electronics

MDPI

Article

A Novel Unsupervised Outlier Detection Algorithm Based on Mutual Information and Reduced Spectral Clustering

Yuehua Huang [1,2,*], Wenfen Liu [1,2], Song Li [1], Ying Guo [1] and Wen Chen [1]

[1] School of Computer Science and Information Security & School of Software Engineering, Guilin University of Electronic Technology, Guilin 541004, China
[2] Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China
* Correspondence: huangyehua@163.com

Abstract: Outlier detection is an essential research field in data mining, especially in the areas of network security, credit card fraud detection, industrial flaw detection, etc. The existing outlier detection algorithms, which can be divided into supervised methods and unsupervised methods, suffer from the following problems: curse of dimensionality, lack of labeled data, and hyperparameter tuning. To address these issues, we present a novel unsupervised outlier detection algorithm based on mutual information and reduced spectral clustering, called MISC-OD (Mutual Information and reduced Spectral Clustering—Outlier Detection). MISC-OD first constructs a mutual information matrix between features, then, by applying reduced spectral clustering, divides the feature set into subsets, utilizing the LOF (Local Outlier Factor) for outlier detection within each subset and combining the outlier scores found within each subset. Finally, it outputs the outlier score. Our contributions are as follows: (1) we propose a novel outlier detection method called MISC-OD with high interpretability and scalability; (2) numerous experiments on 18 benchmark datasets demonstrate the superior performance of the MISC-OD algorithm compared with eight state-of-the-art baselines in terms of ROC (receiver operating characteristic) and AP (average precision).

Keywords: outlier detection; unsupervised; mutual information; spectral clustering

Citation: Huang, Y.; Liu, W.; Li, S.; Guo, Y.; Chen, W. A Novel Unsupervised Outlier Detection Algorithm Based on Mutual Information and Reduced Spectral Clustering. *Electronics* **2023**, *12*, 4864. https://doi.org/10.3390/electronics12234864

Academic Editor: Alberto Fernandez Hilario

Received: 4 November 2023
Revised: 24 November 2023
Accepted: 29 November 2023
Published: 2 December 2023

1. Introduction

Outlier detection, sometimes referred to as anomaly detection or novelty detection, is the process of picking out the outliers from normal values. According to Hawking, "an outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" [1].

Outlier detection is an essential research field in data mining due to its widespread use in a wide range of applications, such as network intrusion detection [2–7], intelligent transportation [8–10], video content analysis and detection [11–13], fraud detection [14–18], and social media analysis [19–21].

Outlier detection is an important field of research and is a concern for industry and academia. By identifying outliers, researchers can obtain vital knowledge, which assists in making better decisions or avoiding risks.

As mentioned before, these applications require outlier detection to achieve the following goals: (1) a short running time, especially for some online detection tasks, (2) high detection accuracy, and (3) being easy to interpret, especially in fraud detection and network security applications.

Over the past few decades, many outlier detection algorithms have been proposed [22–25]. These can be divided into five main categories: (1) statistical and probabilistic-based methods; (2) proximity-based methods; (3) clustering-based methods; (4) ensemble-based methods; and (5) learning-based methods. We will give a more detailed introduction in Section 2.

Although the above algorithms have achieved good performance in the past, with the arrival of the era of big data, these algorithms have shown the following shortcomings:

(1) Curse of dimensionality. In the era of big data, the dimensions of data are steadily increasing. The performance of traditional outlier detection algorithms, especially those based on proximity, will decrease rapidly with the increase in dimensions.
(2) Missing labeled data. Cluster-based algorithms require a large amount of labeled data, which is difficult to implement or comes at a high cost in many scenarios.
(3) Hyperparameter tuning. Algorithms based on learning have a large number of hyperparameters, which need hyperparameter tuning to achieve good performance. Hyperparameter tuning requires a large amount of time and computing power, which is difficult to implement in some online outlier detection applications, especially in unsupervised settings.

To address these limitations, this manuscript proposes a new outlier detection algorithm based on mutual information and reduced spectral clustering (Mutual Information and reduced Spectral Clustering—Outlier Detection, abbreviated as MISC-OD). Firstly, the mutual information between features is calculated and a mutual information matrix is constructed. Then, reduced spectral clustering is used to cluster the features. On this basis, the LOF outlier detection algorithm is used to detect outliers in each feature subset, and the result of outlier detection is finally obtained.

Compared with traditional outlier detection algorithms, the MISC-OD algorithm has better performance in the face of the curse of dimensionality, scalability, and higher interpretability. In addition, since the algorithm is unsupervised, there is no need to carry out a significant amount of hyperparameter tuning, which is critical in online detection scenarios.

The contributions are as follows:

(1) A new anomaly detection algorithm, MISC-OD, is proposed. Especially when faced with highly dimensional data, its anomaly detection effect is better than other classic algorithms. The MISC-OD algorithm has high interpretability and scalability. These characteristics are critical in practical applications.
(2) A large number of experiments have been carried out, and the experimental results prove that the algorithm has good performance compared with state-of-the-art outlier detection methods.

