12-2020

Heterogeneous univariate outlier ensembles in multidimensional data

Guansong PANG
Singapore Management University, gspang@smu.edu.sg

Longbing CAO
University of Technology Sydney

Heterogeneous Univariate Outlier Ensembles in Multidimensional Data

GUANSONG PANG, The University of Adelaide
LONGBING CAO, University of Technology Sydney

In outlier detection, recent major research has shifted from developing univariate methods to multivariate methods due to the rapid growth of multidimensional data. However, one typical issue of this paradigm shift is that many multidimensional data often mainly contains *univariate outliers*, in which many features are actually irrelevant. In such cases, multivariate methods are ineffective in identifying such outliers due to the potential biases and the curse of dimensionality brought by irrelevant features. Those univariate outliers might be well detected by applying univariate outlier detectors in individually relevant features. However, it is very challenging to choose a right univariate detector for each individual feature since different features may take very different probability distributions. To address this challenge, we introduce a novel Heterogeneous Univariate Outlier Ensembles (HUOE) framework and its instance ZDD to synthesize a set of heterogeneous univariate outlier detectors as base learners to build heterogeneous ensembles that are optimized for each individual feature. Extensive results on 19 real-world datasets and a collection of synthetic datasets show that ZDD obtains 5%–14% average AUC improvement over four state-of-the-art multivariate ensembles and performs substantially more robustly w.r.t. irrelevant features.

CCS Concepts: • **Computing methodologies** → **Anomaly detection**; **Bagging**;

Additional Key Words and Phrases: Outlier detection, outlier ensemble, anomaly detection, univariate outlier, multidimensional data, heterogeneous data

ACM Reference format:
Guansong Pang and Longbing Cao. 2020. Heterogeneous Univariate Outlier Ensembles in Multidimensional Data. *ACM Trans. Knowl. Discov. Data* 14, 6, Article 68 (September 2020), 27 pages.
https://doi.org/10.1145/3403934

1 INTRODUCTION

Outliers are data objects that are significantly different from the majority of objects. Outlier detection can offer important insights into many real-world domains, such as cybersecurity, finance, and health care. For example, outlier detection is widely used to detect network attacks and credit card frauds [2, 10]. In general, outlier detection methods can be categorized into univariate and

This work was partially sponsored by the Australian Research Council Discovery Grant DP190101079.
This work was mainly done when Guansong Pang was with the University of Technology Sydney.
Authors' addresses: G. Pang, The University of Adelaide, Adelaide SA 5005, Australia; email: guansong.pang@adelaide.edu.au; L. Cao, University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Australia; email: longbing.cao@uts.edu.au.

1556-4681/2020/09-ART68 $15.00
https://doi.org/10.1145/3403934

multivariate methods, of which univariate methods detect outliers that are exceptional in individual features (i.e., *univariate outliers*) while multivariate methods identify outliers in a multidimensional space. Univariate methods [6] dominated the area of outlier detection for a long period in the last century, but the ubiquitous multidimensional data and the endeavor to build more sophisticated models have led to supreme efforts on multivariate outlier detection methods in recent years. Consequently, univariate methods have been often disregarded in the development and evaluation of outlier detection in multidimensional data [5, 8, 16, 23, 27, 32, 33, 36, 37]. Such a paradigm shift may result in some critical issues. One major issue is that multivariate methods may fail to detect univariate outliers due to the potential biases and curse of dimensionality brought by irrelevant features, i.e., features that cannot highlight or explain the outlyingness of the outliers [4, 30]. By contrast, univariate methods are not affected by irrelevant features and thus work well in identifying univariate outliers when there are many irrelevant features. Another major issue is that multivariate methods often treat all features in a homogeneous way and thus fail to capture heterogeneous distributions of different features, whereas it may be more appropriate to identify a set of heterogeneous univariate methods to capture complex feature heterogeneities [9] in such data.

However, due to the aforementioned paradigm shift, as far as we know, there is very limited research on inventing appropriate univariate outlier detection methods for multidimensional data. This work aims to fill this gap. There are generally the following three major challenges in developing univariate methods for multidimensional data: (i) since univariate outlier detectors are often only applicable for a certain distribution while heterogeneous data distributions may exist across the features, it is very difficult to determine suitable univariate detectors for specific features; (ii) it would be challenging to properly combine the D detection results from D features since univariate detectors work in a feature-wise manner; and (iii) most univariate outlier detectors, such as the popular Z-Score, Dixon's Q test (Dixon test for short) and boxplot [6], are frustrated with the presence of outliers and may face the swamping or masking problem, i.e., an outlier is masked as an inlier due to the presence of inliers (swamping) or other outliers (masking).

Some recent efforts have been made to partially address these challenges. For example, studies in the statistic community [14, 21] focus on improving the robustness and theoretical bounds of the popular univariate detectors, which help address the above third issue. However, since these efforts focus on univariate data, they do not consider the first two issues. Some recent studies [18, 33–35] have attempted to use multiple *heterogeneous* multivariate outlier detection methods (i.e., different outlier detection methods or the same method with different parameter settings) to improve the detection accuracy on heterogeneous datasets. However, the heterogeneous ensembles in these studies work on multidimensional feature subspaces, which fail to work effectively in the presence of irrelevant features, and they also ignore the heterogeneities between individual features, thus ineffective for data with heterogeneous features.

To address the aforementioned three challenges, this article introduces a novel outlier detection framework to learn Heterogeneous Univariate Outlier Ensembles (HUOE) to identify univariate outliers in multidimensional data with feature heterogeneities. HUOE first defines multiple different univariate detectors to effectively compute outlier scores w.r.t. different data distributions in each feature. It then defines an outlier ranking evaluation measure to find an optimal combination of the outlier scores obtained from these heterogeneous outlier detectors, resulting in a high-quality outlier ranking per feature. HUOE finally exploits the correlation between the outlier rankings to integrate them into one final outlier ranking. Additionally, HUOE operates on random subsamples to avoid the swamping and masking problems [38].

We further instantiate the HUOE framework into an instance, called ZDD,[1] which uses multiple specifications of Z-Score, Dixon test, and data-dependent-based outlier detectors to well capture outlierness in a variety of data distributions. ZDD then defines a *Cantelli*'s inequality-based outlier ranking evaluation measure to produce an optimized outlier score ranking per feature. ZDD further computes a weight for each ranking based on its internal score distribution and its correlation with other rankings, and then conducts a weighted combination of all the rankings to obtain a global outlier ranking.

Accordingly, this article makes the following three major contributions.

- This is the first work to have a comprehensive empirical study of using *univariate* outlier detection methods to identify outliers in a large number of real-world multidimensional datasets. It reveals that many real-world multidimensional data mainly contains univariate outliers, and consequently univariate methods are better choice than multivariate methods in such cases.
- In contrast to the existing multivariate methods that treat each feature equally and work on feature subspaces or the full space, the proposed HUOE framework can model complex heterogeneous data distributions within individual features. As a result, our framework enables more effective solutions to detect univariate outliers in multidimensional data with many irrelevant or heterogeneous features than multivariate methods.
- The HUOE-instantiated ZDD method defines heterogeneous univariate outlier detectors and *Cantelli*'s inequality-based outlier ranking evaluation measures to yield feature-wise optimal heterogeneous outlier ensembles for multidimensional data.

Extensive experiments on 19 real-world datasets and 1 synthetic dataset show the following: (i) ZDD obtains 5%–14% average AUC (Area Under the receiver operating characteristic Curve) improvement over four state-of-the-art multivariate ensembles, and perform much more stably than three simple ensembles of univariate detectors; (ii) surprisingly, the three simple univariate ensembles consistently outperform the four advanced multivariate ensembles; and (iii) the benefit of each module of ZDD is empirically justified via an ablation study. Also, our empirical results on a set of synthetic datasets show that univariate methods show substantially better robustness w.r.t. irrelevant features than multivariate methods, in which ZDD (or its variants) achieves the best robustness.

The rest of this article is organized as follows. The related work is given in Section 2. The HUOE framework is introduced in Section 3. The instantiated model, ZDD, is introduced in Section 4. A theoretical analysis is provided in Section 5. A series of empirical results is presented in Section 6. After discussing the research implications in Section 7, we conclude this work in Section 8.

2 RELATED WORK

2.1 Univariate Outlier Detection Methods

Many univariate outlier detection methods were developed in the statistic community in the last century [6]. Most of these methods were designed for specific probability distributions. Two popular methods of this kind are Z-Score and Dixon test. Z-Score [19] uses the difference between a given value and the mean divided by the standard deviation as the outlier score of the tested value. It may apply to any distribution in which the mean and variance are defined. It can obtain a guaranteed error bound based on the 68-95-99.7 rule by making the normal distribution assumption of the data. In contrast to the Z-Score that focuses on the global statistics

[1] ZDD comes from the first character of the three different univariate detectors used in this instantiation.

to characterize the outlierness, the Dixon test [11] focuses on local information, which defines the outlier scores w.r.t. the distance of a given value to its nearest-neighbor values normalized by the range of all the values. The Dixon test is well suited to exponential or extreme value distributions. Many variants of these two methods can be found in [6].

There are other popular methods such as boxplot [14, 39] that use the quartile-based summary statistics to determine outliers, e.g., the query values that are smaller than the first quartile or larger than the third quartile. These methods attempt to directly label the outliers with certain error bounds, while we focus on assigning outlier scores to the objects and returning an outlier ranking. Some recent studies are dedicated to understand some important properties of popular methods, such as addressing the masking and swamping problems for Z-Score [40]. To gain more information for measuring univariate outlierness, one interesting idea is to convert univariate data into multivariate data, e.g., each data object in univariate time series data can be represented by the data object together with a set of its consecutive context neighbors, and then employ multivariate methods to detect univariate outliers [22].

However, the above studies on univariate data ignore the heterogeneities in each individual feature. Also, they focus on univariate data, so they do not deal with the problem of combining multiple outlier scoring results from different features.

2.2 Multivariate Outlier Detection Methods

There have been numerous multivariate outlier detection methods in the data mining and machine learning community. These methods can be generally categorized into the following five groups [2]: probabilistic methods, e.g., Gaussian mixture model (GMM) [16]; linear model-based method, e.g., principal component analysis (PCA) [32]; distance-based methods, e.g., kNN [37]; density-based methods, e.g., local outlier factor [36]; and clustering-based methods, e.g., CBUID [23].

One major issue with these methods is that they assume all the features are drawn from homogeneous distributions, which make them less effective in datasets with heterogeneous distributions across the features. Another major issue is due to the presence of irrelevant features. These irrelevant features form a main cause to the curse of dimensionality [44], and they can hide outliers and consequently become noise to the multivariate outlier detectors.

2.3 Outlier Ensembles

The ensemble methods have been well established for learning tasks such as clustering and classification, but ensemble learning for outlier detection attracts wide attention only in recent years [1, 42]. It has been shown in [24, 26, 31, 34, 35, 37, 38, 41, 43] that building outlier ensembles can substantially improve the efficacy of the above traditional multivariate detectors. We discuss the following two main groups of relevant outlier ensembles: subspace-based methods and subsample-based methods. Subspace-based methods [24, 26, 27] work on a set of relevant or randomly selected feature subspaces, while subsample-based methods [28, 31, 37, 38, 41, 43] build ensembles using a set of randomly selected subsamples. As far as we know, existing outlier ensembles focus on multivariate methods and no work has been reported on how to make use of univariate outlier detection methods to build effective outlier ensemble methods. As shown in Section 6, a proper ensemble of univariate detectors can perform significantly better than state-of-the-art multivariate methods.

