from the presence of vast majorities of surrounding inlier neighbors that can easily confuse the LSCP outlier detector.

For the second perspective (where outliers can be present in any class of the imbalanced data), we simulated three scenarios of class imbalance in gene expression data of cancer patients and healthy control subjects. Our simulation scheme follows the approach of Barghash and Arslan [35]. For the first scenario, the simulated data resembles a typical cancer dataset of 1000 data samples including 900 samples of healthy subjects (Class 0) and 100 samples of cancer patients (Class 1). The class-specific distributions for Class 0 and Class 1 were assumed to be the normal distributions N$(0,2^2)$ and $N(5,2^2)$, respectively. The number of outliers was set to 50. The outliers were assumed to be present in both classes with 46 outliers (following the normal distribution $N(10,2^2)$) from Class 0 and 4 outliers (following the normal distribution $N(12,2^2)$) from Class 1. The class imbalance ratio in this simulated dataset is 10%:90%. For the other two scenarios, we simulated similar datasets but with two different class ratios of 35%:65% and 50%:50%, respectively. The outlier count and distribution remain the same as those in the first scenario.

Table 8 shows the outlier detection results for the above-mentioned three simulated datasets. As expected, the results show that the performance scores of both the CBDS and LSCP methods degrade strongly with the increase in class imbalance. However, the CBDS-associated degradation is generally less severe than that of the LSCP method. In particular, the precision and recall degradation rates for the LSCP method are evidently worse than those of our CBDS method (although the LSCP method shows comparatively slower degradation of the AUC score).

VI. DISCUSSION

A. COMPUTATIONAL COMPLEXITY ANALYSIS AND PROCESSING TIMES

We show here detailed and overall analyses of the computational complexities of the CBDS and LSCP methods. In these analyses, L denotes the number of detectors, n denotes the number of training samples, m denotes the number of test samples, d denotes the data dimensionality, K denotes the number of CBDS clusters, and S denotes the number of selected top competent detectors. Tabulated summaries of these analyses are included in the Supplementary Materials. For the CBDS method, detector generation and initialization is linear in the number of detectors L. Each of the LOF detectors is trained with a complexity of $O(n^2)$. Then, for each training sample and detector, the detector output is normalized, giving a complexity of $O(Ln)$. The cost of the training data clustering is proportional to the product of the numbers of training samples n, clusters K, and dimensions d. Each cluster is scored by each detector, giving a complexity of *O(KL)*. Thus, a ground-truth label is generated for each training sample based on the outputs of the L detectors, and this gives a complexity of *O(Ln)*. Each training sample is then associated with one cluster (*O(Kn)*). For each cluster, the Pearson correlation is thus computed between the set of the ground-truth scores and scores obtained by each of the L detectors. This gives a complexity of *O(KLn)*. The best detectors associated with each cluster are picked by sorting all detectors based on the computed correlation values ($O(KL \log L)$). For the m test samples, the distance of each sample to each cluster is found (*O(Kdm)*), the closest cluster is identified (*O(Km)*), and then the top competent detectors of that cluster are used to assign an outlier score to the test sample (*O(Sm)*). For the LSCP method, the computational complexities of the base detector initialization, training, and sampling are similar to the corresponding CBDS ones. The ground-truth labels are generated for the training samples based on the outputs of the L detectors, with a complexity of *O(Ln)*. Outlier scoring for the training data is linear in the number of samples ($O(n)$). Then, using a k-d tree, the construction of the local regions requires distance calculations (*O(md)*) in addition to summation and sorting ($O(m \log m)$). The test sample labeling and scoring is linear in the number of samples ($O(m)$). Finally, Pearson correlation is computed for all test samples and detectors (*O(Lm)*), the detectors are sorted ($O(L \log L)$), and the top S detectors are selected ($O(S)$). See the Supplementary Materials for the overall complexities of the CBDS and LSCP methods.

Based on the above calculations, the overall CBDS time complexity includes quadratic and linear functions in n and a linear function in m, while the LSCP time complexity involves a quadratic function in n, as well as logarithmic and linear functions in m. This analysis shows that the overall CBDS running time should be generally lower than the LSCP time.