The rest of this manuscript is organized as follows: we introduce some existing outlier detection techniques in Section 2, which can be divided into statistical and probabilistic-based methods, proximity-based methods, clustering-based methods, ensemble-based methods, and learning-based methods. We introduce some preliminary knowledge in Section 3, including mutual information, spectral clustering, and the definition of anomaly detection problems. We describe the algorithmic details of MISC-OD and its properties in Section 4. We carry out experiments to compare MISC-OD with state-of-the-art outlier detection methods and demonstrate its excellent performance in Section 5. Finally, we conclude this manuscript in Section 6 with future research directions.

2. Related Work

As mentioned above, outlier detection methods can be classified into five categories. In this section, the main ideas and pros and cons of each category are introduced in detail.

2.1. Statistical and Probabilistic-Based Methods

Statistical and probabilistic-based methods firstly make an assumption about the distribution of data, and then calculate the extreme degree of a certain data point based on this assumption. The extreme degree of a certain data point is called an "outlier score".

Depending on whether it has parameters or not, these methods can be divided into two classes: parametric methods and nonparametric methods. Representative parametric methods are linear regression [26,27] and Gaussian mixture models (GMM) [26,27]. Nonparametric methods do not assume a parametric model for the data. Some typical examples

include histogram-based methods (HBOS) [28,29], Kernel Density Estimation (KDE) [30], and other variants.

The premise that methods can achieve better results is that the assumed data distribution is consistent with actual situations. This means that sometimes this assumption does not hold, leading to poor detection results using these methods.

2.2. Proximity-Based Methods

Proximity-based methods are based on local neighborhood information and define a data point as an outlier when its locality (or proximity) is sparsely populated [31]. These algorithms can be categorized into density-based and distance-based algorithms [31].

The core principle of the density-based outlier detection methods is that an outlier can be found in a low-density region, whereas non-outliers (inliers) are assumed to appear in dense neighborhoods [27]. A typical representative of this method is LOF. Some improvements were later introduced, such as ELOF [32] (extract local outlier factor) and COF [33] (class outlier factor).

Distance-based outlier detection methods detect outliers by computing the distances between data points. A data point that is at a far distance from its nearest neighbor is regarded as an outlier [27].

One classic distance-based outlier detection method is KNN (k-nearest neighbor), which is often used as a benchmark for comparison with other outlier detection algorithms.

The main advantages of proximity-based methods are nonparametric and easy to interpret. However, these methods are computationally expensive for computing each pair of data points, sensitive to hyperparameters such as how to define neighbor or distance.

2.3. Clustering-Based Methods

The core of clustering-based methods is clustering. Through clustering, smaller-sized clusters in the dataset can be found, and the smaller-sized clusters that comprise significantly fewer data points than other clusters are labeled as outliers.

A detailed description of these methods can be found in [34]. The merit of clustering-based methods are that they are (1) unsupervised, which means they are easy to use, and (2) robust to different data types. However, the shortcomings are obvious, including (1) the need to specify the number of clusters in advance, which is a difficult task on certain occasions, and (2) the result of clustering-based methods is binary, which means no quantitative indication of the outlierness.

2.4. Ensemble-Based Methods

Ensemble-based methods combine the results obtained from various base outlier detectors to produce more robust and more accurate results.

The key to the success of this approach is how base detectors are selected and how the different results are combined. Some remarkable research on this category includes isolation forests, feature bagging, and LSCPs [22] (Locally Selective Combination in Parallel Outlier Ensembles).

Ensemble-based methods can be very useful in areas where the data are noisy and in streaming scenarios because they are more stable and robust. In addition to this, they are suitable for outlier analysis in highly dimensional data.

However, selecting the right base outlier detectors is a difficult task. Furthermore, it results in a yes or no answer, which prevents further comparisons from being made.

2.5. Learning-Based Methods

With the emergence of big data and significant improvements in computing power, an increasing number of machine learning methods are applied to outlier detection.

A machine learning model is trained to distinguish between normal values (inliers) and abnormal values (outliers). When the model is trained, it will be used to find outliers

from unlabeled data. The OCSVM (One-Class Support Vector Machine) is a classical method in this category.

In recent years, some researchers have tried to use neural networks for outlier detection. Some representative examples include GANs (generative adversarial networks) [24], variational autoencoders [35], and reinforcement learning [36].

Better results can be achieved from learning-based methods when large datasets exist. However, in most scenarios of outlier detection, this assumption is not valid. In addition to this, most learning-based methods are computationally expensive and need nontrivial hyperparameter tuning to yield the best performance, especially in the unsupervised setting. Lastly, learning-based methods are black box models, which means that it is hard to interpret the results obtained using learning-based methods.

3. Preliminaries

3.1. Problem of Outlier Detection

Unsupervised outlier detection, without supervision, employs some criteria to identify outlier candidates that deviate from major normal points [27,34]. Based on different assumptions of outliers, many algorithms are proposed that assign an outlier score to each point and return top-ks as outlier candidates.

We have n data points $X_1,\ X_2,\ \ldots,\ X_n \in R^d$, which are sampled independently and identically distributed. We use the matrix $\boldsymbol{X} \in R^{n \times d}$ as the notation of the entire dataset, which is formed by stacking each data point's vectors as rows. Given $\boldsymbol{X}$, an outlier detector obtains an outlier score $o_i \in R$ for each data point x_i, $1 \leq i \leq n$. Having a higher outlier score means it is more likely to be an outlier.