There have been some work dedicated to handling some issues of data heterogeneities [9] by building heterogeneous ensembles, including the use of the same outlier scoring method but with different parameter settings [33, 35] and the use of diversified outlier scoring methods [34]. Additionally, recent studies have also introduced some score unification methods [18, 25]) to properly transform outlier scores from different detectors into a space of the same semantic for the final aggregation of the scores. Some other studies (e.g., [34, 35]) focus on the score combination part and

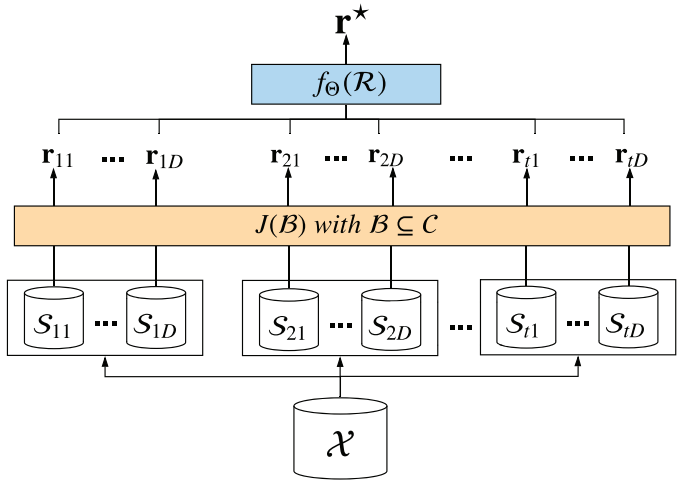

Fig. 1. Our proposed HUOE framework. HUOE first applies a set of heterogeneous univariate outlier detectors $C = \{\phi_1, \phi_2, \ldots, \phi_K\}$ to each univariate subsample $\mathcal{S}_{ij}$ of $\mathcal{X}$ to obtain $|C|$ outlier rankings per feature. An objective function J is then defined to select and aggregate an optimal subset of $|C|$ outlier rankings to a unified outlier ranking $\mathbf{r}_{ij}$. The same process is applied to all D individual features of t subsamples, obtaining a set of $t \times D$ rankings, $\mathcal{R} = \{\mathbf{r}_{11}, \mathbf{r}_{12}, \ldots, \mathbf{r}_{tD}\}$, which is lastly aggregated by a weighted combination function f to produce a final outlier ranking $\mathbf{r}^\star$.

use correlations between outlier scores from different detectors to selectively combine their output scores through the goal of diversifying the obtained outlier ensembles. These ensembles improve the original individual outlier detectors, but they focus on the high-level data heterogeneity by ignoring the feature-level heterogeneity, which may render the ensembles ineffective in datasets with strong feature heterogeneity.

3 THE PROPOSED HUOE FRAMEWORK

The general outlier detection problem can be stated as follows. Given a dataset $\mathcal{X}$ with D features, we aim to return a ranking of the data objects based on their outlierness, and identify the data objects that have the largest outlier scores as outliers. The class labels are not available in this setting, i.e., we focus on unsupervised outlier detection, as it is too costly to collect these label information in many outlier detection applications. A rarely explored subproblem within outlier detection is how to effectively detect univariate outliers in multivariate datasets with many irrelevant and/or heterogeneous features.

We introduce the HUOE framework to address this problem, which builds a feature-wise heterogeneous outlier ensemble to capture the fine-grained feature heterogeneities. The resulted model can well identify univariate outliers in data with heterogeneous features and/or irrelevant features. As described in Figure 1, given a dataset $\mathcal{X}$ with D features, HUOE first samples a set of t random subsamples, $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_t\}$ (i.e., $\mathcal{S}_i \subset \mathcal{X}$), and employs a set of heterogeneous univariate base outlier detectors, $C = \{\phi_1, \phi_2, \ldots, \phi_K\}$, on each feature of the subsample $\mathcal{S}_i$ to identify outliers w.r.t. different data distributions taken in individual features. HUOE then defines an objective function J to unify a selective subset of the outlier rankings produced by the base detectors in C into an optimal ranking, $\mathbf{r}_{ij}$ (i.e., there are $2^{|C|} - 1$ subsets of the collection of $|C|$ base detectors; $\mathcal{B}$ in Figure 1 is one of the possible subsets). This procedure is iteratively applied to all D features of t subsamples, resulting in a set of $t \times D$ outliers rankings, i.e., $\mathcal{R} = \{\mathbf{r}_{11}, \mathbf{r}_{12}, \ldots, \mathbf{r}_{tD}\}$. HUOE further

defines a weighted combination function f to combine all the outlier rankings in $\mathcal{R}$ so as to obtain a final outlier ranking $\mathbf{r}^\star$.

HUOE is very different from the existing outlier ensemble frameworks in that: (i) HUOE models the low-level feature heterogeneities, while most current frameworks are for homogeneous ensembles and the existing heterogeneous ensembles are built upon the data object level; as a result, HUOE works better than the other frameworks when the data distributions in different features are very different; and (ii) HUOE is a univariate outlier ensemble for identifying univariate outliers in data with many irrelevant features, whereas the existing frameworks focus on building ensembles on the full feature space or subspaces to identify multivariate outliers and they are often misled by irrelevant features and thus do not work well when data contains many irrelevant features.

Note that previous subspace-based ensembles like [24, 26] may be reduced to univariate feature subspace-based methods by setting the subspace size to be one. However, these ensembles focus on reducing the influence of irrelevant features on multivariate methods and also do not consider the feature heterogeneity issue. HUOE is introduced to provide a novel perspective of addressing the issues of both irrelevant features and feature heterogeneity in a univariate fashion.

We introduce the motivation of each component of HUOE in detail in the following subsections.

3.1 Building a Set of Heterogeneous Univariate Base Outlier Detectors C on Each Feature of Subsamples

Features in real-world datasets often follow different data distributions. Moreover, each feature may be taken from a mixture of data distributions [9]. Since outliers are defined per data distribution, outlier detection methods designed for different distributions are required to identify these distribution-sensitive outliers. To address this issue, HUOE first defines a set of heterogeneous univariate outlier detection methods, $C = \{\phi_1, \phi_2, \ldots, \phi_K\}$, where $\phi_i : \mathcal{X}_{.k} \mapsto \mathbb{R}$ is a univariate outlier scoring method, and ϕ_i and ϕ_j are different methods or the same method but with different parameter settings. These heterogeneous detectors are then used as base learners to build a heterogeneous ensemble to detect the aforementioned heterogeneous outliers.

The previous work on outlier ensembles [27, 31, 37, 38] has shown that building multivariate outlier detectors on random subsamples of the full dataset helps address the swamping and masking problems, leading to significant improvement of detection accuracy. Motivated by this success, the base detectors in HUOE also operate on subsamples $\mathcal{S}$ to address the same problems for univariate methods.

3.2 Finding an Optimal Combination of the Heterogeneous Detectors w.r.t. the Objective Function J

Given a set of $|C|$ outlier scores output by the heterogeneous outlier detectors per feature, a key problem is how to reasonably integrate them to produce a high-quality outlier ranking. HUOE defines an objective function J to find a selective combination of the outlier scores to maximize a quality measure of outlier rankings:

$$\mathcal{B}^* = \arg\max_{\mathcal{B} \in \mathcal{P}} J(\mathcal{B}). \tag{1}$$

where $\mathcal{P} = P(C)$ is the power set of C excluding the empty set.

Since C contains a set of different outlier detectors, their detection results may disagree. The J function is to find $|\mathcal{B}^*|$ outlier detectors that work consistently and/or complement with each other by filtering conflicting results. We then aggregate the $|\mathcal{B}^*|$ outlier rankings to get a unified outlier ranking for each feature of each subsample.

3.3 Combining the Outlier Rankings in $\mathcal{R}$

Since J works on a feature-wise manner, HUOE obtains a set of $t \times D$ outlier rankings after applying J to all D features of the t subsamples, namely $\mathcal{R} = \{\mathbf{r}_{11}, \mathbf{r}_{12}, \ldots, \mathbf{r}_{tD}\}$. Among these rankings, only a certain percentage of them is from the relevant features and the others are from the irrelevant features. HUOE therefore defines a function f below to have a weighted combination of these rankings:

$$\mathbf{r}^{\star} = f_{\Theta}(\mathcal{R}) = \omega_{11}\mathbf{r}_{11} + \omega_{12}\mathbf{r}_{12} + \cdots + \omega_{tD}\mathbf{r}_{tD}, \tag{2}$$

where $\Theta = \{\omega_{11}, \ldots, \omega_{tD}\}$ are the weights to be learned.

Alternatively, this stage may be an outlier ranking selection for only retaining highly relevant rankings rather than the weighted combination. Here, we focus on the weighted combination because the number of rankings in $\mathcal{R}$ is very large in high-dimensional data, and it can be very computationally expensive to perform an optimal ranking selection.

4 A HUOE'S INSTANCE: ZDD

The HUOE framework is instantiated into a heterogeneous univariate ensemble model, ZDD. To leverage the advantages of different types of outlier detectors, ZDD uses three different types of base detectors to build an ensemble per univariate input. An exhaustive search and the *Cantalli*'s inequality are then used to offer an optimal combination of these base detectors. Homophily weights are further used to well combine the outlier scores produced in each individual feature.

4.1 Specifying the Base Outlier Detector Set with Z-Score, Dixon Test, and kNN

Three different types of univariate outlier detection methods are used in instantiating HUOE, including Z-Score, Dixon test, and kNN, because these methods can well-complement each other in detecting different distribution-sensitive outliers (See Section 6.5.4 for details). Below, we introduce these three outlier scoring methods in detail.

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with $\mathbf{x}_i \in \mathbb{R}^D$, let $\mathbf{x} \in \mathcal{X}$ and $\mathcal{S}_{ij}$ be the j-th feature of its i-th subsample that contains s randomly selected data objects with replacement, Z-Score defines the outlierness of the value of $\mathbf{x}$ in the j-th feature, x_j, as follows:

$$z\_score(x_j) = \left(\frac{x_j - \mu_{\mathcal{S}_{ij}}}{\sigma_{\mathcal{S}_{ij}}}\right)^2, \tag{3}$$

where $\mu_{\mathcal{S}_{ij}}$ and $\sigma_{\mathcal{S}_{ij}}$ are the mean and standard deviation of the univariate random subsample $\mathcal{S}_{ij}$. Note that the squared Z-Score is used because the sum of squared Z-Scores is always equal to the number of Z-Score values in different univariate samples, which offers not only a natural way of capturing suspicious negative deviations but also score normalization.

The Dixon test uses the difference between the given object $\mathbf{x}$ and its nearest neighbor to define the outlierness:

$$dixon(x_j) = \frac{|x_j - nn_{x_j}|}{max_j - min_j}, \tag{4}$$

where nn_{x_j} is the closest value to x_j, max_j is the maximum value in $\mathcal{S}_{ij}$, and min_j is the minimum value.

Motivated by the success of subsample distance-based methods reported in [31, 37, 38], we design a univariate average kNN outlier scoring method as follows:

$$knn(x_j) = \frac{1}{k}\sqrt{\sum_{x'_j \in \mathcal{N}_{x_j}} (x_j - x'_j)^2}, \tag{5}$$

where $\mathcal{N}_{x_j}$ denotes the set of k nearest neighbors of x_j.

Let $\mathcal{A} = \{z\_score, dixon, knn\}$ be the set of basic outlier detectors. ZDD then applies these three base methods to each $\mathcal{S}_{ij}$ to generate a set of $|C| = c|\mathcal{A}|$ heterogeneous detectors per univariate input, with each base method specified with c different parameter settings (see Section 6.2 for the detailed specifications). Alternatively, we can apply one of these base methods with different parameter settings to each $\mathcal{S}_{ij}$ to produce a set of heterogeneous detectors, however, the learning ability relies on a single detector. In contrast, ZDD learns a much better ensemble than the alternative method in heterogeneous data (see Section 6.5.4 for empirical support of this observation).