Figure 9 compares the overall execution times of the CBDS and LSCP variants for the 16 datasets. The y-axis represents the execution time in seconds, while the x-axis represents the ODDS datasets (ordered by the dataset size from the smallest to the largest). For the small-size datasets (*Glass, Vertebral, WBC, Ionosphere, Breastw, Pima*), the CBDS execution times are slightly less than their LSCP counterparts. For large-size datasets, the CBDS execution times are far less than the LSCP ones. For example, for the *Shuttle* dataset, the CBDS schemes had an average execution time of 295.4 seconds whereas the LSCP schemes had a much larger average execution time of 4309.56 seconds. This huge difference in execution time represents a key advantage of the CBDS approach, which still gives good outlier detection performance.

B. COMPARISON WITH RELATED METHODS

We show in Table 9 a comparison of our work against several related methods. For each method, the table shows the employed ensemble fusion technique, the detector selection mechanism, the number of explored datasets, the highest result achieved (on any dataset), and the employed performance metric. As the listed studies use different performance metrics and report results on different datasets, no quantitative direct comparison can be easily made. However, our work and that of Zhao et al. [20] had the most comprehensive evaluation with 20 datasets. Our proposed method clearly shows competitive performance and it compares well with the results obtained by the BORE [61], XGBOD [43], and the LSCP [20] methods.

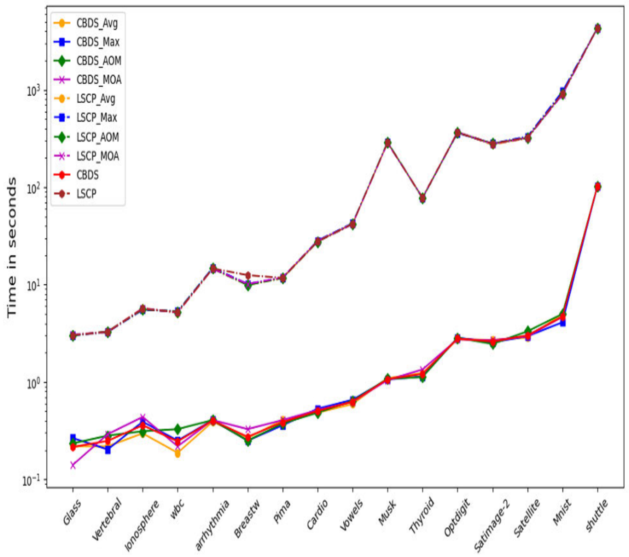

FIGURE 9. Execution times in seconds for the four variants of CBDS, LSCP and the average over 16 datasets. The datasets are labeled on the x-axis from the smallest to the largest dataset.

C. CLUSTERING ALGORITHM SELECTION

As explained above, the CBDS outlier detection method employs a clustering scheme to reduce the computational cost and enhance the outlier detection performance. In this paper, we used particularly the bisecting K-means algorithm as it generally leads to better performance compared to the conventional K-means clustering algorithm which shows clustering inconsistencies when outliers are present [62]. However, the bisecting K-means algorithm can lead to the creation of a larger number of small clusters. Alternately, more robust clustering algorithms could be considered, and this shall be addressed in future work.

D. FUSION METHOD SELECTION

In our work, we only explored (for simplicity) Averaging, Maximization, AOM, and MOA fusion schemes. No clear superiority can be claimed for one of these schemes over the other. Although the Max fusion method has unstable performance on small datasets, this method effectively captures outliers in high-dimensional spaces. Averaging-based fusion reduces the variance of the model prediction, reduces the detection error, and hence boosts the overall model performance. On the other hand, the Max fusion function enhances the detector accuracy by reducing the model bias.

Aggregating the Avg and Max operators in one fusion function (as in the case of MOA or AOM) combines the benefits of the two operators and expectedly improves the overall ensemble performance [51]. Recently, advanced combination methods for binary classification (detection) have been introduced [63]. Most of these methods agree that the choice of the fusion method depends on the ensemble structure and input data [64]. In particular, the alpha integration technique was proposed by Amari [65] to account for this dependency. Essentially, the integration characteristics are determined by a parameter α, while a weight vector w is used to assign a degree of importance to each measure. These parameters have been generally fixed. Later on, Soriano et al. [66] proposed a new alpha-integration method based on a minimum-probability-of-error criterion to learn optimal α and w parameter settings for detection (or binary classification) problems. Safon et al. [67] extended this work to late fusion of multiclass classification outcomes where optimal model parameters were obtained using a least-mean-square error formulation. We investigate in future work the effects of such advanced fusion methodologies on outlier detection performance.

