Outlier Detection Algorithm Based on Random Projection and Ensemble Learning (Postprint)

Authors: Guo Yiyang, Yu Jiong, Dù Xùshēng, Cao Ming

Date: 2022-05-10T00:00:00+00:00

Abstract

To address the issue that traditional similarity-based outlier detection algorithms exhibit unsatisfactory performance on high-dimensional imbalanced datasets, this paper proposes a novel ensemble learning and random projection-based outlier detection (EROD) framework. The algorithm first integrates multiple random projection methods to reduce the dimensionality of high-dimensional data, thereby enhancing data diversity; then integrates multiple distinct traditional outlier detectors to construct a heterogeneous ensemble model, increasing algorithmic robustness; finally, the heterogeneous model is trained on the dimensionality-reduced data, and the trained models undergo a two-stage optimized combination to reduce generalization error, outputting the final outlier scores for objects, where objects with high outlier scores are identified as outliers by the algorithm. Comparative experiments were conducted on four high-dimensional imbalanced real-world datasets from different domains. The results demonstrate that, compared with traditional outlier detection algorithms and ensemble learning-based outlier detection algorithms, the proposed algorithm achieves average improvements of 3.6% and 14.45% in AUC and Precision@n values, respectively, proving that the EROD algorithm possesses superior capabilities in handling anomalies in high-dimensional imbalanced data.

Full Text

Preamble

Vol. 39 No. 9
Application Research of Computers
ChinaXiv Cooperative Journal

Outlier Detection Algorithm Based on Random Projection and Ensemble Learning

Guo Yiyang[1] , Yu Jiong[1] ,[1] †, Du Xusheng[1] , Cao Ming[2]
(1. a. College of Information Science & Engineering; b. School of Software, Xinjiang University, Urumqi 830091, China;
2. College of Information Science & Engineering, Ocean University of China, Qingdao, Shandong 266100, China)

Abstract: To address the limitations of traditional similarity-based outlier detection algorithms on high-dimensional unbalanced datasets, this paper proposes a novel Ensemble learning and Random projection-based Outlier Detection (EROD) framework. The algorithm first integrates multiple random projection methods to reduce the dimensionality of high-dimensional data, thereby enhancing data diversity. It then constructs a heterogeneous ensemble model by integrating several different traditional outlier detectors to increase algorithmic robustness. Finally, the heterogeneous model is trained on the dimensionality-reduced data, and the trained models undergo two-stage optimal combination to reduce generalization error, outputting final outlier scores for objects. Objects with high outlier scores are identified as outliers. Comparative experiments on four high-dimensional unbalanced real-world datasets from different domains demonstrate that the proposed algorithm achieves average improvements of 3.6% and 14.45% in AUC and Precision@n values compared with traditional outlier detection algorithms and ensemble learning-based outlier detection algorithms, proving that EROD effectively handles anomalies in high-dimensional unbalanced data.

Keywords: data mining; outlier detection; random projection; ensemble learning

---

0 Introduction

Compared with normal data, outliers are data points with distinct characteristics. They are defined as follows: assuming a certain data point deviates significantly from the vast majority of other data in the dataset, then this data point is considered to be generated by a mechanism different from other data and is thus identified as an outlier [1]. The reason why outlier removal is an indispensable preprocessing step in data mining is that the presence of outliers severely impacts the results of statistical data analysis [2]. Therefore, to remove outliers, they must first be identified, which is the primary objective of outlier detection algorithms.

Outlier detection is an important machine learning task that can detect anomalous objects among regular data objects in many high-risk applications, such as traffic anti-fraud detection. According to the *2020 China Abnormal Traffic Report*, abnormal traffic accounts for approximately 8.6 percentage points of the total. As the world's largest advertising traffic platform, Alimama (a subsidiary of Alibaba Group) manages over $100 billion in commercial traffic, making it a primary target for black and gray industries. From the business perspective

of the Alimama team, the core idea of traffic anti-fraud detection is identifying fraudulent and low-quality abnormal traffic content to protect the rights of customers and the platform. In the current machine learning domain, traffic anti-fraud detection may be the business with the highest requirements for algorithmic robustness and interpretability, precision, system scale and timeliness, and industry scale. Consequently, traffic anti-fraud detection technology teams must have an "ironclad" foundation to integrate outlier detection technology more closely with traffic anti-fraud applications. In traffic anti-fraud detection tasks, outlier detection for high-dimensional unbalanced data has become the primary focus of relevant teams both domestically and internationally.