4.2 Exhaustive Search of the Optimal Combination Using *Cantelli*'s Inequality-based Outlier Ranking Measure

We obtain a set of $|C| = K$ outlier score rankings for each feature of a subsample after the above stage. Our next task is to find an optimal selective combination of these score rankings as in Equation (1). The same operation will be applied to aggregate $|C|$ score rankings in each of the D feature of each subsample, so this stage will yield a set of $t \times D$ outlier rankings if t subsamples are used.

Specifically, let $\mathcal{B} \in \mathcal{P}$, the J function is specified as:

$$J(\mathcal{B}) = \psi(\mathbf{r}_{\mathcal{B}}), \tag{6}$$

where $\psi : \mathbb{R}^N \mapsto \mathbb{R}$ is an outlier ranking evaluation measure function, and $\mathbf{r}_{\mathcal{B}} = \sum_{\phi \in \mathcal{B}} g(\phi(\mathcal{X}_{.j}))$ is a combined outlier score ranking of $|\mathcal{B}|$ outlier score lists, in which ϕ returns an outlier score list for a univariate input $\mathcal{X}_{.j}$ and g is a score unification function that normalizes heterogeneous outlier score rankings into comparable ones.

Two key ingredients of the objective function J are the combination search methods (i.e., how to generate $\mathcal{B}$) and the outlier ranking quality measures (i.e., the ψ function). There are generally two types of search methods: exhaustive and heuristic search. An exhaustive search is computationally expensive if $|C|$ is large, i.e., the search space is $2^{|C|} - 1$, but it produces a globally optimal solution. A heuristic search, such as the breadth-first or depth-first search, is efficient but may produce a suboptimal solution. Since the detectors $|C|$ used is quite small, the exhaustive search is used to find a *globally optimal* combination solution.

The outlier ranking quality evaluation is essentially an internal outlier detection evaluation problem, i.e., evaluating outlier rankings without class labels. Internal clustering evaluation measures have been well established, while very limited related work has been reported on unsupervised outlier detection [29]. The work reported in [29] uses maximum margin classifiers to evaluate the deviant distance of each outlier candidate to the classification boundary as the quality criterion. The outlier ranking that produces the average largest deviant distance for all the outlier candidates is of the best quality. As shown in [29], the output best outlier ranking has high correlation to the true ranking. However, this method has a time complexity of $O(N^3)$, which is computationally prohibitive to be used in our framework, since the evaluation measure is needed to be recursively used in each feature of different subsamples. We introduce a linear-time and statistically sound *Cantelli*'s inequality-based outlier ranking measure below to address this efficiency issue.

Definition 4.1 (Cantelli's Inequality-based Outlier Ranking Measure). Given an outlier score ranking $\mathbf{r} \in \mathbb{R}^N$ produced on $\mathcal{X}_{.j}$, in which large scores indicate high outlierness, and let μ and σ^2 be its expected value and variance, then its overall ranking quality is defined as follows:

$$\psi_{ci}(\mathbf{r}) = \frac{1}{|O|} \sum_{x_j \in O} [\mathbf{r}(x_j) - med_{\mathbf{r}}], \tag{7}$$

where $O \subset \mathcal{X}_{.j}$ is the set of outlier candidates that have outlierness no less than $\mu + \alpha\sigma$, $\mathbf{r}(x_j)$ returns the outlier score of x_j, and $med_{\mathbf{r}}$ denotes the median outlier score of the inlier candidates, i.e., $\mathcal{X} \setminus O$.

The purpose of $\psi_{ci}(\cdot)$ is to obtain the outlier candidate set O, which is built upon the *Cantelli*'s inequality. We show in Section 5.3 that we can obtain an outlier candidate set with a false positive upper bound of $\frac{1}{1+\alpha^2}$ due to the properties of the *Cantelli*'s inequality, in which α is a user-defined parameter of determining the bound.

Similar to [29], Equation (7) aims to maximize the margin between outlier candidates and inlier candidates. One main difference is that Equation (7) simplifies the problem and maximizes the margins between their outlier scores; while the measure in [29] performs this maximization in the original data space, which leads to substantially higher time complexity than Equation (7).

Combining Equations (6) and (7), we aim to obtain the optimal $\mathcal{B}^*$ by:

$$\mathcal{B}^* = \arg\max_{\mathcal{B}\in\mathcal{P}} \psi_{ci}(\mathbf{r}_{\mathcal{B}}). \tag{8}$$

We then combine the outlier scores resulted from the detectors in $\mathcal{B}^*$ by:

$$\mathbf{r}_{\mathcal{B}^*} = \sum_{\phi\in\mathcal{B}^*} g(\phi(\mathcal{X}_{.j})). \tag{9}$$

The ℓ_1-norm length-based normalization is used in the score unification g function: $g(r_i) = \frac{r_i}{||\mathbf{r}||_1}$, where r_i is an entry of the outlier ranking vector $\mathbf{r}$. Unlike the unification methods in [18, 25] that assume the distribution of the outlier scores following a specific probability distribution, the ℓ_1-norm length-based normalization does not have this assumption. Our experiments also show that the ℓ_1-norm length-based normalization produces better and more stable detection performance than the methods in [18, 25].

4.3 Homophily Weights for the Weighted Combination of Univariate Outlier Rankings

Lastly, we need to learn the weight parameters $\Theta = \{\omega_{11}, \omega_{12}, \ldots, \omega_{tD}\}$ in Equation (2) to highlight high-quality outlier rankings in the final ranking aggregation. Homophily weights are defined below to assign large weights to outlier rankings that are consensus to other high-quality outlier rankings in the ranking set $\mathcal{R}$.

Definition 4.2 (Homophily Weight). Given an outlier ranking $\mathbf{r} \in \mathcal{R}$, its homophily weight ω is defined as follows:

$$\omega = \sum_{\mathbf{r}'\in\mathcal{R}\setminus\mathbf{r}} \psi_{ci}(\mathbf{r})\rho(\mathbf{r},\mathbf{r}')\psi_{ci}(\mathbf{r}'), \tag{10}$$

where ρ is a correlation coefficient.

Equation (10) states that the ranking $\mathbf{r}$ has the highest quality if and only if (i) it has a large outlierness margin between candidate inliers and outliers, i.e., having a large ψ_{ci}; and (ii) it is also strongly associated with the other outlier rankings that have a large ψ_{ci}. Since the ranks of the data objects in $\mathbf{r}$ are important, *Spearman*'s rank correlation coefficient is used to specify ρ to capture the rank-sensitive correlation. After obtaining all the weights in Θ by Equation (10), we then obtain the final outlier ranking $\mathbf{r}^\star$ by the weighted combination function $f_\Theta(\mathcal{R})$ as in Equation (2).

4.4 The Algorithms and Their Time Complexities

The procedure of ZDD is presented in Algorithm 1. In Steps 2–8, ZDD applies the three detectors in $\mathcal{A}$ with c different parameter settings to obtain $|C| = c|\mathcal{A}|$ heterogeneous detectors. ZDD achieves this by using bootstrapping approaches, i.e., to estimate the parameters using different random subsamples (see Section 6.2 for details). After obtaining $|C|$ candidate outlier rankings per feature of each subsample, Steps 9–12 use the exhaustive search and the *Cantelli*'s inequality-based ranking quality measure to produce the globally optimal subset of these $|C|$ outlier rankings and

unify them into one outlier ranking, $\mathbf{r}_{ij}$. Particularly, in Step 10, $\mathcal{P}_j$ is the power set of the $|C_j|$ rankings obtained in Steps 2–8. Steps 2–12 are repeated t times and yield a set of t outlier rankings in each feature, resulting in a total of $t \times D$ rankings for D features. Steps 14–18 compute a weight for each ranking $\mathbf{r}_{ij}$. ZDD finally outputs a weighted combination of the $t \times D$ rankings in Steps 19–20.

ALGORITHM 1: *ZDD*

Require: $\mathcal{X}$ - data objects, s - subsampling size, t - bagging size
Ensure: $\mathbf{r}^\star$ - an outlier ranking of objects

1: **for** $i = 1$ to t **do**
2: **for** $k = 1$ to c **do**
3: $\mathcal{S} \leftarrow$ Randomly select a subsample of size s from $\mathcal{X}$
4: **for** $j = 1$ to D **do**
5: Apply univariate outlier detectors in $\mathcal{A}$ to $\mathcal{S}_{ij}$
6: $C_j \leftarrow C_j \cup \mathcal{A}$
7: **end for**
8: **end for**
9: **for** $j = 1$ to D **do**
10: $\mathcal{B}^* \leftarrow \arg\max_{\mathcal{B} \in \mathcal{P}_j} \psi_{ci}(\mathbf{r}_{\mathcal{B}})$
11: $\mathbf{r}_{ij} \leftarrow \sum_{\phi \in \mathcal{B}^*} g(\phi(\mathcal{X}_{\cdot j}))$
12: **end for**
13: **end for**
14: **for** $i = 1$ to t **do**
15: **for** $j = 1$ to D **do**
16: $\omega_{ij} \leftarrow \sum_{i'} \sum_{j'} \psi_{ci}(\mathbf{r}_{ij}) \rho(\mathbf{r}_{ij}, \mathbf{r}_{i'j'}) \psi_{ci}(\mathbf{r}_{i'j'})$
17: **end for**
18: **end for**
19: $\mathbf{r}^\star \leftarrow \omega_{11}\mathbf{r}_{11} + \omega_{12}\mathbf{r}_{12} + \cdots + \omega_{tD}\mathbf{r}_{tD}$
20: **return** $\mathbf{r}^\star$

In Algorithm 1, since the outer two loops have linear time complexity w.r.t. the bagging size m and the dimensionality D, the time complexity of ZDD is determined by the more complex inner operations in Steps 4–7 and Steps 9–12. The time complexity of the outlier scoring methods in Step 5 is between $O(\log N)$ and $O(N \log N)$. Since we apply $|C|$ (i.e., $c|\mathcal{A}|$) detectors in D features, we have $O(|C|DN \log N)$ in Steps 2–8. Finding the optimal solution in Step 10 takes $O(2^{|C|} - 1)$, resulting in $O(2^{|C|}D)$ in Steps 9–12. Obtaining t outlier rankings using t subsamples result in $O(|C|tDN \log N + 2^{|C|}tD)$. The pairwise *Spearman*'s rank correlation computation takes $O(D^2)$ in Step 11. Therefore, the worst-case time complexity is $O(|C|tDN \log N + 2^{|C|}tD + D^2)$. $|C|$ and t are typically very small constants and far smaller than N, e.g., using $|C| = 6$ and $t = 10$ enables ZDD to perform very well, so the complexity can be simplified to $O(DN \log N + D^2)$. Hence, ZDD is expected to have the time complexity that is nearly linear w.r.t. N and quadratic w.r.t. D.

5 THEORETICAL FOUNDATION

5.1 When Can Univariate Methods Outperform Multivariate Methods?

There are two main reasons for when multivariate outlier detection methods are ineffective in multidimensional data. One is due to the curse of dimensionality brought by the concentration of distances in high-dimensional spaces.

Theorem 5.1 (Concentration of All p-norms [17]). *Let $X = (X_1, X_2, \ldots, X_D)$ be a random vector with i.i.d. components: $X_i \sim \mathcal{F}$. Then,*

$$\lim_{D\to\infty} \frac{\sqrt{Var(||X||_p)}}{E(||X||_p)} = 0, \tag{11}$$

where $p \in (0, \infty]$, i.e., including Minkowski norms and fractional norms.