E. BIAS-VARIANCE ANALYSIS

One of the fundamental aspects in the design of outlier ensembles is to control and reduce the detector prediction errors. The overall error consists essentially of bias and variance errors. On the one hand, a high variance results from the detector sensitivity to slight fluctuations in the training data, and this error is associated with overfitting. On the other hand, a high bias indicates the detector failure to learn key features, and this leads to poor data fitting and unreliable predictions.

The prediction errors in outlier ensembles can be mitigated by either reducing bias or reducing variance (but not both due to the tradeoff between the two types of error). In our CBDS method, we reduced the prediction error by combining diversified base detectors, where the diversity is induced by both initializing detectors with different parameter settings and training detectors on different data subsets generated by the employed clustering algorithm. To promote variance reduction, the scores of the base detectors can be combined by averaging as in the case of the CBDS_Avg variant.

Moreover, Rayana et al. [19] suggested that the removal of inaccurate detectors improves the overall accuracy, and therefore leads to bias error reduction. Following this suggestion, the CBDS method identifies and selects low-bias detectors based on local regions of competence. For example, this selection mechanism is realized by the CBDS_MOA and CBDS_AOM variants in the late combination phase, where high-bias detectors are excluded.

Nevertheless, the effect of fusion by maximization in bias-variance reduction is unpredictable, and this fusion scheme may improve bias but increase variance [13]. The CBDS algorithm still achieves variance reduction by averaging over the maximization function as in the case of the CBDS_AOM variant

F. LIMITATIONS

The proposed technique has indeed some limitations. First of all, we explored simple fusion approaches for ground-truth simulation (averaging or maximization) and fusion of final outputs (AOM or MOA). As shown by the statistical significance results of the Nemenyi test, these fusion functions mostly lead to statistically similar results. More powerful fusion techniques should be considered such as actively pruning base detectors [18] (for generating ground truth labels), and alpha integrating with learned optimized parameters [66] (for late classification fusion). Secondly, data locality induced by the bisecting K-means method merely groups data samples in subsets. Outlier detection performance can be possibly boosted if the input data is projected into feature subspaces. This might be carried out through advanced clustering techniques such as subspace K-means clustering [68] or spectral clustering [69].

VII. CONCLUSION AND FUTURE WORK

A. GENERAL CONCLUSION

In this study, we propose a clustering-based dynamic selection (CBDS) ensemble outlier detection method. In this method, an ensemble is composed of a collection of independent outlier detectors. The proposed CDBS improves the outlier detection performance through dynamically selecting the best-performing detectors in order to produce the final outlier scores. Another key advantage of the proposed method is reducing the time complexity of defining the local regions of competence into linear time ($O\ (km)$) compared to the log-linear time complexity ($O(n\ log\ n)$) of the LSCP. In fact, the selection of the competent detectors is determined during the training stage which results in a large speedup for the CBDS algorithm. The CBDS approach can be extended to other datasets and also the algorithmic parameters can be experimentally fine-tuned to enhance the outlier detection performance.

B. PRACTICAL IMPLICATIONS

Our work has the following practical implications. First of all, our approach showed remarkable performance improvements on datasets collected in different application scenarios. This shall increase the feasibility and applicability of outlier detection systems in a large number of real-world settings. Secondly, the substantial reduction in the computational cost improves the chances of reaching real-time detection performance in time-critical applications. Thirdly, the graceful degradation in the performance of our method under increasing levels of class imbalance alleviates the need for collecting well-balanced datasets for outlier detector training.

C. FUTURE DIRECTIONS

Several future research directions are worthy of investigation:

1) ALLEVIATING CLASS-IMBALANCE EFFECTS

In future work, we shall consider strategies for alleviating the effects of class imbalance [70] on the performance of our CBDS method.