Similarity-based outlier detection algorithms are common traditional unsupervised machine learning methods. However, when detecting high-dimensional data, these algorithms face the challenge of the curse of dimensionality in distance calculation, making it difficult to measure the similarity of object distribution patterns in high-dimensional space. This leads to problems such as low detection rates and high parameter sensitivity when detecting high-dimensional unbalanced datasets. In real-world industrial environments without true data labels, engineers typically need to construct numerous unsupervised heterogeneous ensemble models—ensembles of different algorithms with different hyperparameters—for further combined research and analysis, rather than relying on a single algorithm. Therefore, this paper proposes an outlier detection algorithm based on ensemble learning and random projection (EROD).

To improve the detection accuracy of traditional outlier detection algorithms on high-dimensional unbalanced datasets, EROD applies random projection to reduce the dimensionality of the dataset to be detected, integrates traditional outlier detection algorithms to calculate outlier scores for all data objects in the reduced-dimensional data, and dynamically groups and optimally combines the traditional outlier detection algorithms. The combined outlier scores serve as the final outlier scores determined by the algorithm. Experiments on real-world UCI (University of California, Irvine) datasets demonstrate that EROD achieves significantly improved detection rates compared with other outlier detection algorithms.

The main contributions of this paper are summarized as follows:

a) We propose a novel unsupervised outlier detection framework that performs heterogeneous ensemble at both the data and model levels. By integrating random projection methods to enhance data diversity and integrating traditional outlier detection algorithms to enhance model diversity, the framework improves detection rates through two-stage combination.

b) To address the instability of traditional outlier detection algorithms on different high-dimensional unbalanced datasets, we leverage ensemble characteristics to balance these algorithms, making the overall framework more stable and improving detection rates.

c) We conduct a comprehensive parameter sensitivity analysis of traditional

outlier detection algorithms, predict the relationship between framework parameters and performance, and provide visual analysis and discussion for specific datasets.

1 Related Work

Since the 19th century, researchers have conducted scientific studies on outlier detection [3]. Statistical and probability-based outlier detection methods are early approaches proposed. These methods detect outliers based on statistical and probabilistic principles and have the advantage of low time complexity. The core idea is to first estimate the distribution model of the dataset, then assume that the data objects in the dataset follow the distribution pattern of that model, and finally detect outliers in the dataset by evaluating whether data objects are consistent with the distribution model. Inspired by the statistical Copula function, literature [4] utilizes the Copula function to predict the tail distribution probability of each given sample to determine its outlier degree. However, this method requires accurately calculating the parameters of the distribution model in advance. If these parameters cannot be accurately estimated beforehand, significant differences between the estimated and true values will substantially reduce the accuracy of outlier detection.

Due to the limitations of statistical and probability-based outlier detection methods, similarity-based outlier detection methods were proposed in the early 21st century. These methods detect outliers by measuring the similarity between data objects (e.g., distance, density, angle) based on the characteristic that normal points and outliers have different distributions in the dataset. In literature [5], k Nearest Neighbors (kNN), Average k Nearest Neighbors (Avg-kNN), and Median k Nearest Neighbors (k-Median) detect outliers by calculating Euclidean distances between samples. However, they are highly sensitive to parameter settings and have low detection rates for high-dimensional data. Literature [6] proposes the first density-based clustering Local Outlier Factor (LOF) detection method, which assigns an outlier factor to each data object, solving the problem of treating outlier values as binary attributes but unable to handle multi-granularity and hyperparameter sensitivity issues. Tang et al. [7] improve LOF by proposing the Connectivity-based Outlier Factor (COF) method, which estimates the local density of neighbors by calculating connection distances as the shortest path. The key idea is based on the distinction between low density and isolation, but this method consumes more computational cost than LOF. Literature [8] proposes the Angle-Based Outlier Detection (ABOD) method, which uses the variance of weighted cosine scores with all nearest neighbors as the outlier score. This method has relatively complex decision boundaries, easily leading to overfitting.