This theorem indicates that the relative contrast of any given p-norms vanishes as the dimension increases. The concentration effect is much more severe when the increased dimensions are irrelevant to the underlying data structure, e.g., clusters. As shown in [44], in such cases, the standard deviation of the normalized vector lengths can decrease towards zero at about 10 dimensions. The severe concentration mixes outliers with inliers in the metric space, which makes it difficult for multivariate distance-based outlier detection methods to work well. Although many other state-of-the-art multivariate methods do not involve distances in its outlier scoring, their foundation is built on the meaningfulness of data distances. For example, although density-based methods use local densities as outlier scores, they rely on distance measures to find the local region of a given data object; Loda [33] uses the likelihood of falling into bins of one-dimensional random projection as outlier scores, but the objective of random projection is to retain the pairwise distances in the original space. As a result, although these non-distance-based methods may suffer less from the concentration effect compared to distance-based methods, they fail to have an accurate outlierness estimation. Since univariate detectors work on a feature-wise manner, they are not affected by this concentration problem. Another reason is due to the presence of noisy features, which is defined as follows.

Definition 5.2 (Noisy Feature). Let $\mathcal{X}_{.l}$ be the l-th feature of the data $\mathcal{X}$, ϕ be a multivariate outlier scoring function, and $0 \le k \le D-1$. $\mathcal{X}_{.l}$ is said to be a noisy feature w.r.t. ϕ if there exist some outliers $\mathbf{x}_i$ and inliers $\mathbf{x}'_i$ s.t. $\phi(x_{ij}, \ldots, x_{ij+k}) > \phi(x'_{ij}, \ldots, x'_{ij+k})$ but $\phi(x_{ij}, \ldots, x_{ij+k}, x_{il}) \le \phi(x'_{ij}, \ldots, x'_{ij+k}, x'_{il})$.

Such noisy features override the relevant features and render the outlier scoring functions less effective. When there exist some individually relevant features, univariate outlier detectors can well detect these outliers, while the multivariate detectors fail due to the presence of the noisy feature $\mathcal{X}_l$.

5.2 Modeling Heterogeneous Distributions on Subsamples

Outlier detectors that are capable of detecting outliers in different probability distributions are required to build an effective heterogeneous ensemble. Z-Score, Dixon, and kNN are chosen based on this principle. According to [6], Z-Score was designed to identify outliers that do not fit normal distributions well, and the Dixon test excels at detecting outliers that violate exponential, Gumbel, Frechet, or Weibull distributions. Since they are tailored for specific data distributions, the expected distribution of the outlier scores and the statistical significance levels of detecting the outliers can accordingly be obtained. A series of significance levels for reporting the upper (or lower) outliers of specific data distributions can be found in [6]. kNN is a data-dependent method that does not make any assumption on the distribution of inliers, which is used to complement Z-Score and Dixon in features with mixed distributions.

Formally building ensembles with highly biased heterogeneous models on subsamples helps achieve low variance and low bias of our models. Specifically, let $f(\cdot)$ be an unknown outlier scoring function that can provide the idea score of each data object in the dataset $\mathcal{X}$ and $g(\cdot;\theta)$ be

an outlier detection model that estimates the outlier scores with the parameter set θ, then according to [3], we have

$$\mathrm{E}[MSE] = \frac{1}{N}\sum_{i=1}^{N}\Big(f(\mathbf{x}_i) - g(\mathbf{x}_i, \mathcal{X}; \theta)\Big)^2 + \frac{1}{N}\mathrm{E}\Big[\Big(\mathrm{E}\big[g(\mathbf{x}_i, \mathcal{X}; \theta)\big] - g(\mathbf{x}_i, \mathcal{X}; \theta)\Big)^2\Big], \tag{12}$$

where $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - g(\mathbf{x}_i, \mathcal{X}; \theta))^2$ with y_i be the ideal outlier score of $\mathbf{x}_i$ yielded by an underlying specification of the function f. The two terms on the right-hand side of Equation (12) are respectively analogous to the well-known model bias and variance. In this work, we assume the real-world data contains highly heterogeneous distributions per feature, so the underlying true model of such datasets, i.e., the function f, presumed to be composed by a set of feature-wise heterogeneous g functions. ZDD uses a set of heterogeneous and complementary weak univariate outlier detectors to specify the function g so as to approximate the true model f as much as possible in the first term. This is expected to achieve lower bias than the models that ignore the feature-level heterogeneity. On the other hand, we aggregate models on bootstrapped subsamples that represent a collection of different realizations of the data $\mathcal{X}$, which helps reduce the variance of our ZDD model, i.e., the error in $(\mathrm{E}[g(\mathbf{x}_i, \mathcal{X}; \theta)] - g(\mathbf{x}_i, \mathcal{X}; \theta))^2$, due to the difference between the data subsets used for modeling. Note that the bias-variance analysis is built upon the setting where we have separate training and test data in supervised learning, which is different from our setting (i.e., training and evaluating unsupervised outlier detection models on the same dataset). Nevertheless, the generalized bias-variance analysis shown in Equation (12) provides some straightforward insights into the explanation of the possible errors made by our outlier ensemble.

5.3 Building Optimal Ensembles

As we use the exhaustive search to find the optimal $\mathcal{B}^*$ in Equation (8), $\mathcal{B}^*$ is guaranteed to be globally optimal. Then, the key to the optimization problem is the effectiveness of the objective function, Equation (7), which uses the margin of the outlier scores between the pseudo outliers in O and the pseudo inliers in $\mathcal{X} \setminus O$. Accordingly, the quality of the solution to the score margin maximization relies on the quality of the outlier candidate set O, i.e., Equation (7) is effective only when most, if not all, of the objects in O are truly outliers. Below, we show that the outlier thresholding strategy used in Equation (7) can well guarantee the quality of O.

Corollary 5.3 (False Positive Bound [30]). *Assume the scores in* $\mathbf{r}$ *have the expected value* μ *and variance* σ^2*. Then the outlier candidate set* O *resulted by the threshold* $\mu + \alpha\sigma$ *has a false positive upper bound of* $\frac{1}{1+\alpha^2}$.

We have $\mathrm{Prob}(r_i \geq \mu + b) \leq \frac{\sigma^2}{\sigma^2+b^2}$ per *Cantelli*'s inequality. By replacing $b = \alpha\sigma$, we obtain

$$\mathrm{Prob}(r_i \geq \mu + \alpha\sigma) \leq \frac{1}{1+\alpha^2}, \tag{13}$$

in which it is assumed that most outlier scores in $\mathbf{r}$ distribute around μ, with the probability of up to $\frac{1}{1+\alpha^2}$ that a few exceptions occur. Since large values in $\mathbf{r}$ indicate high outlierness, $\mu + \alpha\sigma$ can be used as a threshold to label data objects that have outlierness no less than the threshold as outliers. In other words, we have a probability of up to only $\frac{1}{1+\alpha^2}$ to falsely treat inliers that have large outlier scores as outliers. Also, *Cantelli*'s inequality makes no assumption on specific probability distributions, which holds for any distributions that have statistical mean and variance.

Table 1. AUC Performance of ZDD, Its Five Variants, and Four Competing Methods on 20 Datasets

	Basic Data Characteristics			Multivariate Ensembles				Univariate Ensembles: ZDD and Its Variants					
Data	*N*	*D*	Outliers (%)	iForest	LeSiNN	Loda	EGMM	ZDD	ZDD-fc	HOMZ	Z-Score	Dixon	*k*NN
http	567497	3	0.39%	0.9998	**1.0000**	0.9958	**1.0000**	0.9923	0.9983	0.9840	0.9988	0.9980	0.9980
census	299285	7	6.20%	0.6633	0.7160	0.6863	0.7334	0.7426	0.7667	0.5500	0.7051	0.6991	**0.8048**
FC	286048	10	0.96%	0.8733	0.8966	0.9066	0.9217	0.9304	0.9309	0.9351	0.9326	**0.9452**	0.8881
fraud	284807	29	0.17%	0.9510	**0.9531**	0.9482	0.9504	0.9452	0.9526	0.9454	0.9525	0.9536	0.9485
mulcross	262144	4	10.00%	0.9581	0.9994	0.6993	0.7415	0.9744	0.9790	0.9993	0.9987	0.7157	**0.9995**
celeba	202599	39	2.24%	0.6797	0.7594	0.7487	0.7000	0.8104	0.7893	**0.8108**	0.7814	0.7834	0.7876
breast	181903	13	3.45%	0.7630	0.8223	0.7768	0.8409	0.8677	0.8657	**0.8726**	0.8619	0.8322	0.8678
smtp	95156	3	0.03%	0.8825	0.8326	0.8344	0.7012	**0.9469**	0.9004	0.7947	0.7575	0.9072	0.8855
probe	64759	34	6.58%	0.9952	**0.9974**	0.9549	0.9086	0.9883	0.9887	0.9883	0.9928	0.9793	0.9877
u2r	60821	36	2.97%	0.9881	0.9877	**0.9895**	0.9855	0.9884	0.9891	0.9884	0.9793	0.9882	0.9887
w7a	49749	300	5.39%	0.4053	0.4851	0.4679	0.6750	0.8058	0.7947	0.8045	0.5081	**0.8259**	0.8028
shuttle	49097	9	7.15%	**0.9966**	0.9903	0.9784	0.9808	0.9764	0.9953	0.9924	0.9899	0.9899	0.9957
bank	41188	62	11.27%	0.7110	0.6879	0.6688	0.7412	0.7584	0.7376	0.7519	0.7422	**0.8031**	0.7085
MG	11183	6	2.33%	0.8505	0.8253	0.8183	0.8290	0.8661	0.8772	0.8161	**0.8847**	0.8666	0.8687
mc1	9466	38	0.72%	0.9051	0.8936	0.8742	0.8882	0.9151	0.9093	**0.9202**	0.9154	0.9060	0.9076
thyroid	7200	6	7.42%	0.8364	0.6578	0.6135	0.6294	**0.9040**	0.7644	0.7317	0.6617	0.7847	0.8221
PB	5473	10	6.43%	0.8971	0.8866	0.8883	0.8856	0.9084	0.8808	**0.9098**	0.9048	0.8537	0.8846
hiva	4229	1617	7.71%	0.6809	0.6843	0.6743	0.5000	0.7286	0.7257	0.7279	0.6912	**0.7421**	0.7297
isolet	730	617	1.37%	**1.0000**	**1.0000**	0.9997	0.4108	0.9994	0.9995	0.9995	0.9998	0.9996	0.9937
mfeat	410	649	2.44%	0.9462	**0.9742**	0.8377	NaN	0.9535	0.9513	0.9615	0.9321	0.9504	0.9177
			Average	0.8556	0.8611	0.8261	0.7903	**0.9053**	0.8943	0.8815	0.8631	0.8812	0.8938
			P-value	0.0072	0.0333	0.0017	0.0005	-	0.2322	0.6791	0.0400	0.2110	0.1305

EGMM cannot obtain the results on mfeat due to its algorithmic constraints. The following acronyms are used: FC = ForestCover, MG = mammography, thyroid = annthyroid, PB = PageBlocks. The best performance per data is boldfaced.

6 EXPERIMENTS AND EVALUATION

6.1 Datasets

As shown in Table 1, 20 publicly available datasets[2], including 19 real-world datasets and 1 synthetic dataset (i.e., *mulcross*), are used in our evaluation, which cover a broad range of application domains, e.g., intrusion detection, credit card fraud detection, disease detection, molecular bioactivity detection, and imaging object detection. About half of the real-world datasets, including *http*, *fraud*, *smtp*, *probe*, *u2r*, *thyroid*, and *hiva*, contain real outliers. Following the literature (e.g., [26, 27, 31, 33, 34, 37, 38]), the other datasets are transformed from very imbalanced classification datasets by treating the rare class(es) as outliers and the largest class as the normal class, which results in data with semantically real outliers. The synthetic data *mulcross* contains two large dense Gaussian clusters as the normal classes and two small clusters as clustered outliers.

[2] *fraud*, *celeba*, *breast*, and *w7a* datasets are respectively available at https://www.kaggle.com/mlg-ulb/creditcardfraud, http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html, http://www.bcsc-research.org/rfdataset/dataset.html, and https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. *mulcross* is taken from [27]. All the other dataset are from the UCI Machine Learning Repository [15].