2) REFINED LOCAL REGIONS OF COMPETENCE

Also, more robust clustering algorithms [69] shall be considered to overcome the limitations of both the traditional and bisecting K-means algorithms, and obtain refined local regions of competence.

3) BASE DETECTOR SELECTION

In this work, we created an ensemble of LOF-type detectors with different parameter settings. Alternately, isolation forests (IF) [71] can be used in combination with the LOF detectors.

REFERENCES

[1] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognit.*, vol. 74, pp. 406–421, Feb. 2018.

[2] W. P. Krijnen, *Applied Statistics for Bioinformatics Using R*. Groningen, The Netherlands: GNU Document, 2009.

[3] E. M. Knorr and R. T. Ng, "A unified notion of outliers: Properties and computation," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1997, pp. 219–222.

[4] S. Mehnaz and E. Bertino, "Ghostbuster: A fine-grained approach for anomaly detection in file system accesses," in *Proc. 7th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2017, pp. 3–14.

[5] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, Jun. 2014.

[6] R. Bolboacă, "Adaptive ensemble methods for tampering detection in automotive aftertreatment systems," *IEEE Access*, vol. 10, pp. 105497–105517, 2022.

[7] L. Stojanovic, M. Dinic, N. Stojanovic, and A. Stojadinovic, "Big-data-driven anomaly detection in industry (4.0): An approach and a case study," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1647–1652.

[8] N. L. Fitriyani, M. Syafrudin, G. Alfian, and J. Rhee, "Development of disease prediction model based on ensemble learning approach for diabetes and hypertension," *IEEE Access*, vol. 7, pp. 144777–144789, 2019.

[9] M. Prastawa, E. Bullitt, S. Ho, and G. Gerig, "A brain tumor segmentation framework based on outlier detection," *Med. Image Anal.*, vol. 8, no. 3, pp. 275–283, 2004.

[10] H. D. Cheng, X. J. Shi, R. Min, L. M. Hu, X. P. Cai, and H. N. Du, "Approaches for automated detection and classification of masses in mammograms," *Pattern Recognit.*, vol. 39, no. 4, pp. 646–668, Apr. 2006.

[11] B. Wu, "Cancer outlier differential gene expression detection," *Biostatistics*, vol. 8, no. 3, pp. 566–575, Jul. 2007.

[12] D. M. Hawkins, *Identification of Outliers*, vol. 11. London, U.K.: Chapman & Hall, 1980.

[13] C. C. Aggarwal and S. Sathe, *Outlier Ensembles: An Introduction*. Cham, Switzerland: Springer, 2017.

[14] C. C. Aggarwal, *An Introduction to Outlier Analysis*, 2nd ed. Cham, Switzerland: Springer, 2017.

[15] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2012, pp. 1047–1058.

[16] A. S. Britto, R. Sabourin, and L. E. S. Oliveira, "Dynamic selection of classifiers—A comprehensive review," *Pattern Recognit.*, vol. 47, no. 11, pp. 3665–3680, Nov. 2014.

[17] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Inf. Fusion* vol. 41, pp. 195–216, Sep. 2018.

[18] S. Rayana and L. Akoglu, "Less is more: Building selective anomaly ensembles," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 4, pp. 1–33, Jul. 2016.

[19] S. Rayana, W. Zhong, and L. Akoglu, "Sequential ensemble learning for outlier detection: A bias-variance perspective," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1167–1172.

[20] Y. Zhao, Z. Nasrullah, M. K. Hryniewicki, and Z. Li, "LSCP: Locally selective combination in parallel outlier ensembles," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, Jan. 2019, pp. 585–593.

[21] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large datasets," in *Proc. ACM Int. Conf. Manage. Data (SIGMOD)*, 2000, pp. 427–438.

[22] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proc. Int. Conf. Very Large Data Bases (VLDB)*, New York, NY, USA, 1998, pp. 392–403.

[23] N. N. R. Ranga Suri, N. M. Murty, and G. Athithan, "Outlier detection: Techniques and applications," in *Intellgent Systems Reference Library*, vol. 155. Cham, Switzerland: Springer, 2018, pp. 28–30.

[24] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *Proc. 6th Eur. Conf. Princ. Data Mining Knowl. Discovery (PKDD)*, Sep. 2002, pp. 15–27.