3.2. Mutual Information

In information theory, the mutual information (MI) of two random variables is a measure of the mutual dependence between two variables [12]. In other words, the mutual information between two random variables measures non-linear relations between them. More specifically, it quantifies the "amount of information" (in units such as bits, nats, or hartleys) obtained about one random variable by observing the other random variables.

The mutual information and entropy of a random variable are closely related concepts in information theory. Entropy measures the anticipated "amount of information" carried in a random variable.

For two discrete variables X and Y whose joint probability distribution is $P_{XY}(x, y)$, the mutual information between them, denoted as $I(X;Y)$, can be obtained using Equation (1).

$$I(X;Y) = \sum_{x;y} P_{XY}(x,y) log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} = E_{P_{XY}} log \frac{P_{XY}}{P_X P_Y} \tag{1}$$

Here, $P_{XY}(x, y)$ is a joint probability distribution, and $P_X(x)$ and $P_Y(y)$ are the marginal distribution, which can be obtained through Equations (2) and (3).

$E_{P_{XY}}$, P_{XY}, P_X, P_Y means the expectation, joint probability value, marginal distribution value on X, and marginal distribution value on Y when given a specific x and y.

$$P_X(x) = \sum_y P_{XY}(x,y) \tag{2}$$

$$P_Y(y) = \sum_x P_{XY}(x,y) \tag{3}$$

Whereas to compute the mutual information for continuous random variables, the summations must be replaced by the integrals.

$$I(X;Y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p_{(x,y)}(x,y) \cdot log \left(\frac{p_{(X,Y)}(x,y)}{p_X(x) p_Y(y)} \right) dxdy \tag{4}$$

3.3. Spectral Clustering

Spectral clustering is a technique with roots in graph theory, where the approach is used to identify communities of nodes in a graph based on the edges connecting them [27,34]. The method is flexible and clusters non-graph data as well.

Spectral clustering uses information from the eigenvalues (spectrum) of special matrices built from the graph or the dataset. Its main process includes constructing matrices and using the eigenvectors of matrices to assign data to different clusters.

By clustering the features, the highly correlated features are grouped into the same group, which reduces the complexity of the subsequent training, and also produces more accurate results and better interpretability.

In fact, different clustering algorithms can be employed to cluster the quantities describing correlations, such as k-means clustering, or DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering, for Pearson correlation coefficients or mutual information. In many cases, the correlation between features is not necessarily linear. Therefore, this manuscript uses reduced spectral clustering to cluster the mutual information to realize the subsets of features.

4. Proposed MISC-OD Method

Here, we provide the details of the new outlier detection algorithm based on mutual information and reduced spectral clustering (Mutual Information and reduced Spectral Clustering—Outlier Detection, abbreviated as MISC-OD).

Firstly, we provide the motivation and technical details of MISC-OD in Section 4.1, and then discuss the properties of MISC-OD in Section 4.2, including interpretability and scalability.

4.1. Detailed Steps of MISC-OD

Table 1 shows the major notations used in the following sections.

Table 1. Algorithm notations.

Notation	Domain	Description
n	$\mathbb{Z}$	Number of samples
d	$\mathbb{Z}$	Number of features
K	$\mathbb{Z}$	Number of clusters
$\boldsymbol{X}$	$\mathcal{R}^{n\times d}$	Dataset
X_i	$\mathcal{R}^{1\times d}$	One instance, $1 \leq i \leq n$
C_j	$\mathcal{R}^{n\times 1}$	One features, $1 \leq j \leq d$
$\boldsymbol{W}$	$\mathcal{R}^{d\times d}$	Mutual information matrix between features
$\boldsymbol{D}$	$\mathcal{R}^{d\times d}$	Degree matrix
$\boldsymbol{V}$	$(C_1, C_2, \cdots C_d)$	Vertex set
G	$(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$	Weighted undirected graph
$\boldsymbol{L}$	$\mathcal{R}^{d\times d}$	Laplacian matrix of G

4.1.1. Construction Mutual Information Matrix

As mentioned in Section 3.1, now we have n data points $X_1, X_2, \ldots, X_n \in R^d$, which are sampled independently and are identically distributed. We use the matrix $\boldsymbol{X} \in R^{n\times d}$ as the notation of the entire dataset, which is formed by stacking each data point's vectors as rows.

First, we perform data preprocessing on each data point, discretizing continuous features using the equal width method. Then, the mutual information between each feature can be calculated using Equation (5) and (6) to obtain the mutual information matrix.

$$w(C_i, C_j) = \sum_{x \in C_i} \sum_{y \in C_j} p(x,y) log\left(\frac{p(x,y)}{p(x) \cdot p(y)}\right) \tag{5}$$

$$p(x,y) = \frac{n(x,y)}{d}, p(x) = \frac{n(x)}{d}, p(y) = \frac{n(y)}{d} \tag{6}$$

In Equation (5), $p(x,y)$ means the probability that a particular x and y appear in the joint probability distribution, $p(x)$ means the probability that the same x appears in the marginal distribution P_X, and $p(y)$ means the probability that the same y appears in the marginal distribution P_Y.