6.2 Default Settings of ZDD

The subsampling size s and the ensemble size t are respectively set to 30 and 10 by default in ZDD. ZDD is built by ensembles of six base learners, i.e., ZDD use two specifications (i.e., $c = 2$) of each of the three outlier scoring methods presented in Section 4.1. Specifically, Z-Score with two sets of μ and σ computed on two different random subsamples are used. Since the subsamples are different, we obtain two different sets of μ and σ for the same feature. Similarly, the Dixon score has different ranges and neighborhood on two different subsamples, which also lead to heterogeneous Dixon-based detectors. For the kNN detector, both $k = 10$ and $k = s$ are used to obtain heterogeneous kNN detectors. Other specifications of the three base learners may also be applicable, but our empirical results show this set of specifications performs most stably across the 20 datasets. Thus, these settings are used throughout the experiments by default. The source code of ZDD is available at https://sites.google.com/site/gspangsite/sourcecode.

6.3 Performance Evaluation Methods

6.3.1 Performance Metric. Following the literature [27, 30, 33, 34, 37, 38], the AUC is used to evaluate the performance of outlier detection. Specifically, all outlier detectors first produce a ranking of data objects based on their outlier scores. The AUC is then calculated based on the outlier ranking. AUC has been widely used in outlier detection. One main reason of its popularity is due to its straightforward interpretability. That is, an AUC value of 0.5 indicates a random ranking of the objects while an AUC value of one indicates perfect performance; having an AUC above (below) 0.5 indicates that the performance is better (worse) than random results. Since ensemble methods involve randomness, we report average AUC results over 10 independent runs.

6.3.2 Significance Test. Two different statistical significance tests, namely the *Wilcoxon* signed rank test and *Friedman-Nemenyi* test, are used to have a good summarized description of our results of multiple detectors on a large set of datasets. The *Wilcoxon* signed rank test is a pairwise approach, which is used to examine the significance of the AUC performance of ZDD against individual competitors. The *Friedman-Nemenyi* test is for comparing multiple participated detectors simultaneously. We refer readers to [13] for detailed introduction of these two tests.

6.4 Effectiveness in Real-world Multidimensional Data

6.4.1 Experiment Settings. ZDD is compared with four state-of-the-art multivariate outlier ensembles to verify their effectiveness in multidimensional data, including two homogeneous ensembles, iForest and LeSiNN, and two heterogeneous ensembles, Loda and EGMM.

- Isolation-based Ensemble: iForest [27]. iForest defines outliers by the number of partitions to isolate data objects. Following [27], the ensemble size is set to 100 and the subsampling size is set to 256.
- Distance-based Ensemble: Ensembles of distance-based methods (e.g., Sp [37] and LeSiNN [31]) use the nearest neighbor distances in small subsamples as outlier scores. Since LeSiNN performs substantially better and more stably than Sp as shown in our previous studies [31], LeSiNN is used. The subsampling size is set to 8 and 50 random subsamples are used as these settings often enable the best performance of LeSiNN.
- One-dimensional random histograms: Loda [33]. Loda defines outlierness via the log-likelihood computed on a set of one-dimensional optimized histograms produced by using random projection with different random Gaussian projection vectors. Loda is parameter-free.
- Ensembles of Gaussian Mixture Models: EGMM [2]. EGMM defines outliers based on the probability densities of fitting GMMs. Similar to ZDD, 10 base models are used to construct

the EGMM ensemble. Following [16], 15 bootstrap replicates are used to train a single GMM. Following [7], we use *Akaike*'s information criterion (AIC) to search the optimal number of components in the range of [1, 6].

Note that the above methods may use different subsampling and/or ensemble sizes. In general, most methods are not sensitive to the ensemble size, which perform very stably using 10–100 base models to build the ensemble. iForest and LeSiNN are sensitive to the subsampling size. We use the recommended subsampling sizes for iForest and LeSiNN as in [27, 31]. All methods are implemented in MATLAB except iForest that is in Java in Weka [20].

It should also be noted that this work focuses on comparing univariate ensembles and multivariate ensembles. The competing multivariate ensembles include state-of-the-art specifically designed outlier ensemble method, iForest, and the methods that are substantially enhanced ensemble versions of traditional single-model outlier detectors, i.e., LeSiNN, Loda, and EGMM.

6.4.2 Finding - ZDD Substantially Outperforms State-of-the-art Multivariate Methods. The AUC performance of ZDD and four multivariate ensembles on the 20 datasets is shown in Table 1 (the five variants of ZDD and their performance will be discussed in Section 6.5). Compared to the competing multivariate ensembles iForest, LeSiNN, Loda, and EGMM, ZDD obtains the best AUC performance on 12 datasets, with 5 close to the best competitor (having difference in AUC less than 0.01). On average, ZDD substantially outperforms iForest (6%), LeSiNN (5%), Loda (9%), and EGMM (14%). The pairwise *Wilcoxon* signed rank test shows the improvement of ZDD over these four competing methods is statistically significant at the 95% confidence level. The superiority of ZDD is due to two main reasons. First, many real-world datasets contain univariate outliers, and thus, univariate methods are sufficient to detect such outliers. Second, there often exist heterogeneous data distributions within each individual feature. In such cases, ZDD models the fine-grained feature-level heterogeneities much better than multivariate ensembles do, resulting in better performance in identifying data distribution-sensitive outliers.

6.5 Ablation Study

6.5.1 Experiment Settings. The success of ZDD motivates us to examine the performance of their individual components to answer the following four key questions:

- **Why do we use univariate ensembles other than multivariate ensembles?** We investigate this question by examining the performance of simple bagging ensembles of each individual univariate detector against the above four advanced multivariate outlier ensembles. Similar to ZDD, the final outlier scores of data objects are based on the bootstrap aggregation of 10 random subsamples, and the subsampling size is set to 30.
- **What is the benefit of building univariate ensembles on subsamples other than original full data?** To answer this question, we examine the performance of the three univariate ensembles on the subsamples compared to that on the original full data.
- **What is the benefit of using heterogeneous univariate base learners other than the homogeneous ones?** We investigate this question by comparing ZDD to its variants that are more homogeneous.
- **What is the benefit of the selective combination of $|C|$ detectors?** We answer this question by comparing the performance of ZDD to its variant that uses a full combination of all $|C|$ detectors.

To enable the analysis of the above four questions, the following five variants of ZDD are implemented and used as baselines.

- Z-Score, Dixon, kNN are the simple bagging ensembles of the three univariate outlier measures Z-Score, Dixon, and kNN, respectively. kNN is used with $k = s$ for this case. The mean and variance in Z-Score are computed based on each univariate subsample. Their subsampling size and the number of subsamples are set to the same values as that in ZDD.
- HOMZ (short for HeterOgeneous enseMbles of Z-Score) is a simplified heterogeneous ensemble of ZDD. HOMZ is exactly the same as ZDD except that HOMZ uses only the Z-Score measure rather than all the three detectors in $\mathcal{A}$ to specify the base detectors. Particularly, HOMZ is composed by six independent Z-Score-based outlier detectors per univariate subsample. That is, for each specific univariate subsample $\mathcal{S}_{ij}$, we randomly sample six versions of $\mathcal{S}_{ij}$, and compute the mean and standard deviation of each $\mathcal{S}_{ij}$ independently, resulting in six different Z-Score-based outlier detectors on each univariate subsample.
- ZDD-fc is a variant of ZDD, which is exactly the same as ZDD except that ZDD-fc uses a Full Combination of all $|C|$ detectors while ZDD uses the selective optimal combination of the detectors.

6.5.2 Findings I - Univariate Ensembles Improve 4%–13% Performance over Advanced Multivariate Ensembles. The performance of the bagging ensembles of Z-Score, Dixon, and kNN is shown in Table 1. Interestingly, the simple Z-Score, Dixon, and kNN ensembles substantially outperform the state-of-the-art multivariate methods iForest, LeSiNN, Loda, and EGMM on most datasets, and obtain comparable performance on the datasets where the multivariate methods perform better. On average, the AUC improvement of univariate ensembles over the four multivariate ensembles is up to 4%–13%. In addition to the fact that there exist strongly relevant features for identifying univariate outliers, another major reason for this result is that the datasets may contain a large percentage of irrelevant features that largely bias the outlier scoring of the multivariate outlier detectors. Although univariate methods may also be affected by the irrelevant features in the final aggregation of outlier scores obtained from each feature, they can accurately compute the outlier scores in relevant features. Therefore, univariate methods can obtain higher-quality outlier scores than multivariate methods. More results about the robustness w.r.t. irrelevant features are discussed in Section 6.6.

6.5.3 Findings II - The Subsampling Significantly Improves Univariate Outlier Detectors. We further compare the Z-Score, Dixon, kNN ensembles with their corresponding single models that work on the original data, denoted as Z-Score′, Dixon′, and kNN′, respectively. The comparison is summarized as in the *Friedman-Nemenyi* test results in Figure 2. It shows that the bagging ensembles, kNN and Dixon, obtain significantly better performance than their single models, kNN′ and Dixon′; and Z-Score performs the same well as Z-Score′. This demonstrates the effectiveness of using bagging ensembles to address the swamping and masking problems in univariate outlier detectors, which also justifies the value of the subsampling component in ZDD.

6.5.4 Findings III - Combination of Heterogeneous Univariate Detectors Achieves More Stable and Better Performance. This section compares ZDD with three homogeneous univariate ensembles, namely, Z-Score, Dixon, and kNN, and the simplified heterogeneous ensemble HOMZ to show the importance of building heterogeneous ensembles.

The last column in Table 1 shows that ZDD and HOMZ achieve nearly the same performance on most datasets, but HOMZ performs unstably and obtains poor performance on some datasets, such as *census* and *thyroid*, whereas ZDD performs very stably. As a result, ZDD obtains over 3% average AUC improvement over HOMZ. This is mainly because HOMZ only uses Z-Score to define outliers, which fail to work on data distributions other than normal distributions, while ZDD cover

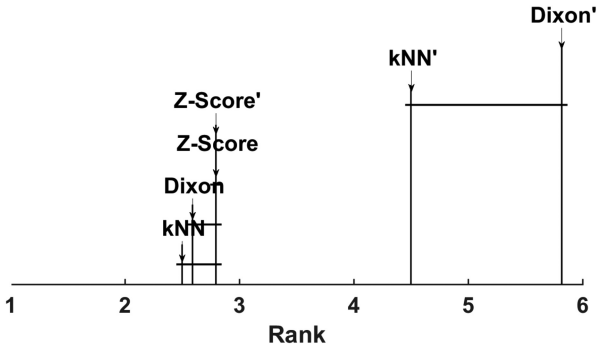

Fig. 2. *Friedman-Nemenyi* test results for ensembles Z-Score, Dixon, *k*NN, and their corresponding single models Z-Score′, Dixon′, *k*NN′ over 20 datasets. Each outlier detector is represented by a vertical line. There is no significant difference between the performance of outlier detectors if their corresponding vertical lines are intersected by a horizontal line on the top; and otherwise the performance difference is significant at the 95% confidence level.

outliers in a broader distribution family. Similar results are expected if only Dixon or *k*NN is used to specify the simplified heterogeneous ensemble.

Similar to HOMZ, since the Z-Score, Dixon, and *k*NN ensembles consist of only one type of base detectors, they perform unstably and fail to obtain good performance on such datasets as *mulcross*, *celeba*, *smtp*, and *thyroid*. Different from these three univariate ensembles, ZDD uses the *Cantelli*'s inequality-based outlier ranking quality measure to effectively find an optimal combination of the outlier scores from multiple different types of outlier detectors, resulting in better and more stable performance. This set of results demonstrates the importance of using different types of base detectors in building heterogeneous ensembles.