From reviewing relevant literature, we learn that different base detectors in ensemble learning produce independent errors. Combining multiple base detectors

can alleviate problems such as hyperparameter sensitivity, training difficulty, and poor fitting effects of single base detectors to some extent [9]. Literature [10] proposes the feature bagging (FB) outlier detection algorithm, which separates original features and creates random feature subsets, then merges multiple algorithms applied to these subsets to generate corresponding outlier scores. This algorithm improves detection performance, but since its detectors are homogeneous, the method lacks sufficient diversity. Literature [11] proposes a lightweight on-line detector of anomalies (LODA) method that detects outliers by identifying data deviating from the majority of features. This algorithm has low time complexity, but its detection rate is relatively low due to unstable output results from individual detectors. Literature [12] proposes the Isolation Forest (IForest) algorithm, which integrates multiple isolation trees and records the path lengths of these trees as the basis for calculating outlier scores. However, if the proportion of outlier samples is high, it conflicts with the theoretical foundation that outliers are easily isolated, resulting in unsatisfactory outcomes.

Evidently, ensemble learning-based outlier detection algorithms can effectively detect outliers by focusing on combining model outputs to generate stable ensemble models. This provides insight for solving the aforementioned limitations of similarity-based outlier detection algorithms—namely, the EROD algorithm. EROD leverages ensemble characteristics to balance traditional algorithms, and the theoretical foundations of its component detectors complement each other, improving algorithmic robustness. Simultaneously, EROD performs heterogeneous ensemble at both the data and model levels, enhancing overall structural diversity and improving detection rates through two-stage combination.

Therefore, compared with the above outlier detection algorithms, EROD has the advantages of stronger robustness, higher detection rates, and no reliance on prior assumptions.

2 Methodology and Theoretical Properties

This section first presents the framework and process of the EROD outlier detection algorithm based on random projection and ensemble learning, then introduces the integrated random projection method, heterogeneous ensemble model, and two-stage aggregation algorithm. Table 1 lists the partial symbol definitions required for subsequent content.

Table 1 Definition of symbols

Symbol	Definition
m	Number of component detectors
A	Random projection matrix
X	Original dataset

Symbol	Definition
Y_i	m datasets generated after random projection of X using m A matrices, $i = 1, 2, \ldots, m$
y_j	The j-th data point in Y_i
D_i	The i-th component detector for detecting Y_i
$D_i(y_j)$	Outlier score of y_j on the i-th component detector
OF	Outlier value matrix of Y_i, composed of elements $D_i(y_j)$
ZOF	Outlier value matrix after normalization of OF
row	Number of rows in ZOF
$outlierScore$	Final outlier score determined by EROD algorithm

2.1 EROD Algorithm Framework and Process

The EROD algorithm determines the final outlier score set for objects through three implementation steps:

a) **Dimensionality reduction**: Primarily uses random projection methods to project high-dimensional data into low-dimensional space.

b) **Construction of component detector ensemble model**: To enhance the robustness of EROD, different categories of outlier detection models are heterogeneously integrated.

c) **Two-stage aggregation**: The multiple component detectors in the heterogeneous ensemble are randomly divided into several different clusters. The maximum value is selected from each cluster, and the mean of these maximum values is calculated as the final outlier score determined by EROD.

The overall framework and process of the EROD algorithm are shown in Figure 1.

Table 2 Description of random projection methods

Random Projection Method	Description of A or A_{ij}
Gaussian Random Projection	A_{ij} satisfies independent standard normal distribution
Discrete Random Projection	A_{ij} satisfies independent discrete distribution
Circulant Random Projection	A is a circulant matrix

Random Projection Method	Description of A or A_{ij}
Sparse Random Projection	A_{ij} satisfies independent sparse distribution

2.2 Random Projection Ensemble

During outlier detection, most outlier detection algorithms are severely affected by the curse of dimensionality on high-dimensional data [13]. To solve this problem, JL random projection is widely used to eliminate the negative effects of the curse of dimensionality. JL random projection is a dimensionality reduction algorithm widely used in outlier detection because its reduction mechanism preserves pairwise relative distances between data, enabling low-distortion compression of high-dimensional data in Euclidean space while preserving outlier information during compression. More importantly, the random mechanism of JL random projection enhances the diversity of ensemble learning.

The purpose of JL random projection is approximate distance preservation, with its theoretical foundation being the Johnson-Lindenstrauss lemma [14]. As shown in Equation (1), JL random projection represents a linear mapping relationship $f : \mathbb{R}^d \to \mathbb{R}^k$, which projects d-dimensional data into k-dimensional data. As shown in Equation (2), the Johnson-Lindenstrauss lemma states that for $1 \leq i \neq j \leq n$, $\epsilon \in (0, 3)$, to satisfy with high probability P that the relative distance between pairwise data objects remains within $(1-\epsilon, 1+\epsilon)$, data objects must be reduced to $k = O(\log(n)/\epsilon^2)$ dimensions.