Equation (5) shows how to calculate $p(x,y)$, $p(x)$, $p(y)$, and Equation (6) gives the detailed calculation processes. In Equation (6), $n(x,y)$ represents the number of occurrences of this value (x,y) in (C_i, C_j). Similarly, $n(x)$ represents the number of occurrences of this x value in C_i, while $n(y)$ represents the number of occurrences of this y value in C_j.

Through the above equations, we can obtain matrix $\boldsymbol{W}$, composed of mutual information between features.

4.1.2. Reduced Spectral Clustering

Let $G = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ be a weighted undirected graph with vertex set $\boldsymbol{V} = (C_1, C_2, \cdots C_d)$. In the following section, we suppose that the graph G is weighted; that is, each edge between two vertices C_i and C_j carries a non-negative weight $w_{ij} \geq 0$. The weighted adjacency matrix of the graph is the matrix $\boldsymbol{W} = \left[w_{i,j}\right]_{d \times d}$, which can be obtained using Equation (5) in previous section.

(1) As G is undirected, this means $w_{ij} = w_{ji}$. The degree of a vertex $C_i \in V$ is defined as $d_i = \sum_{j=1}^{d} w_{ij}$.
(2) The degree matrix D is defined as the diagonal matrix with the degrees $d_i, 1 \leq i \leq d$ on the diagonal. The unnormalized graph Laplacian matrix can be obtained by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$.
(3) Then, we can obtain the normalized graph Laplacians by

$$\boldsymbol{L}' = \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2}$$

(4) We computed the first k eigenvectors (the eigenvectors corresponding to the k smallest eigenvalues of L').
(5) We obtained a new matrix formed by the first k eigenvectors; the l-th row defines the features of graph node C_l.
(6) We clustered the graph nodes based on these features (e.g., using k-means clustering).
(7) We finally obtained subsets of features, $Sub = Sub_1 \cup Sub_2 \cup \cdots \cup Sub_K$, which realized the use of mutual information to divide features into several closely related sub-categories, which will help improve the efficiency and accuracy of detection in the subsequent outlier detection stage.

4.1.3. Outlier Detection Using LOF

(1) For each sub-category of feature set, by using the classical LOF algorithm, we can obtain the outlier score of each instance. We use the notation o_{ij}, which means sample X_i's outlier score in Sub_j, $1 \leq j \leq K$.
(2) Then, we aggregate the outlier scores obtained from each sub-category $\{o_1, o_2, \ldots, o_i, \ldots, o_n\}$, $o_i = \sum_{j=1}^{K} o_{ij}$.
(3) The output is $\{o_1, o_2, \ldots, o_i, \ldots, o_n\}$.

4.1.4. Pseudocode of Algorithm MISC-OD

Through the previous steps (Sections 4.1.1–4.1.3), the entire process of outlier detection is completed. To summarize the process, we have given the pseudocode of the MISC-OD algorithm in Algorithm 1.

Algorithm 1 MISC-OD

Input: input data $\boldsymbol{X} = \left[x_{ij}\right]_{n\times d}$ with n samples and d features
Output: Outlier scores $\{o_1, o_2, \ldots, o_i, \ldots, o_n\}$
1. Obtain mutual information matrix $\boldsymbol{W}$:
   $w\left(C_i, C_j\right) = \sum_{x\in C_i}\sum_{y\in C_j} p(x,y)log\left(\frac{p(x,y)}{p(x)\cdot p(y)}\right)$,
   $p(x,y) = \frac{n(x,y)}{d}, p(x) = \frac{n(x)}{d}, p(y) = \frac{n(y)}{d}$
2. Calculate degree matrix $\boldsymbol{D} = \left[d_{ij}\right]_{d\times d}$
   $d_{ij} = d_{ji} = \sum_{j=1}^{d} w_{ij}$.
3. Obtain Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$
4. Normalized graph Laplacians $\boldsymbol{L}' = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}}$
5. Obtain subsets of features: $Sub = Sub_1 \cup Sub_2 \cup \cdots \cup Sub_K$
6. **For** each subset of feature sub in $1, \cdots, i, \cdots K$, do
   apply LOF algorithm
   return o_{ij}
7. **Return** $\{o_1, o_2, \ldots, o_i, \ldots, o_n\}$,$o_i = \sum_{j=1}^{K} o_{ij}$

4.2. Properties of MISC-OD

(1) Interpretability. This is a key idea in machine learning because it gives domain specialists a better understanding of how algorithms decide what to do. When a model is transparent, people may try to understand why a certain data point is categorized and obtain insight from the way models think. Transparency and dependability are both offered by interpretable algorithms.

In credit card fraud detection scenarios, recognizing a fraudulent transaction and providing an explanation for it are equally crucial. Thus, in applications involving outlier detection, interpretability is equally vital.

Conventional anomaly detection algorithms identify patterns in every feature of the complete dataset. Many anomaly detection algorithms, particularly learning-based algorithms, become black boxes that can only output anomaly detection results when faced with high-dimensional data; they are unable to determine why the data are anomalous. In many real-world anomaly detection applications, this is unacceptable.