6.5.5 Findings IV - Selective Combination of Univariate Detectors Yields Substantial Improvement in Complex Data. This section compares ZDD with ZDD-fc to investigate the contribution of the module of selectively combining the heterogeneous base detectors in ZDD. The results in the last column in Table 1 demonstrates that ZDD achieves averagely better AUC performance than ZDD-fc. In many datasets, ZDD and ZDD-fc performs comparably well, but ZDD significantly outperforms ZDD-fc in datasets with highly heterogeneous data distributions across the features, such as *celeba*, *smtp*, and *thyroid*, in which each feature may demand a different combination of the base detectors to work well. In those data, the outstanding base detectors in ZDD-fc may be severely dragged down by under-performed base detectors due to the simple average aggregation, whereas ZDD can avoid this issue by its selective optimal combination component. This observation becomes clearer in our experiments on synthetic data with irrelevant features in Section 6.6.

6.6 Robustness w.r.t. Irrelevant Features

6.6.1 Experiment Settings. We use a similar method as in [44] to generate a collection of 100-dimensional synthetic datasets with different percentages of relevant features (or irrelevant features). In this data, inliers are drawn from a Gaussian distribution, in which outliers are set at two standard deviations of the distribution in relevant features and the other features are from uniform distribution and used as irrelevant features. Each dataset contains 10,000 data objects with 2% outliers. The average AUC with ±1 standard deviation over 10 runs is reported to have more reliable and straightforward comparisons.

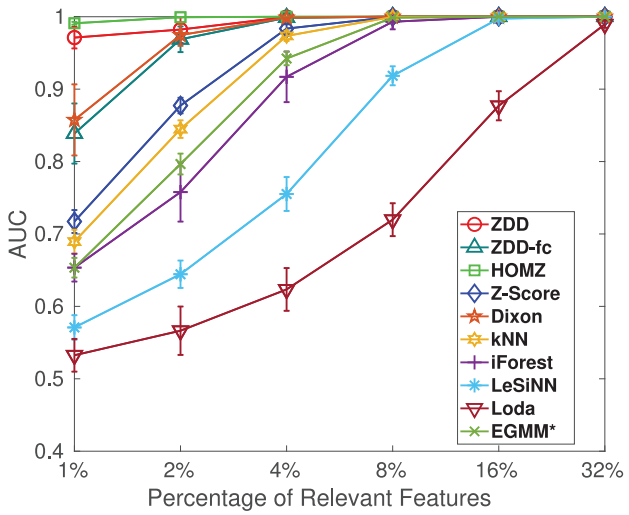

Fig. 3. AUC performance w.r.t. different percentage of relevant features. All detectors perform stably and achieve the AUC of one with over 32% relevant features, except Loda whose AUC performance increases slower than the other detectors.

Because the AIC search in EGMM performs poorly with irrelevant features, we use EGMM* that sets the number of optimal components to be one, i.e., the ground truth. All the other detectors are used with the default settings.

6.6.2 Findings I - ZDD Achieves Excellent Robustness w.r.t. Irrelevant Features. The AUC performance of all 10 detectors w.r.t. different percentage of irrelevant features is illustrated in Figure 3. ZDD obtains excellent robustness w.r.t. the irrelevant features, which can obtain an AUC of nearly one even when there are only 1% relevant features, i.e., only one relevant feature in 100-dimensional data, and they perform very stably with increasing number of relevant features. HOMZ outperforms ZDD and becomes the best performer among all the detectors. This is because the synthetic data is strictly drawn from Gaussian distributions and HOMZ excels at identifying outliers from Gaussian distributions. Although ZDD is slightly dragged down by the base detectors other than Z-Score, it is interesting that the individual Z-Score, Dixon, and kNN perform badly while ZDD, which is a mixture of Z-Score, Dixon, and kNN, can achieve performance very comparable to the best performer HOMZ. This is mainly because the *Cantelli*'s inequality-based outlier ranking measure enables ZDD to effectively filter out irrelevant outlier rankings to produce high-quality ranking combination. This explanation can also apply to the AUC difference between HOMZ and the other two detectors, Z-Score and EGMM*.

6.6.3 Findings II - Univariate Ensembles Obtain Consistently Better Robustness Than Multivariate Ensembles. As shown in Figure 3, it is clear that the six univariate ensembles, including ZDD, ZDD-fc, HOMZ, Z-Score, Dixon, and kNN, consistently outperform the four multivariate ensembles, iForest, LeSiNN, Loda, and EGMM. This result can be explained by the bias and dimensionality curse brought by the irrelevant features to the multivariate ensembles. Since univariate methods work on an individual feature basis, they are less affected by the irrelevant features. By working on feature subspaces, iForest reduces the effects of irrelevant features and accordingly obtains better robustness than the full space-based methods like LeSiNN and Loda. Although EGMM* is specifically designed to detect outliers in a single Gaussian distribution, but it cannot find the

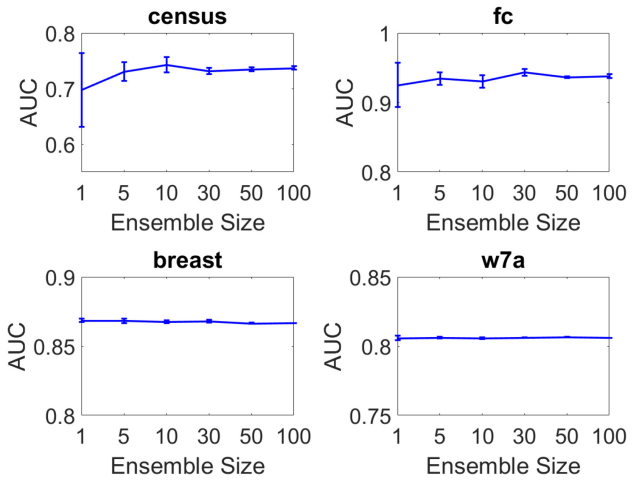

Fig. 4. AUC Performance w.r.t. different ensemble sizes. Only representative results are shown. Similar results are obtained in other data. Vertical bars indicate ±1 standard deviation.

Gaussian cluster exactly due to the distortion of the large percentage of irrelevant features and consequently can only achieve fairly good performance.

6.7 Sensitivity Test

6.7.1 Experiment Settings. This section examines the sensitivity of ZDD w.r.t. the ensemble size, t, and the subsampling size, s. $t \in \{1, 5, 10, 30, 50, 100\}$ and $s \in \{15, 30, 60, 120, 240, 480\}$ are used. ZDD was tested on the 20 datasets. We report representative results on four datasets for brevity. Similar trends are found on the rest of other datasets. The runtimes of all detectors are calculated at a node in a 2.8 GHz Titan cluster with 256 GB memory.

6.7.2 Findings I - ZDD Performs More Stably with Increasing Ensemble Size. The AUC performance of ZDD w.r.t. different ensemble size is presented in Figure 4. The average AUC performance of ZDD flattens from $t = 10$, and the standard deviation becomes smaller as the ensemble size t increases. Similar results can also be observed in the other datasets. A sufficiently large ensemble size (e.g., ≥ 10) is generally required for bagging approach-based ensembles like ZDD to obtain statistically sound performance. The performance of these ensembles then becomes very stable w.r.t. increasing ensemble size due to the law of large numbers [12]. This is consistent to the results in the prior work [27, 31, 38]. Additionally, ZDD can perform very well and stably on data with simple distributions as *breast* and *w7a* even when using $t = 1$, while it requires a slightly large ensemble size to achieve similar performance in complex data such as *census* and *fc*. This is because outliers are much more difficult to be distinguished from the inliers in random subsamples taken from complex data than simple data.

6.7.3 Findings II - ZDD Obtains Consistently Good Performance Using Small Subsampling Size. The AUC performance of ZDD w.r.t. different subsampling size is presented in Figure 5. ZDD achieves consistently good performance when using a small subsampling size. Using larger subsampling size has a higher probability of including outliers into subsamples and consequently masking outliers or biasing the outlier scoring functions, leading to worse AUC performance, such as the case in *census*. In some other cases as in *fc*, a larger subsampling size is required to have a better approximation of the inliers. In simple datasets such as *breast* and *w7a*, ZDD performs very stably when using a wide range of subsampling size.

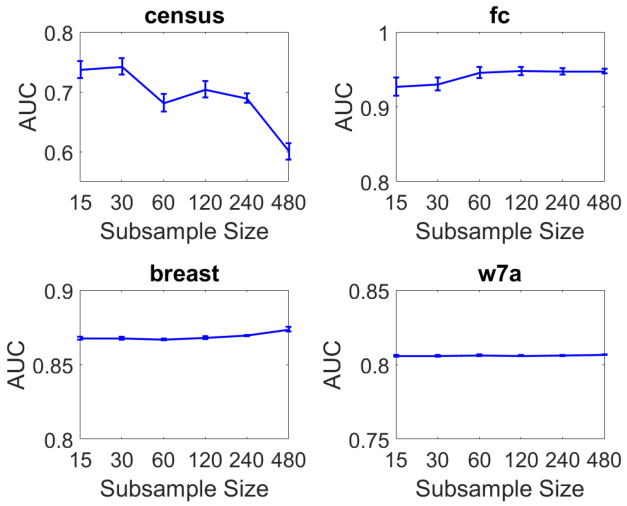

Fig. 5. AUC performance w.r.t. different subsample sizes. Representative results are shown. Similar results are obtained in other data. Vertical bars indicate ±1 standard deviation.

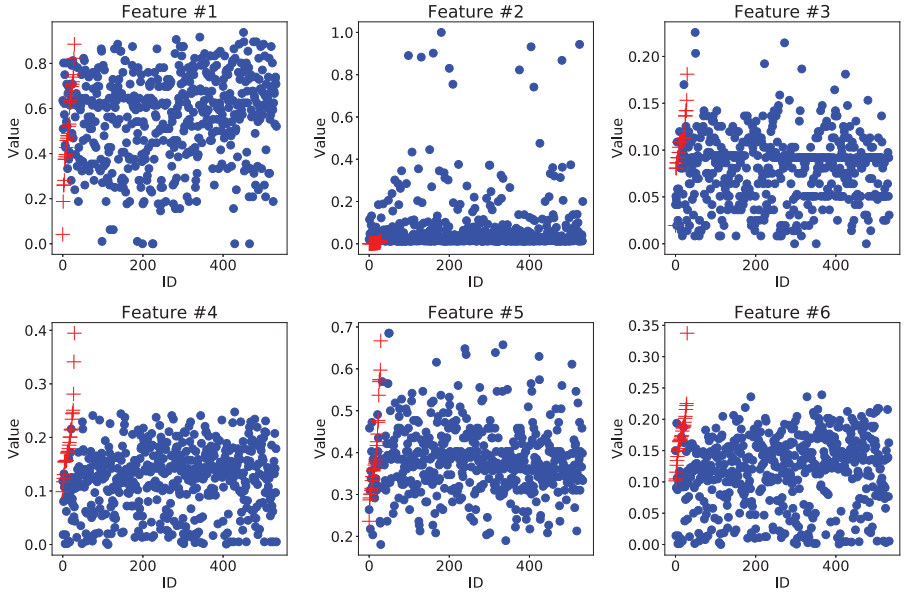

Fig. 6. Distribution of the subsamples and all outliers in each individual feature of *thyroid*.