As shown in Table 2, according to four widely used random matrices A in literature [15], JL random projection can be divided into four methods.

Among the four JL random projection methods, sparse random projection has slightly better time efficiency than the other three methods [16]; therefore, EROD adopts sparse random projection.

As shown in Equations (3) and (4), the feature space of the original dataset X consists of n data points with d-dimensional features. The sparse random matrix A comprises m different sparse random projection matrices, each $\in \mathbb{R}^{d\times k}$. Y_i is the data with n k-dimensional features obtained by applying sparse random matrix A to the original dataset X, where $0 < k < d$ and $i = 1, 2, \ldots, m$.

EROD uses JL random projection for ensemble through the following process: First, EROD uses sparse random projection to generate m different sparse random projection matrices $A \in \mathbb{R}^{d\times k}$. Then, these m sparse matrices A project the high-dimensional dataset $X \in \mathbb{R}^{n\times d}$ to obtain m projected datasets $Y_i \in \mathbb{R}^{n\times k}$. Finally, Y_i is stored in set Y, and set Y is output.

The specific process is shown in Algorithm 1.

Algorithm 1 Random Projection Ensemble Algorithm

Input: Data $X \in \mathbb{R}^{n\times d}$, dimension k after dimensionality reduction of dataset X.
Output: Set Y.

a) Initialize m Sparse Random Projection matrices $A = \{A_1, A_2, A_3, \dots, A_m\} \in \mathbb{R}^{d\times k}$

b) for A_i in A do
c) $Y_i = \langle X, A_i \rangle \in \mathbb{R}^{n\times k}$ // Project X to obtain projected data Y_i
d) Add(Y_i, Y) // Store data Y_i in set Y

c) end for

d) Output(Y) // Output set Y

2.3 Heterogeneous Ensemble Learning

The EROD outlier detection algorithm selects kNN detector, Avg-kNN detector, k-Median detector, LOF detector, COF detector, and ABOD detector as component detectors for the heterogeneous ensemble learning model (i.e., $m = 6$).

These six different outlier detection algorithms are chosen as component detectors in the heterogeneous ensemble learning model because identical outlier detection algorithms producing identical outputs have limited positive effects on ensemble learning [17]. In other words, heterogeneous ensemble learning models constructed from different outlier detection algorithms generally produce significant positive effects. This is because different component detectors promote diversity in the learning process, enabling the learning of different data features and further improving model generalization capability. Additionally, outlier detection algorithms with high similarity produce similar errors, which negatively impacts prediction results [18].

While using different, low-detection-rate outlier detection algorithms ensures a certain degree of diversity, the prediction rate of the model will decrease. Therefore, a balance must be struck between diversity and detection rate.

Consequently, this paper uses six outlier detection algorithms with different characteristics and relatively high detection rates among all mainstream outlier detection algorithms—kNN, Avg-kNN, k-Median, LOF, COF, and ABOD—as component detectors for the heterogeneous ensemble learning model.

As shown in Equation (5), the scores obtained by each component detector in the heterogeneous ensemble learning model on data Y are called $Outlier\_Factor$. The output of each component detector is $D(X) \in \mathbb{R}^{n\times 1}$.

The heterogeneous ensemble process is as follows: First, initialize the six component detectors in the heterogeneous ensemble model. Second, use the initialized component detectors to detect data Y output by Algorithm 1. Finally, the output values of the component detectors are determined as the outlier scores of

data Y. The specific process is shown in Algorithm 2.

Algorithm 2 Heterogeneous Ensemble Learning Algorithm

Input: Set $Y = \{Y_1, Y_2, Y_3, \ldots, Y_m\}$, set $D = \{D_1, D_2, D_3, \ldots, D_m\}$.
Output: Outlier value matrix OF.

a) for $i = 1 : \text{Size}(D)$ do
b) Initialize component detector D_i /* Initialize each component detector */

b) end for

c) for Y_i in Y do // Iterate through set Y
e) for y_j in Y_i do // Iterate through dataset Y_i
f) $OF = D_i(y_j)$ /* Use the i-th component detector to detect y_j, obtain outlier score $D_i(y_j)$ as an element in outlier value matrix OF */
g) end for

d) end for

e) Output(OF) // Output outlier value matrix OF

The outlier value matrix OF output by all component detectors in Algorithm 2 on dataset Y_i is shown in Equation (6).