[25] L. Tran, M. Y. Mun, and C. Shahabi, "Real-time distance-based, outlier detection in data streams," in *Proc. VLDB Endowment (PVLDB)*, vol. 14, no. 2, pp. 141–153, Oct. 2021.

[26] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A review of local outlier factor algorithms for outlier detection in big data streams," *Big Data Cognit. Comput.*, vol. 5, no. 1, p. 1, Dec. 2020.

[27] D. Tripathi, Y. Sharma, and T. Lone, "Credit card fraud detection using local outlier factor," *Int. J. Pure Appl. Math.*, vol. 118, no. 7, pp. 229–234, 2018.

[28] M. M. Breunig, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 14–93.

[29] Z. Xu, D. Kakde, and A. Chaudhuri, "Automatic hyperparameter tuning method for local outlier factor, with applications to anomaly detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 4201–4207.

[30] H. Moradi, S. Ibrahim, and J. Hosseinkhani, "Outlier detection in stream data by clustering method," *Int. J. Adv. Comput. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 25–34, 2013.

[31] L. Torgo, C. Soares, and A. Loureiro, "Outlier detection using clustering methods: A data cleaning application," in *Proc. KDNet Symp. Knowl.-Based Syst. Public Sector*, 2004, pp. 1–12.

[32] B. M. Al-Zoubi, A. AL-Dahoud, and A. A. Yahya, "New outlier detection method based on fuzzy clustering," *Inf. Sci. Appl.*, vol. 7, pp. 681–690 May 2010.

[33] V. B. G. Pranali and K. Bhowate, "Outlier detection method for data set based on clustering and EDA technique," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 3, no. 2, p. 2278, 2014.

[34] V. Barnett and T. Lewis, *Outliers in Statistical Data*. Hoboken, NJ, USA: Wiley, 1994.

[35] A. Barghash and T. Arslan, "Robust detection of outlier samples and genes in expression datasets," *J. Proteomics Bioinf.*, vol. 9, no. 2, pp. 38–48, 2016.

[36] B. G. Peter and J. P. Messina, "Errors in time-series remote sensing and an open access application for detecting and visualizing spatial data outliers using Google Earth engine," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1165–1174, Apr. 2019.

[37] H. Cho, Y.-J. Kim, H. J. Jung, S.-W. Lee, and J. W. Lee, "OutlierD: An r package for outlier detection using quantile regression on mass spectrometry data," *Bioinformatics*, vol. 24, no. 6, pp. 882–884, Jan. 2008.

[38] E. Calikus, S. Nowaczyk, A. Sant'Anna, and O. Dikmen, "No free lunch but a cheaper supper: A general framework for streaming anomaly detection," *Expert Syst. Appl.*, vol. 155, Oct. 2020, Art. no. 113453.

[39] D. Barbará, Y. Li, J. Couto, J. L. Lin, and S. Jajodia, "Bootstrapping a data mining intrusion detection system," in *Proc. ACM Symp. Appl. Comput.*, Mar. 2003, pp. 421–425.

[40] M. A. Bhatti, R. Riaz, S. S. Rizvi, S. Shokat, F. Riaz, and S. J. Kwon, "Outlier detection in indoor localization and Internet of Things (IoT) using machine learning," *J. Commun. Netw.*, vol. 22, no. 3, pp. 236–243, Jun. 2020.

[41] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2005, pp. 157–166.

[42] Y. Zhao and M. K. Hryniewicki, "XGBOD: Improving supervised outlier detection with unsupervised representation learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[43] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 190–237, Dec. 2012.

[44] X. Zhu, X. Wu, and Y. Yang, "Dynamic classifier selection for effective mining from noisy data streams," in *Proc. 4th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2004, pp. 305–312.

[45] J. K. Bii, R. Rimiru, and R. W. Mwangi, "Adaptive boosting in ensembles for outlier detection: Base learner selection and fusion via local domain competence," *ETRI J.*, vol. 42, no. 6, pp. 886–898, Dec. 2020.

[46] B. van Stein, M. van Leeuwen, and T. Bäck, "Local subspace-based outlier detection using global neighbourhoods," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1136–1142.