6.8 Further Analysis of Our Results by Visualizing Underlying Outlying Behaviors

To understand the underlying outlying behaviors and explain the results we obtain, this section presents the visualization of all outliers (blue circles) and the randomly sampled 30 data objects (red crosses), i.e., the subsample used in our ensemble, in each individual feature from four representative datasets, including *thyroid* in Figure 6, *smtp* in Figure 7, *census* in Figure 8, and *PB* in Figures 9 and 10. Here we focus on low-dimensional datasets to well visualize the data. Note that all features

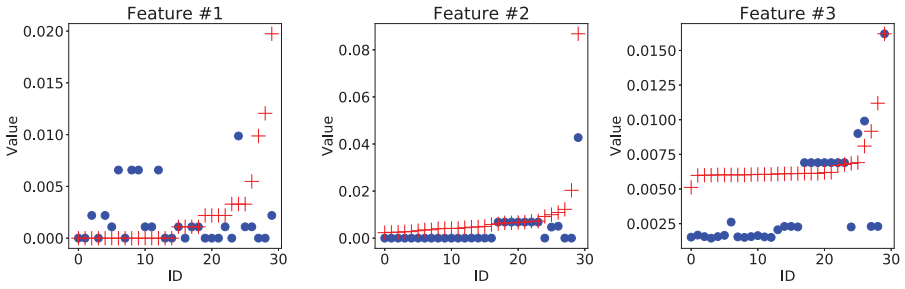

Fig. 7. Distribution of the subsamples and all outliers in each individual feature of *smtp*.

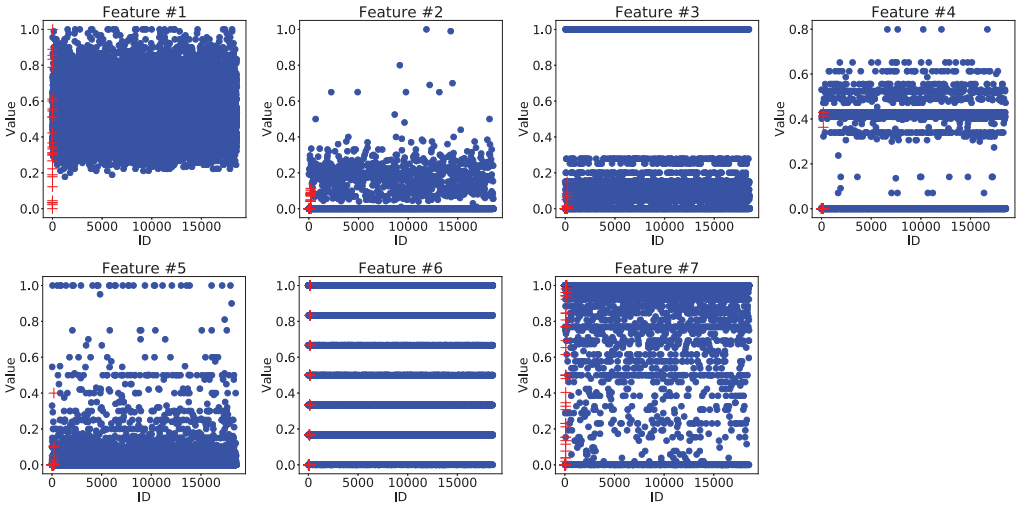

Fig. 8. Distribution of the subsamples and all outliers in each individual feature of *census*.

are normalized into the range $[0, 1]$ before applying the outlier detectors, but some features in the figures have rather narrow value ranges because those features have highly skewed distributions.

The results on *thyroid*, *smtp*, and *census* are provided in Figures 6–8 to gain possible understanding of why the univariate methods can significantly outperform the multivariate methods in Table 1. It is clear that in one or multiple individual features many (or most) of the outliers have significant deviations from the subsamples that represent the majority of the data objects, e.g., Feature #2 in *thyroid*, Feature #3 in *smtp*, Features #2, #3, and #5 in *census*. This means these datasets contain many univariate outliers that are highly separable in this view and can be easily detected by the univariate methods. On the other hand, these datasets also contain a large percentage of noisy features in which outliers have no clear deviations from the data. These noisy features can largely mislead the multivariate methods, making them ineffective in detecting the outliers that are highly visible in the univariate views.

Among these three datasets, the heterogeneous univariate method ZDD substantially outperforms its homogeneous or less heterogeneous variants on *thyroid* and *smtp*. This is because the heterogeneous base detectors in ZDD are complementary to each other and are able to detect different sets of outliers in different features, whereas the homogeneous base detectors in the variants of ZDD suffer from the inherent weakness of the employed base detectors on less relevant features and yield poor overall performance. For example, on *thyroid*, using Z-Score only works very well

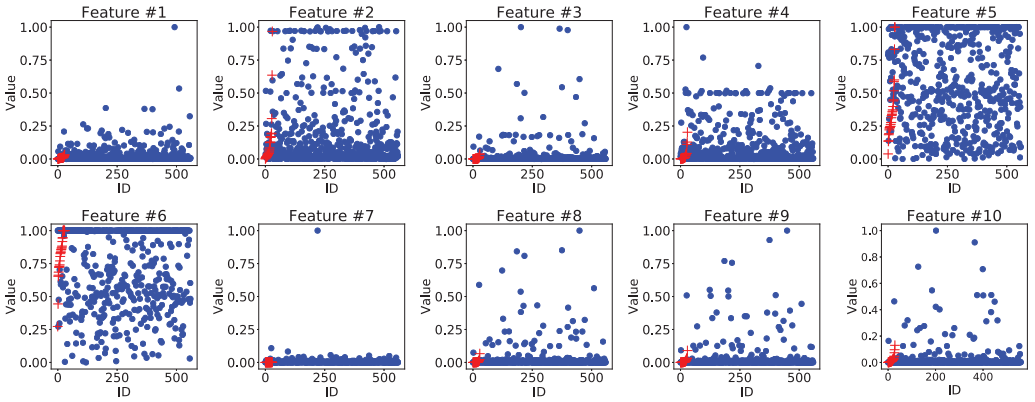

Fig. 9. Distribution of 30 randomly sampled objects and all outliers in each individual feature of *PB*.

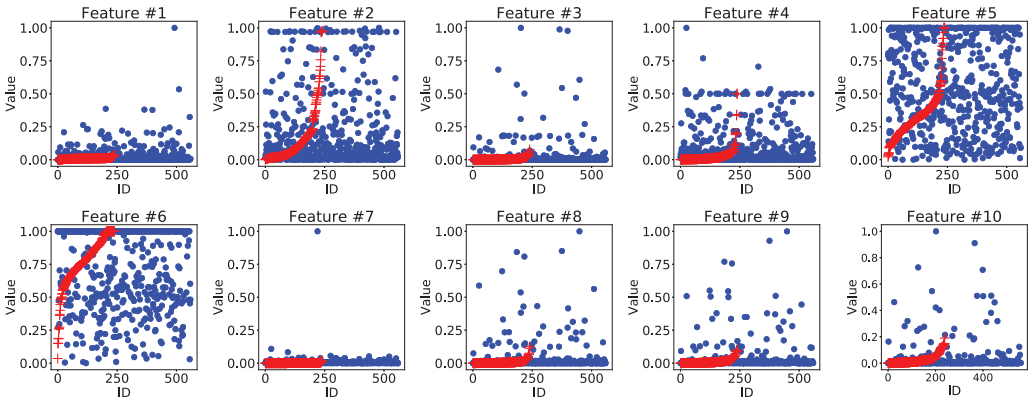

Fig. 10. Distribution of 240 randomly sampled objects and all outliers in each individual feature of *PB*.

on Feature #2 but it can seriously mess up on the rest of the other five features; by contrast, ZDD has base detectors that work well not only on Feature #2 (e.g., using any of its base detectors) but also on Features #3, #4, and #6 (e.g., using Dixon and kNN). On the other hand, homogeneous univariate methods like kNN can outperform the heterogeneous method ZDD, e.g., on *census* in Table 1. This is mainly because Z-Score and/or Dixon are ineffective in some relevant features, e.g., Features #4 and #5, and their performance can drag down the effectiveness of kNN in those features when they are used as an integrated unit in ZDD.

Note that the multivariate methods perform better than the univariate methods on a few datasets such as *http*, *fraud*, *isolet*, and *mfeat*, though the univariate methods can obtain very competitive performance on most of these datasets. This indicates that these datasets may contain a mixture of univariate and multivariate outliers, with the majority of outliers being univariate outliers. The univariate methods cannot detect the small number of multivariate outliers and are thus slightly less effective than the multivariate methods.

It is interesting that the performance of ZDD is very stable w.r.t. its two hyperparameters, especially the subsampling size on datasets like *breast* and *w7a* shown in Figure 5. These two datasets are used as representatives only. Similar results can also be observed in several other datasets such as *PB* on which ZDD achieves very stable AUC results that are consistently within $[0.886, 0.889]$ with the subsampling size in $[15, 480]$ and the ensemble size in $[1, 100]$. To understand why we

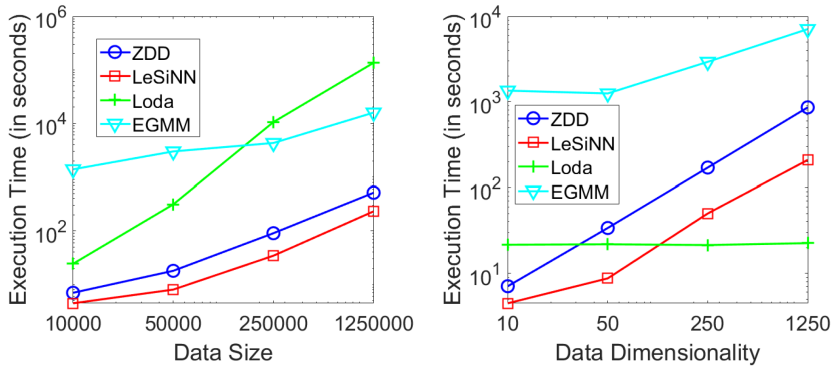

Fig. 11. Scaleup tests w.r.t. data size and dimensionality. iForest is excluded since it is implemented in a programming language different from the others.

have such results, we visualize and analyze the outliers and subsamples of the low-dimensional dataset *PB* in Figures 9 and 10 with two diverse subsampling sizes, 30 and 240. It can be seen that the majority of the outliers clearly deviate from the subsamples on all individual features except Features #2, #5 and #6 when using the subsampling size 30 in Figure 9; this desired pattern persists with a much larger subsampling size, 240 in Figure 10. As a result, those clearly deviated outliers can always be detected with different subsampling sizes, resulting in stable AUC performance. Also, due to the strong outlying signals observed in those relevant features, even using a small ensemble size, ZDD can achieve the similarly good performance to that using a large ensemble size.

6.9 Scalability Test

6.9.1 Experiment Settings. We generate synthetic data by varying the data size in {10,000, 50,000, 250,000, 1250,000} of a 10-dimensional dataset for scaleup test w.r.t. data size, and likewise, varying the dimension in {10, 50, 250, 1, 250} w.r.t. a fixed data size (i.e., 10,000) for scaleup test w.r.t. dimensions. We test the scalability of ZDD, with multivariate ensembles as baselines.

6.9.2 Findings - ZDD Scales Up Well w.r.t. Both Data Size and Dimensionality. The scalability test results are presented in Figure 11. In the left panel, ZDD, LeSiNN, and EGMM have a linear time complexity w.r.t. data size, while that of Loda seems to be quadratic. Specifically, ZDD completes the outlier scoring in a dataset of size 1,250,000 within 520 seconds, i.e., less than one millisecond per data object, which runs comparably fast to LeSiNN and is one or two orders of magnitude faster than Loda and EGMM. In the right panel, ZDD has a quadratic time complexity w.r.t. the number of features. This is expected since it uses the pairwise correlations between the outlier rankings output from individual features to capture the homophily relations among outlying behaviors. As a result, ZDD runs slower than LeSiNN by a factor of 5, but it runs about 10 times faster than EGMM. Since Loda involves only such simple operations as random matrix generation and element-wise matrix multiplications, it is not sensitive to the dimensionality size, resulting in the best efficiency on high-dimensional data.