The physical meaning of outlier value matrix OF: This matrix consists of outlier factors of all samples in dataset Y_i, meaning each element in the matrix represents the outlier degree assessed by a certain detector for a certain sample [19, 20].

2.4 Two-Stage Aggregation Method

As shown in Figure 2, there is an inverse relationship between bias and variance: as the complexity of the ensemble learning model increases, bias decreases while variance increases. This is because models with low complexity have insufficient fitting capability (i.e., component detectors have insufficient learning ability), where bias dominates generalization error; conversely, variance dominates generalization error.

Typically, taking the mean of component detectors can reduce variance and increase bias, while taking the maximum of component detectors can reduce bias and increase variance. Using either combination method alone may cause the obtained outlier scores to deviate significantly from true scores [21].

Therefore, reasonably combining both mean and maximum combination methods for component detectors can balance bias and variance, reducing generalization error to a reasonable range and improving detection rates.

Since generalization error can be approximated as the sum of the square of bias and variance, in the first stage, taking the maximum of component detectors minimizes generalization error; in the second stage, taking the mean of the remaining component detectors minimizes the increase in bias, thereby minimizing the rise in generalization error.

The two-stage aggregation process is as follows: First, normalize the output of Algorithm 2 to standardize the output values of different outlier detection models to the same scale. Second, randomly divide the six component detectors into two clusters, with the three outlier detection models in each cluster being mutually exclusive. Finally, select the maximum value from each cluster as the cluster representative value, and calculate the average of these cluster representative values as the final outlier score determined by EROD. The specific process is shown in Algorithm 3.

Algorithm 3 Two-Stage Aggregation Algorithm

Input: Outlier value matrix OF.
Output: Final outlier scores determined by EROD algorithm.

a) ZOF = Z-normalization(OF) /* Normalize OF (to avoid cluttered data representation, the normalized data still uses mathematical symbols from Table 1) */

b) row = countRow(ZOF) // Calculate matrix ZOF rows

c) for $j = 1 : row$ do // Iterate through the j-th data in $Y_1 \sim Y_6$
d) for $i = 1 : 6$ do // Iterate through component detectors
e) $detectors = D_i(y_j)$ // Store outlier values from each row of matrix ZOF in set $detectors$
f) end for
g) $group1, group2$ = randomDivide($detectors$) // Divide into clusters
h) $max1$ = Max($group1$)
i) $max2$ = Max($group2$)
j) $outlierScore$ = Average($max1, max2$)

d) end for

e) Output($outlierScore$)

2.5 Time Complexity Analysis

Let the number of data points and dimensions be n and d, respectively. In Algorithm 1, preprocessing data and iterating through data for random projection has time complexity $O(n)$. In Algorithm 2, using component detectors to calculate data makes the complexity dependent on the component detectors. Since COF and ABOD detectors are both Fast versions, the time complexities of

kNN, Avg-kNN, k-Median, LOF, COF, and ABOD detectors are $O(nd)$, $O(nd)$, $O(nd)$, $O(n)$, $O(n^2)$, and $O(n^2)$, respectively. Therefore, this stage has time complexity $O(n^2)$. In Algorithm 3, this stage's task is to optimally combine the calculation results from Algorithm 2, with time complexity $O(n)$.

In summary, the time complexity of the EROD algorithm is $O(n^2)$.

---

3 Experiments

3.1 Experimental Environment

The experimental hardware environment is: CPU @ 2.00GHz 2.00 GHz (2 processors), GPU is Nvidia Tesla V100-PCIE-16GB (3 units total), and memory (RAM) is 256GB. The experimental software environment is: operating system Microsoft Windows Server 2016 Standard, algorithm implementation environment PyCharm Professional, Python 3.6.2, TensorFlow 1.14.