[47] Y. Zhao and M. K. Hryniewicki, "DCSO: Dynamic combination of detector scores for outlier ensembles," in *Proc. ACM SIGKDD Workshop Outlier Detection De-Constructed ODD*, vol. 5, 2018, pp. 1–9.

[48] S. M. Savaresi and D. L. Boley, "On the performance of bisecting K-means and PDDP," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2001, pp. 1–14.

[49] R. Nainggolan, R. Perangin-Angin, E. Simarmata, and A. F. Tarigan, "Improved the performance of the K-means cluster using the sum of squared error (SSE) optimized by using the elbow method," *J. Phys., Conf. Ser.*, vol. 1361, no. 1, Nov. 2019, Art. no. 012015.

[50] E. Umargono, J. E. Suseno, and S. V. Gunawan, "K-means clustering optimization using the elbow method and early centroid determination based on mean and median formula," in *Proc. 2nd Int. Seminar Sci. Technol. (ISSTEC)*. Atlantis Press, 2020, pp. 121–129.

[51] C. C. Aggarwal and S. Sathe, "Theoretical foundations and algorithms for outlier ensembles," *ACM SIGKDD Explorations Newslett.*, vol. 17, no. 1, pp. 24–47, Sep. 2015.

[52] *Outlier Detection DataSets (ODDS)*. Accessed: Dec. 15, 2022. [Online]. Available: http://odds.cs.stonybrook.edu

[53] *Deepak, System hack: Highly Imbalanced Data*. Accessed: Dec. 15, 2022. [Online]. Available: https://www.kaggle.com/datasets/deepakat002/system-hack-highly-imbalanced-data

[54] A. Ali. *Credit Card Cheating Detection (CCCD)*. Accessed: Dec. 15, 2022. [Online]. Available: https://www.kaggle.com/datasets/arslanali4343/credit-card-cheating-detection-cccd

[55] C. Manchanda. *Fraudulent Transactions Data*. Accessed: Dec. 15, 2022. [Online]. Available: https://www.kaggle.com/datasets/chitwanmanchanda/fraudulent-transactions-data

[56] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, Jan. 2019.

[57] F. Pedregosa, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.

[58] A. Salazar, G. Safont, and L. Vergara, "Semi-supervised learning for imbalanced classification of credit card transaction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.

[59] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[60] C.-F. Tsai and W.-C. Lin, "Feature selection and ensemble learning techniques in one-class classifiers: An empirical study of two-class imbalanced datasets," *IEEE Access*, vol. 9, pp. 13717–13726, 2021.

[61] B. Micenková, B. McWilliams, and I. Assent, "Learning representations for outlier detection on a budget," 2015, *arXiv:1507.08104*.

[62] M. Steinbach and V. P. K. Tan, "Cluster analysis: Basic concepts and algorithms," in *Introduction to Data Mining*, 1st ed. London, U.K.: Pearson, 2005.

[63] M. Mohandes, M. Deriche, and S. O. Aliyu, "Classifiers combination techniques: A comprehensive review," *IEEE Access*, vol. 6, pp. 19626–19639, 2018.

[64] Y. Zhao, R. A. Rossi, and L. Akoglu, "Automating outlier detection via meta-learning," 2020, *arXiv:2009.10606*.

[65] S.-I. Amari, "Integration of stochastic models by minimizing α-divergence," *Neural Comput.*, vol. 19, no. 10, pp. 2780–2796, Oct. 2007.

[66] A. Soriano, L. Vergara, B. Ahmed, and A. Salazar, "Fusion of scores in a detection context based on alpha integration," *Neural Comput.*, vol. 27, no. 9, pp. 1983–2010, 2015.

[67] G. Safont, A. Salazar, and L. Vergara, "Vector score alpha integration for classifier late fusion," *Pattern Recognit. Lett.*, vol. 136, pp. 48–55, Aug. 2020.

[68] L. Morissette and S. Chartier, "The K-means clustering technique: General considerations and implementation in mathematica," *Tuts. Quant. Methods Psychol.*, vol. 9, no. 1, pp. 15–24, Feb. 2013.

[69] M. Jordan, Y. Weiss, and A. Ng, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 1–8.