MISC-OD created a mutual information matrix from the mutual information between features before performing outlier detection, and then grouped features with the help of spectral clustering. The outlier detection was performed on subsets of features, which accurately explains why the data are recognized as an outlier, as the algorithm can give the specific features on which the outlier scores are higher, when necessary. Therefore, MISC-OD is an outlier detection algorithm with high interpretability.

(2) Scalability. Proximity-based methods (introduced in Section 2.2) require frequent calculations of the distance or density between data points. Unlike these algorithms, MISC-OD only needs to compute the mutual information between features once. Distributed computing or multithreading can be added to the outlier detection process of a subset of features to make the MISC-OD method much more scalable.

(3) Time complexity analysis. In the first step (Sections 4.1.1 and 4.1.2), the construction mutual information matrix and reduced spectral clustering lead to $\mathcal{O}\left(nd^2 + d^3\right)$ time complexity. In the second step (Section 4.1.3), outlier detection using LOF in each feature subset leads to $\mathcal{O}\left(n^2\right)$ time complexity. So, MISC-OD has $\mathcal{O}\left(n^2\right)$ time complexity when $n \gg d$.

We performed anomaly detection through MISC-OD on the synthetic dataset ($n = 1,000,000$,$d = 1000$), and its running time was consistent with the above time complexity.

5. Experimental Results and Analysis

5.1. Evaluation Metric

5.1.1. ROC-AUC (Receiver Operating Characteristic, Area under the Curve)

The receiver operating characteristic (ROC) curve is frequently used for evaluating the performance of binary classification algorithms. It provides a graphical representation of a classifier's performance, rather than a single value like most other metrics. The closer ROC is to 1, the more effective that detection model is. When ROC is equal to or lower than 0.5, this means the inspection model has no value for use.

5.1.2. AP (Average Precision)

Another way to evaluate outlier detection models is to use average precision (AP). AP measures the average precision across all possible thresholds, with a higher value indicating a better model. AP is more suitable for outlier detection problems with rare anomalies or imbalanced data, as it focuses more on the positive class (anomalies) than the negative class (normal instances). However, it may not reflect the overall accuracy or specificity of the model, as it does not account for the true negatives or false negatives. Evaluating outlier detection models can be challenging, especially when you do not have labeled data or ground truth data to compare with. One of the possible ways to evaluate outlier detection models is to use external validation, which means comparing the results with some other sources of information, such as domain experts, feedback, or historical data.

Thirty percent of the dataset in experiments is reserved for testing, while the remaining seventy percent is used for training. The area under the receiver operating characteristic (ROC) and average precision (AP) are used to obtain the average score from ten separate trials to assess performance.

5.2. Experimental Setup

5.2.1. Experimental Environment and Baselines

In the subsequent experiments, a Windows personal computer with an AMD Ryzen 7 5800H CPU and 16G of memory was used.

For the sake of fairness, we selected eight representative anomaly detection algorithms from different categories, including proximity-based methods, clustering-based methods, and ensemble-based methods. These eight algorithms are all classic algorithms in the field of anomaly detection and have gained recognition from academia and industry.

We compared the performance of MISC-OD with eight state-of-the-art outlier detection algorithms. These are k-nearest neighbor (KNN), Local Outlier Factor (LOF), Angle-Based Outlier Detection (ABOD) [37], Histogram-Based Outlier Score (HBOS) [28], Isolation Forest (IForest), Clustering-Based Local Outlier Factor (CBLOF), Locally Selective Combination in Parallel Outlier Ensembles (LSCP) [22], and One-Class Support Vector Machine (OCSVM).

5.2.2. Dataset

To ensure fairness, our experiments chose public datasets instead of synthetic datasets or private datasets. At the same time, in order to reflect the superiority of the MISC-OD algorithm, 18 datasets with data dimensions greater than or equal to 10 were selected from 31 public datasets of ODDS. The volume of datasets ranged from 129 to 286,048, the dimension of the datasets ranged from 10 to 400, and the percentage of outliers ranged from 0.9% to 32%. This shows that the selected datasets are highly representative from multiple perspectives such as the data volume, the distribution of the data dimensions, and the percentage of outliers, and can represent various datasets in real applications.

Table 2 presents the 18 multi-dimensional datasets with data dimensions greater than or equal to 10 from ODDS (https://odds.cs.stonybrook.edu/, accessed on 1 September 2023).

Table 2. Eighteen real-word benchmark datasets.

Dataset	Number of Samples	Number of Dimensions	Outliers (%)
Lympho	148	18	6 (4.1%)
WBC	278	30	21 (5.6%)
Vowels	1456	12	50 (3.4%)
Cardio	1831	21	176 (9.6%)
Musk	3062	166	97 (3.2%)
Satimage-2	5803	36	71 (1.2%)
Letter Recognition	1600	32	100 (6.25%)
Speech	3686	400	61 (1.65%)
Satellite	6435	36	2036 (32%)
Arrhythmia	452	274	66 (15%)
Ionosphere	351	33	126 (36%)
Mnist	7603	100	700 (9.2%)
Optdigits	5216	64	150 (3%)
ForestCover	286,048	10	2747 (0.9%)
Pendigits	6870	16	156 (2.27%)
Wine	129	13	10 (7.7%)
Seismic	2584	11	170 (6.5%)
Heart	224	44	10 (4.4%)

5.3. Experimental Results for Benchmark Datasets

In this section, we give the experimental results of MISC-OD for the benchmark datasets in Tables 3 and 4. The highest ROC or AP score is marked in bold, which means that the algorithm achieves the best performance for this dataset.