3.2 Datasets

As shown in Table 3, to evaluate the detection performance of the proposed method, we selected four real-world datasets from the UCI data repository with different practical application scenarios. The specific information for each dataset is detailed below:

a) **Arrhythmia dataset**: This original dataset contains arrhythmia information and is a multi-class classification dataset with 16 classes and 279 dimensions, used to distinguish the presence of arrhythmia phenomena. The original dataset was preprocessed by deleting 5 dimensions. Small categories including 3, 4, 5, 7, 8, 9, 14, 15, etc., were defined as outliers, with remaining classes as normal. The processed dataset contains 452 sample objects, each with 274 dimensions, including 66 outlier samples.

b) **Mnist dataset**: This original dataset contains handwritten digit image information from digits 0 to 9 (10 image categories). The original dataset was preprocessed by defining digit 0 as normal and other digits as outliers. 100 dimensions were randomly selected from the original 784 dimensions as processed sample dimensions. The processed dataset contains 7,603 sample objects, each with 100 dimensions, including 700 outlier samples.

c) **Musk dataset**: This original dataset contains musk molecule information, used to distinguish whether molecules are musk. The original dataset was preprocessed by defining non-musk classes j146, j147, and 252 as normal, and musk classes 213 and 211 as outliers, with other categories deleted. The processed dataset contains 3,062 sample objects, each with 166 dimensions, including 97 outlier samples.

d) **Speech dataset**: This dataset contains real-world speech information, where the American accent (largest proportion) is defined as normal and

other accents as outliers. The dataset contains 3,686 sample objects, each with 400 dimensions, including 61 outlier samples.

Table 3 Details of four datasets

Dataset	Samples	Dimensions	Outliers	Outlier Ratio (%)
Arrhythmia	452	274	66	14.6
Mnist	7,603	100	700	9.2
Musk	3,062	166	97	3.2
Speech	3,686	400	61	1.7

3.3 Evaluation Metrics

Evaluation metrics play an indispensable role in assessing detection performance and guiding detector modeling. Since the datasets used in this paper are all unbalanced, the Accuracy metric often lacks reference value when data is imbalanced. In the machine learning domain, AUC (Area Under Curve) and Precision@n are used for such datasets. Therefore, this paper employs these two evaluation metrics.

AUC is the area under the ROC (Receiver Operating Characteristic) curve. Higher scores indicate stronger algorithmic detection performance. The calculation formula is shown in Equation (7), where n^+ and n^- represent the numbers of positive and negative samples, respectively, x_i and x_j represent the i-th and j-th samples, d represents the detector, and $I[\cdot]$ represents the indicator function that equals 1 when its parameter is true and 0 otherwise.

Precision@n is a special case of the Precision metric. This evaluation metric measures the Precision score output by the detector when the outlier threshold is set to a specified n positive examples. The calculation formula is shown in Equation (8), where TP (True Positive) represents the number of outlier samples correctly labeled as outliers, and FP (False Positive) represents the number of normal samples incorrectly labeled as anomalies.

3.4 Experimental Design

To verify the effectiveness of EROD's integration of multiple component detectors, we conducted comparative experiments between the proposed method and six component detectors (kNN, Avg-kNN, k-Median, LOF, COF, ABOD) as well as three ensemble learning algorithms (FB, LODA, IForest). To ensure EROD's timeliness, we also compared EROD with two newer similar methods—EAOD (ensemble and autoencoder-based outlier detection) [22] and GAN-VAE (generative adversarial network and variational auto-encoder based outlier detection) [23]—on the high-dimensional unbalanced dataset Mnist using AUC as the evaluation metric.

In the experiments, to balance the curse of dimensionality and data diversity, JL random projection compressed data dimensions to two-thirds of the original.

To investigate the sensitivity of EROD's decisive parameters, we conducted sensitivity experiments on the neighbor parameter k for each component detector in the ensemble learning model. We further selected k values with positive impact on EROD's detection performance to establish the EROD outlier detection model.

In the experiments, after analyzing and selecting k values with positive impact on EROD's detection performance, the parameter k for comparison algorithms kNN, Avg-kNN, k-Median, LOF, COF, and ABOD was kept consistent with the corresponding component detectors in EROD. In the ensemble learning algorithms, FB's base detector was set to LOF with parameters consistent with EROD's LOF component detector; LODA's parameters were automatically optimized; IForest's sampling size parameter Ψ was set to 256 and tree number parameter tn to 100. To ensure experimental fairness and reasonableness, the number of detectors in EAOD was set equal to that in EROD.

To ensure stable experimental results, EROD and comparison algorithms were each executed 10 times, with the mean of these results calculated as the final result.

3.5 Parameter Sensitivity Analysis and Selection

To use EROD for outlier detection, this paper conducted comparative experiments with different values for the neighbor parameter k in each component detector of the ensemble model. We further selected k values with positive impact on EROD's detection performance to establish the EROD outlier detection model.