[70] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Inf. Sci.*, vol. 465, pp. 1–20, Jun. 2018.

[71] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.

[72] B. Ouyang, Y. Song, Y. Li, G. Sant, and M. Bauchy, "EBOD: An ensemble-based outlier detection algorithm for noisy datasets," *Knowl.-Based Syst.*, vol. 231, Nov. 2021, Art. no. 107400.

RASHA RAMADAN Z. KOKO was born in Khartoum, Sudan. She received the B.Sc. degree in electronic engineering from the Sudan University of Science and Technology (SUST), in 2003, and the M.Sc. degree in biomedical engineering from Cairo University, in 2013, where she is currently pursuing the Ph.D. degree with the Department of Biomedical Engineering and Systems. From 2003 to 2008, she worked as a Biomedical Engineer with the Molecular Biology Laboratory and the National Public Health Laboratory (NPHL), Sudan. Since 2009, she has been working as a Bioinformatician and a Researcher with NPHL. Her research interests include clinical informatics, bioinformatics, machine learning, anomaly detection, and software development.

INAS A. YASSINE received the B.Sc. and M.Sc. degrees from the Biomedical Engineering Department (SBME), Cairo University, Cairo, Egypt, in 2003 and 2006, respectively, and the Ph.D. degree from West Virginia University, USA. She is currently a Professor in biomedical engineering with Cairo University. She joined the National Institute of Diabetes, Digestive and Kidney Diseases, National Institutes of Healthcare, MD, USA, as a Research Visitor, in 2019. Moreover, she also worked as an Adjunct Assistant Professor with Nile University. She has published more than 40 peer-reviewed manuscripts. Her research interests include medical image analysis and visualization, feature engineering, machine learning, deep learning, and big data.

She has received several research and academic awards, such as the Nabil Bassiouny Award, the International Scientific Publication Award, and the Best M.Sc. degree Thesis Award, in 2006, 2011, and 2008, respectively. In 2009, she received the Best Paper Award from the International Symposium on Visual Computing, USA. In 2010, she also received the WVU Women Recognition Award, WV, USA.

MANAL ABDEL WAHED was born in Giza, Egypt, in 1965. Since October 1987, she has been with the Department of Systems and Biomedical Engineering, Faculty of Engineering, Cairo University, where she worked as a Teaching Assistant and an Assistant Professor, in 1999, an Associate Professor, in 2008, and a Professor, in 2016. She has coauthored over 50 publications and supervised more than 30 theses for M.Sc. and Ph.D. degrees. Her current research interests include medical imaging, pattern recognition/classification, neural networks, biomedical applications, bioinformatics, data mining, clinical engineering, and healthcare quality systems.

JUNE K. MADETE received the Ph.D. degree in medical engineering, specializing in biomechanics, motion capture, imaging studies, and patient data collection. Her specialty is biomechanics, looking at the body as a machine. She seeks to develop biomedical engineering in Africa through knowledge and skill transfer. She is currently the Chairperson of the Department that houses biomedical engineering with the School of Engineering and Technology, Kenyatta University, where supervises engineering projects and gives lectures on biomechanics, biomaterials, biomedical device design, and quality management. Her passion lies in innovation and capacity building, and has collaborated with various universities to carry out curriculum development, organize various design workshops and schools, and bring together BME universities for networking though the BMEIdea meetings.

MUHAMMAD A. RUSHDI received the B.Sc. degree in biomedical engineering and systems, in 2001, the B.Sc. degree in mathematics from Cairo University, Giza, Egypt, in 2003, the M.Sc. degree in biomedical engineering and systems, in 2005, and the M.Sc. and Ph.D. degrees in computer and information science and engineering from the University of Florida, Gainesville, FL, USA, in 2012 and 2013, respectively. He is currently an Associate Professor with the School of Information Technology, New Giza University, and the Department of Biomedical Engineering and Systems, Cairo University. He has over 40 peer-reviewed publications in reputable journals and conferences. He co-advised more than 25 M.Sc. and Ph.D. students. He received research and development support from EACEA, ITIDA, Flat6Labs, and French Tech Ticket. His research interests include biomedical signal processing, medical imaging, information security and forensics, machine learning, image processing, computer vision, and applied mathematics.