Table 3. ROC scores of outlier detector performance (the highest ROC score is marked in bold).

Dataset	KNN	LOF	ABOD	HBOS	IForest	CBLOF	LSCP	OCSVM	MISC-OD
Lympho	0.916	0.499	0.69	0.953	0.657	0.434	**0.955**	0.48	0.795
WBC	0.72	0.94	0.549	0.418	0.443	0.78	0.735	0.95	**0.954**
Vowels	0.57	0.619	0.455	0.624	0.608	**0.823**	0.789	0.425	0.527
Cardio	0.401	0.483	0.527	0.416	0.668	0.543	0.913	**0.945**	0.917
Musk	0.455	0.582	0.647	0.675	0.716	0.577	0.983	0.509	**0.986**
Satimage-2	0.505	0.717	0.412	0.438	0.781	0.698	0.408	0.776	**0.783**
Letter Recognition	0.465	**0.898**	0.711	0.742	0.728	0.476	0.836	0.964	0.83
Speech	0.926	0.722	0.697	0.651	0.911	0.632	0.764	0.531	**0.948**
Satellite	0.795	0.753	0.513	0.914	0.508	0.969	0.63	0.551	**0.98**
Arrhythmia	0.862	0.668	0.519	0.796	0.538	0.482	0.761	0.747	**0.914**
Ionosphere	0.615	0.686	0.728	0.877	0.487	0.411	0.631	**0.944**	0.92
Mnist	0.484	0.592	0.401	0.529	0.57	0.757	0.655	0.713	**0.868**
Optdigits	0.525	0.861	0.758	0.773	0.507	**0.866**	0.505	0.673	0.677
ForestCover	0.816	0.431	0.654	0.474	0.95	0.532	0.436	0.803	**0.829**
Pendigits	0.787	0.738	0.76	0.5	**0.949**	0.536	0.455	0.814	0.635
Wine	0.592	0.833	0.901	0.743	0.868	0.852	**0.934**	0.603	0.519
Seismic	0.918	0.978	0.844	0.804	0.626	0.877	0.629	0.491	**0.934**
Heart	0.527	**0.895**	0.444	0.782	0.582	0.442	0.809	0.74	0.87
Average ROC	0.66	0.716	0.622	0.672	0.672	0.649	0.712	0.703	**0.827**

Table 4. Average precision (AP) scores of outlier detector performance (the highest AP score is marked in bold).

Dataset	KNN	LOF	ABOD	HBOS	IForest	CBLOF	LSCP	OCSVM	MISC-OD
Lympho	0.874	0.248	0.535	0.929	0.486	0.152	**0.933**	0.692	0.62
WBC	0.58	0.91	0.324	0.127	0.165	0.887	0.603	**0.924**	0.37
Vowels	0.355	0.429	0.183	0.435	0.413	**0.735**	0.684	0.137	0.291
Cardio	0.101	0.225	0.291	0.123	0.501	0.315	**0.959**	0.917	0.676
Musk	0.182	0.98	0.47	0.512	0.573	0.365	0.875	0.263	**0.88**
Satimage-2	0.257	0.575	0.119	0.157	0.672	0.547	0.113	0.664	**0.678**
Letter Recognition	0.197	0.847	0.566	0.614	0.592	0.214	0.755	**0.946**	0.505
Speech	0.889	0.584	0.546	0.476	0.866	0.448	0.646	0.296	**0.922**
Satellite	0.693	0.629	0.27	0.871	0.262	0.954	0.446	0.326	**0.971**
Arrhythmia	0.793	0.502	0.278	0.694	0.307	0.222	0.641	0.621	**0.871**
Ionosphere	0.422	0.529	0.591	0.816	0.23	0.116	0.447	**0.916**	0.88
Mnist	0.226	0.389	0.102	0.293	0.355	0.635	0.483	0.57	**0.803**
Optdigits	0.287	0.792	0.636	0.659	0.26	**0.799**	0.258	0.509	0.516
ForestCover	**0.818**	0.146	0.482	0.211	0.925	0.298	0.153	0.705	0.744
Pendigits	0.681	0.607	0.64	0.25	**0.924**	0.304	0.182	0.72	0.452
Wine	0.388	0.75	0.852	0.615	0.802	0.778	**0.901**	0.404	0.278
Seismic	0.878	0.966	0.766	0.707	0.44	0.815	0.444	0.237	**0.901**
Heart	0.29	0.842	0.166	0.673	0.374	0.163	0.713	0.61	**0.805**
Average AP	0.495	0.608	0.434	0.509	0.508	0.485	0.568	0.58	**0.675**

5.4. Analysis of Experimental Results

The proposed MISCO-OD algorithm achieved the best performance, with an average ROC of 0.827 and an average precision of 0.675. In Table 3, MISC-OD exhibits the highest ROC in 9 out of 18 datasets. Additionally in Table 4, MISC-OD achieves the highest AP (average precision) in 8 out of 18 datasets.