7 DISCUSSION

7.1 Univariate or Multivariate Methods?

Our empirical results suggest that many real-world datasets mainly contain univariate outliers, in which univariate outlier detection methods are sufficient to well identify these outliers. This leads

to an interesting question: When should we use univariate methods (or multivariate methods) in real-life applications? It can be seen as the problem of defining an indicator function to determine whether a given dataset contains univariate outliers or multivariate outliers. However, this problem may be more challenging than outlier detection itself due to the unavailability of class labels. One possible approach is to use the correlation across features as the indicator. We have attempted this approach, but the feature correlation did not show useful hints for the selection of univariate methods or multivariate methods. This is because the full dataset is dominated by inliers, and thus, the feature correlation on the full dataset indicates the feature interdependence for inliers rather than outliers. Consequently, this problem gets into a dilemma: identification of (univariate or multivariate) outliers and the feature correlation for outliers. On the other hand, a given multidimensional dataset may contain both univariate outliers and multivariate outliers. Since multivariate methods are ineffective in identifying univariate outliers, a safer strategy for real-world deployments is to synthesize both univariate and multivariate outlier detection methods via, e.g., ensemble learning, to complement each other.

7.2 Extensions of HUOE

The capacity of our HUOE framework may be further extended w.r.t. the following two main strategies: (i) using more univariate outlier detectors that have different assumptions from Z-Score, Dixon test, and kNN; and (ii) increasing the number of base outlier detectors by using Z-Score, Dixon test, and kNN with different parameter settings. For the first strategy, as shown in Section 6.5.4, the use of different types of outlier detectors in C helps substantially improve the effectiveness and stability of the instances of HUOE. For the second strategy, we have empirically compared the use of three, six and nine base outlier detectors in ZDD. Our results show that the use of six base outlier detectors has substantial AUC improvement over the case of using three bases, e.g., achieving about 7% AUC improvement on *census*, but using nine base outlier detectors does not gain extra AUC improvement. On the other hand, increasing the number of base detectors can largely increase the runtime, since we have an exhaustive search over C to find the globally best combination. Therefore, it is suggested to increase the capacity of the instantiation by including other types of base detectors rather than increasing the base number with different parameter settings.

Additionally, HUOE may also be extended to handle multivariate outliers by changing its data inputs. For example, the original data input may be replaced with the projected data resulted from unsupervised data projection methods like PCA, random projections and many of their variants. In this case, although HUOE works in a feature-wise manner, it works on the resulting latent features that capture the interactions between multiple features in the original data space. As a result, HUOE is also capable to identify multivariate outliers. However, it is challenging to find the latent features that are highly relevant to outlier detection, since the dataset is dominated by inliers and the outliers may be irregularly distributed.

7.3 Homophily Weight vs. Time Cost

HUOE defines the homophily weights of the outlier rankings to obtain a weighted aggregation at its final stage. The homophily weights are important when the relevant feature are weak, i.e., the outlier scores of outliers in weakly relevant features are only marginally higher than that of inliers in irrelevant features. In such cases, the homophily weights leverage the weak relevance across the features to derive large weights for the weakly relevant features, which can substantially enlarge the outlierness gap between outliers and inliers. For example, in the *thyroid* dataset, the homophily weights help ZDD significantly improve its AUC performance, i.e., lifting from 0.7632 to 0.9040. On the other hand, the homophily weight-based aggregation achieves almost the same

performance as the general average aggregation in the other datasets, but it is more computationally costly due to the pairwise correlation computation. Hence, the homophily weights might be removed when handling very high-dimensional data, which may not affect the detection accuracy but substantially reduce the computational cost.

7.4 Multivariate Outlier Detection Benchmark Data

One major concern here is that most of datasets used in our experiments are widely used in the literature, e.g., [8, 27, 33, 37, 38], in which multivariate outlier detection methods are proposed and evaluated, but our results indicate that most, if not all, of the outliers in these datasets are univariate outliers. Consequently, some important questions are, *what types of outliers do these multivariate methods detect, univariate or multivariate outliers?* and *do we really evaluate the performance of these methods in detecting multivariate outliers?* Although applying multivariate methods to detect univariate outliers in multidimensional data helps evaluate their robustness w.r.t. irrelevant features, there may be serious mismatched specifications between our ideal objective of evaluating the detection of multivariate outliers and the real settings of those datasets. We therefore suggest a careful consideration of choosing these widely-used benchmark datasets when we intend to evaluate the capability of outlier detection methods in detecting specific types of outliers. It is also important to develop real-world datasets that mainly contain multivariate outliers to well evaluate newly proposed multivariate outlier detection methods.

8 CONCLUSION

In this article, we introduce a novel framework and its instantiation for building outlier ensembles to detect univariate outliers in multidimensional data with feature heterogeneities. By leveraging heterogeneous outlier detectors with a *Cantelli*'s inequality-based outlier ranking quality measure, we build optimized heterogeneous ensembles for each feature, which enables an effective detection of outliers in features with heterogeneous probability distributions. This is justified by their substantial AUC improvement over state-of-the-art multivariate methods. We empirically justify the necessity and importance of each individual component of the framework. Additionally, we show that simple ensembles of univariate outlier detection methods can substantially outperform advanced multivariate outlier detection methods for such data, which has important implication of choosing outlier detectors in real-world applications and evaluating newly proposed multivariate outlier detectors. In future work, we plan to design effective data indicator functions to determine whether univariate or multivariate methods should be applied to a given dataset.

REFERENCES

[1] Charu C. Aggarwal. 2013. Outlier ensembles: Position paper. *ACM SIGKDD Explorations Newsletter* 14, 2 (2013), 49–58.
[2] Charu C. Aggarwal. 2017. *Outlier Analysis*. Springer.
[3] Charu C. Aggarwal and Saket Sathe. 2015. Theoretical foundations and algorithms for outlier ensembles. *ACM SIGKDD Explorations Newsletter* 17, 1 (2015), 24–47.
[4] Fabrizio Angiulli, Fabio Fassetti, and Luigi Palopoli. 2009. Detecting outlying properties of exceptional objects. *ACM Transactions on Database Systems* 34, 1 (2009), 7.
[5] Fabrizio Angiulli and Clara Pizzuti. 2005. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering* 17, 2 (2005), 203–215.
[6] Vic Barnett and Toby Lewis. 1994. *Outliers in Statistical Data*. Wiley.
[7] Kenneth P. Burnham and David R. Anderson. 2003. *Model Selection and Multimodel Inference: A Practical Information-theoretic Approach*. Springer.
[8] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. 2016. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery* 30, 4 (2016), 891–927.
[9] Longbing Cao. 2014. Non-iidness learning in behavioral and social data. *The Computer Journal* 57, 9 (2014), 1358–1370.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys* 41, 3 (2009), 15.
[11] Robert B. Dean and W. J. Dixon. 1951. Simplified statistics for small numbers of observations. *Analytical Chemistry* 23, 4 (1951), 636–638.
[12] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. 2005. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media.
[13] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.
[14] Y. H. Dovoedo and Subha Chakraborti. 2015. Boxplot-based outlier detection for the location-scale family. *Communications in Statistics-Simulation and Computation* 44, 6 (2015), 1492–1513.
[15] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml
[16] Andrew F. Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. 2013. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. ACM, 16–21.
[17] Damien Francois, Vincent Wertz, and Michel Verleysen. 2007. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering* 19, 7 (2007), 873–886.
[18] Jing Gao and Pang-Ning Tan. 2006. Converting output scores from outlier detection algorithms into probability estimates. In *Proceedings of the 6th IEEE International Conference on Data Mining*. IEEE, 212–221.
[19] Frank E. Grubbs. 1969. Procedures for detecting outlying observations in samples. *Technometrics* 11, 1 (1969), 1–21.
[20] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10–18.
[21] Rob J. Hyndman and Han Lin Shang. 2010. Rainbow plots, bagplots, and boxplots for functional data. *Journal of Computational and Graphical Statistics* 19, 1 (2010), 29–45.
[22] Vladislav Ishimtsev, Alexander Bernstein, Evgeny Burnaev, and Ivan Nazarov. 2017. Conformal k-NN anomaly detector for univariate data streams. In *Proceedings of Machine Learning Research*. Vol. 60. 1–15.
[23] Shengyi Jiang, Xiaoyu Song, Hui Wang, Jianjun Han, and Qinghua Li. 2006. A clustering-based method for unsupervised intrusion detections. *Pattern Recognition Letters* 27, 7 (2006), 802–810.
[24] Fabian Keller, Emmanuel Muller, and Klemens Bohm. 2012. HiCS: High contrast subspaces for density-based outlier ranking. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*. 1037–1048.
[25] Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek. 2011. Interpreting and unifying outlier scores. In *Proceedings of the 11th SIAM International Conference on Data Mining*. 13–24.
[26] Aleksandar Lazarevic and Vipin Kumar. 2005. Feature bagging for outlier detection. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. ACM, 157–166.
[27] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2012. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data* 6, 1 (2012), 3:1–3:39.
[28] Hongfu Liu, Yuchao Zhang, Bo Deng, and Yun Fu. 2016. Outlier detection via sampling ensemble. In *Proceedings of the 2016 IEEE International Conference on Big Data*. IEEE, 726–735.
[29] Henrique O. Marques, Ricardo J. G. B. Campello, Arthur Zimek, and Jörg Sander. 2015. On the internal evaluation of unsupervised outlier detection. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*. ACM.
[30] Guansong Pang, Longbing Cao, Ling Chen, Defu Lian, and Huan Liu. 2018. Sparse modeling-based sequential ensemble learning for effective outlier detection in high-dimensional numeric data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 3892–3899.
[31] Guansong Pang, Kai Ming Ting, and David Albrecht. 2015. LeSiNN: Detecting anomalies by identifying least similar nearest neighbours. In *Proceedings of the 2015 IEEE International Conference on Data Mining Workshop*. IEEE, 623–630.
[32] Lucas Parra, Gustavo Deco, and Stefan Miesbach. 1996. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation* 8, 2 (1996), 260–269.
[33] Tomáš Pevnỳ. 2016. Loda: Lightweight on-line detector of anomalies. *Machine Learning* 102, 2 (2016), 275–304.
[34] Shebuti Rayana and Leman Akoglu. 2016. Less is more: Building selective anomaly ensembles. *ACM Transactions on Knowledge Discovery from Data* 10, 4 (2016), 42.
[35] Erich Schubert, Remigius Wojdanowski, Arthur Zimek, and Hans-Peter Kriegel. 2012. On evaluation of outlier rankings and outlier scores. In *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 1047–1058.
[36] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. 2014. Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery* 28, 1 (2014), 190–237.
[37] Mahito Sugiyama and Karsten Borgwardt. 2013. Rapid distance-based outlier detection via sampling. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*. 467–475.

[38] Kai Ming Ting, Takashi Washio, Jonathan R. Wells, and Sunil Aryal. 2017. Defying the gravity of learning curve: A characteristic of nearest neighbour anomaly detectors. *Machine Learning* 106, 1 (2017), 55–91.
[39] John W. Tukey. 1977. *Exploratory Data Analysis*. Addison-Wesley.
[40] Shanshan Wang and Robert Serfling. 2015. On masking and swamping robustness of leading nonparametric outlier identifiers for univariate data. *Journal of Statistical Planning and Inference* 162 (2015), 62–74.
[41] Mingxi Wu and Christopher Jermaine. 2006. Outlier detection by sampling with accuracy guarantees. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 767–772.
[42] Arthur Zimek, Ricardo J. G. B. Campello, and Jörg Sander. 2013. Ensembles for unsupervised outlier detection: Challenges and research questions. *SIGKDD Explorations Newsletter* 15, 1 (2013), 11–22.
[43] Arthur Zimek, Matthew Gaudet, Ricardo J. G. B. Campello, and Jörg Sander. 2013. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 428–436.
[44] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining* 5, 5 (2012), 363–387.

Received July 2019; revised March 2020; accepted May 2020