It is noteworthy that, by analyzing the data in Tables 3 and 4, it can be found that the higher the data dimensionality, the better results the MISC-OD algorithm can achieve, for example, in the Musk, Letter Recognition, Speech, Arrhythmia, and Mnist datasets. This confirms that the MISC-OD algorithm is highly scalable and performs well with high-dimensional data, as we proposed in Section 4.2.

6. Conclusions

In this manuscript, we present a novel unsupervised outlier detection algorithm based on mutual information and reduced spectral clustering called MISC-OD. MISC-OD can be mainly divided into three stages: (1) constructing a mutual information matrix between features; (2) dividing the feature set into subsets using reduced spectral clustering; and (3) utilizing LOF for outlier detection within each subset and combining the outlier scores found within each subset. Finally, the outlier scores are given as output.

We carried out a large number of experiments on 18 benchmark datasets from OODS. The proposed MISCO-OD achieves the best performance, with an average ROC of 0.827 and an average precision of 0.675. In addition to providing better experimental results, the MISC-OD algorithm also has a high interpretability and scalability, as explained in Section 4.2.

In specific applications of MISC-OD, algorithm tuning can be performed as follows: (1) use domain knowledge to determine the initial values of hyperparameters; (2) if the

mutual information between two features is relatively small, mutual information can be set to 0. This has little impact on the results of the algorithm, but can speed up its execution.

In further study, we will investigate the parallel method of the proposed MISC-OD algorithm to apply the method to large data in real-life applications. In the first stage, if it is an ultra-high-dimensional dataset, it will face an extremely large-scale matrix decomposition problem and will need to be implemented in parallel computing frameworks such as MPL, MapReduce, and Spark. In the second stage, the anomaly detection tasks of different feature subsets can be assigned to different computing nodes to achieve parallel computing, and finally, the anomaly scores of each computing node are summarized. In this way, the execution speed of the algorithm is accelerated, which is very effective in real-time anomaly detection scenarios. In addition to the study of static datasets, the datastream outlier detection method considering mutual information between features will also be investigated in the future.

Author Contributions: Conceptualization, Y.H.; funding acquisition, W.L.; methodology, Y.H.; project administration, W.L.; supervision, W.L.; validation, Y.H., S.L., Y.G. and W.C.; writing—original draft, Y.H.; writing—review and editing, Y.H. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No. 61862011), the Guangxi Natural Science Foundation (No. 2019GXNSFGA245004), and the Innovation Project of Guangxi Graduate Education (No. YCBZ2023128).

Data Availability Statement: Data are contained within the article.

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

References

1. Hawkins, D.M. *Identification of Outliers*; Springer: Berlin/Heidelberg, Germany, 1980; Volume 11.
2. Zhang, S.; Li, B.; Li, J.; Zhang, M.; Chen, Y. A novel anomaly detection approach for mitigating web-based attacks against clouds. In Proceedings of the 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, New York, NY, USA, 3–5 November 2015; IEEE: Piscataway, NJ, USA, 2015.
3. Wang, Z.; Shao, L.; Cheng, K.; Liu, Y.; Jiang, J.; Nie, Y.; Li, X.; Kuang, X. ICDF: Intrusion collaborative detection framework based on confidence. *Int. J. Intell. Syst.* **2022**, *37*, 7180–7199. [CrossRef]
4. Seong, C.; Song, Y.; Hyun, J.; Cheong, Y.-G. Towards Building Intrusion Detection Systems for Multivariate Time-Series Data. In Proceedings of the 2nd Silicon Valley Cybersecurity Conference (SVCC), San Jose, CA, USA, 2–3 December 2021.
5. Phan, T.V.; Bauschert, T. DeepAir: Deep Reinforcement Learning for Adaptive Intrusion Response in Software-Defined Networks. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 2207–2218. [CrossRef]
6. Horchulhack, P.; Viegas, E.K.; Santin, A.O. Toward feasible machine learning model updates in network-based intrusion detection. *Comput. Netw.* **2022**, *202*, 108618. [CrossRef]
7. Tao, J.; Han, T.; Li, R. Deep-Reinforcement-Learning-Based Intrusion Detection in Aerial Computing Networks. *IEEE Netw.* **2021**, *35*, 66–72. [CrossRef]
8. Yu, L.; Wu, C.; Xiong, N.N. An Intelligent Data Analysis System Combining ARIMA and LSTM for Persistent Organic Pollutants Concentration Prediction. *Electronics* **2022**, *11*, 652. [CrossRef]
9. Xu, R.; Guo, Y.; Han, X.; Xia, X.; Xiang, H.; Ma, J. OpenCDA: An open cooperative driving automation framework integrated with co-simulation. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; IEEE: Piscataway, NJ, USA, 2021.
10. Topac, O.T.; Ha, S.Y.S.; Chen, X.; Gamble, L.; Inman, D.; Chang, F.-K. Hybrid Models for Situational Awareness of an Aerial Vehicle from Multimodal Sensing. *AIAA J.* **2023**, *61*, 305–314. [CrossRef]
11. Mansour, R.F.; Escorcia-Gutierrez, J.; Gamarra, M.; Villanueva, J.A.; Leal, N. Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model. *Image Vis. Comput.* **2021**, *112*, 104229. [CrossRef]
12. Pawar, K.; Attar, V. Deep learning approaches for video-based anomalous activity detection. *World Wide Web* **2019**, *22*, 571–601. [CrossRef]
13. Zhao, Y.; Deng, B.; Shen, C.; Liu, Y.; Lu, H.; Hua, X.S. Spatio-Temporal AutoEncoder for Video Anomaly Detection. In Proceedings of the 25th ACM International Conference on Multimedia (MM), Mountain View, CA, USA, 23–27 October 2017.
14. Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; Yu, P.S. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020.

15. Tao, J.; Lin, J.; Zhang, S.; Zhao, S.; Wu, R.; Fan, C.; Cui, P. Mvan: Multi-view attention networks for real money trading detection in online games. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019.
16. Cao, B.; Mao, M.; Viidu, S.; Yu, P. Collective fraud detection capturing inter-transaction dependency. In Proceedings of the KDD 2017 Workshop on Anomaly Detection in Finance, Halifax, NS, Canada, 14 August 2017.
17. Porwal, U.; Mukund, S. Credit card fraud detection in e-commerce: An outlier detection approach. *arXiv* **2018**, arXiv:1811.02196.
18. Paula, E.L.; Ladeira, M.; Carvalho, R.N.; Marzagão, T. Deep Learning Anomaly Detection as Support Fraud Investigation in Brazilian Exports and Anti-Money Laundering. In Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016.
19. Kumar, P.; Jain, R.; Chaudhary, S.; Kumar, S. Solving Community Detection in Social Networks: A comprehensive study. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 239–345.
20. Liu, Y.; Xiao, T. Pricing and Collection Rate Decisions and Reverse Channel Choice in a Socially Responsible Supply Chain With Green Consumers. *IEEE Trans. Eng. Manag.* **2020**, *67*, 483–495. [CrossRef]
21. Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* **2019**, *267*, 1–38. [CrossRef]
22. Zhao, Y.; Nasrullah, Z.; Hryniewicki, M.K.; Li, Z. LSCP: Locally selective combination in parallel outlier ensembles. In Proceedings of the 2019 SIAM International Conference on Data Mining, Calgary, AB, Canada, 2–4 May 2019; SIAM: Philadelphia, PA, USA, 2019.
23. Markovic, N.; Vahle, D.; Staudt, V.; Kolossa, D. Condition Monitoring for Power Converters via Deep One-Class Classification. In Proceedings of the 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Virtual, 13–16 December 2021.
24. Liu, Y.; Li, Z.; Zhou, C.; Jiang, Y.; Sun, J.; Wang, M.; He, X. Generative Adversarial Active Learning for Unsupervised Outlier Detection. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1517–1528. [CrossRef]
25. Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S.A.; Binder, A.; Müller, E.; Kloft, M. Deep One-Class Classification. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.
26. Xu, X.; Liu, H.; Yao, M. Recent Progress of Anomaly Detection. *Complexity* **2019**, *2019*, 2686378. [CrossRef]
27. Wang, H.; Bah, M.J.; Hammad, M. Progress in outlier detection techniques: A survey. *IEEE Access* **2019**, *7*, 107964–108000. [CrossRef]
28. Goldstein, M.; Dengel, A. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012 Poster Demo Track* **2012**, *1*, 59–63.
29. Xiao, T.; Zhang, C.; Zha, H. Learning to Detect Anomalies in Surveillance Video. *IEEE Signal Process. Lett.* **2015**, *22*, 1477–1481. [CrossRef]
30. Latecki, L.J.; Lazarevic, A.; Pokrajac, D. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2007.
31. Aggarwal, C.C. *An Introduction to Outlier Analysis*; Springer: Berlin/Heidelberg, Germany, 2017.
32. Yang, Y.; Chen, L.; Fan, C. ELOF: Fast and memory-efficient anomaly detection algorithm in data streams. *Soft Comput.* **2021**, *25*, 4283–4294. [CrossRef]
33. Tang, J.; Chen, Z.; Fu AW, C.; Cheung, D.W. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, 6–8 May 2002 Proceedings 6*; Springer: Berlin/Heidelberg, Germany, 2002.
34. Sikder, M.N.K.; Batarseh, F.A. Outlier detection using AI: A survey. In *AI Assurance*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 231–291.
35. Chen, J.; Sathe, S.; Aggarwal, C.; Turaga, D. Outlier detection with autoencoder ensembles. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; SIAM: Philadelphia, PA, USA, 2017.
36. Hsu, Y.-F.; Matsuoka, M. A Deep Reinforcement Learning Approach for Anomaly Network Intrusion Detection System. In Proceedings of the IEEE 9th International Conference on Cloud Networking (CloudNet), Virtual, 9–11 November 2020.
37. Alimohammadi, H.; Chen, S.N. Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis. *Expert Syst. Appl.* **2022**, *191*, 116371. [CrossRef]