The specific selection strategy for neighbor parameter k is as follows: First, the range of neighbor parameter k is $[10, 100]$ with an interval of 10. Then, on different k values, we analyze the four AUC scores of component detectors on the Arrhythmia, Mnist, Musk, and Speech datasets, and calculate their average. Finally, we select the maximum of the 10 AUC averages, with the corresponding k value serving as the reference for ideal neighbor parameter selection for component detectors. The specific process is shown in Algorithm 4.

Algorithm 4 Component Detector Neighbor Parameter k Selection Strategy

Input: Initial k value, component detector D, datasets Arrhythmia, Mnist, Musk, Speech.
Output: Reference k value.

a) $k = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$

b) $AUC = []$

c) $avgAUC = []$

d) $max = 0$

e) $j = 1$

f) for $i = k[j]$, $i < 101$, $j = j + 1$ do
g) AUC.append($D(i,$ Arrhythmia$)$)
h) AUC.append($D(i,$ Mnist$)$)
i) AUC.append($D(i,$ Musk$)$)
j) AUC.append($D(i,$ Speech$)$)
k) $avgAUC$.append(Average(AUC))

g) end for

h) $k\_reference = \text{Max}(avgAUC)$

i) output($k\_reference$)

As shown in Figure 3, for the kNN component detector on the Arrhythmia, Mnist, Musk, and Speech datasets: The AUC average shows a significant upward trend as k increases from 10 to 40; the increase becomes minor as k increases from 40 to 80; the AUC average shows an insignificant decline as k increases from 80 to 90; the AUC averages at $k = 90$ and $k = 100$ are equal; the maximum AUC average of 0.7838 is reached at $k = 80$. However, the AUC average changes little from $k = 40$ onward. Therefore, the kNN component detector is optimal at $k = 40$.

As shown in Figure 4, for the Avg-kNN component detector: The AUC average shows an upward trend as k increases from 10 to 100, with significant increase from $k = 10$ to $k = 50$ and minor increase after $k = 50$. The maximum AUC average of 0.7840 is reached at $k = 100$. Therefore, the Avg-kNN component detector is optimal at $k = 50$.

As shown in Figure 5, for the k-Median component detector: The AUC average continuously increases as k increases from 10 to 100, with significant increase from $k = 10$ to $k = 60$ and minor increase from $k = 60$ to $k = 100$. The maximum AUC average of 0.7821 is reached at $k = 100$. Therefore, the k-Median component detector is optimal at $k = 60$.

As shown in Figure 6, for the LOF component detector: The AUC average first increases, then decreases, then increases again as k increases from 10 to 100. Specifically, it increases significantly from $k = 10$ to 20, decreases approximately linearly from $k = 20$ to 80, and surges from $k = 80$ to 100. The maximum AUC average of 0.7612 is reached at $k = 100$, which is significantly higher than other k values. Therefore, the LOF component detector is optimal at $k = 100$.

As shown in Figure 7, for the COF component detector: The AUC average first increases then decreases as k increases from 10 to 100, rising from $k = 10$ to 50 and declining from $k = 50$ to 100. The peak AUC average of 0.6397 is reached

at $k = 50$. Therefore, the COF component detector is optimal at $k = 50$.

As shown in Figure 8, for the ABOD component detector: The AUC average first decreases, then increases, but with very minor variation. The peak AUC average of 0.5807 is reached at $k = 10$. Therefore, the ABOD component detector is optimal at $k = 10$.

In summary, the neighbor parameters k for the six component detectors (kNN, Avg-kNN, k-Median, LOF, COF, ABOD) are optimal at values of 40, 50, 60, 100, 50, and 10, respectively. These neighbor parameter values are selected for the component detectors in the EROD algorithm.

3.6 Experimental Results and Analysis

Table 4 presents comparison results between EROD and kNN, Avg-kNN, k-Median, LOF, COF, and ABOD on four different high-dimensional datasets. Bold numbers in the table represent the two algorithms with strongest detection performance. Figures 9 and 10 show AUC and Precision score comparisons across different datasets.

Table 5 presents comparison results between EROD and three ensemble learning algorithms (FB, LODA, IForest) on four high-dimensional datasets. Bold numbers represent the two algorithms with strongest detection performance. Figures 11 and 12 show AUC and Precision score comparisons across different datasets.

Figure 13 shows the comparison between EROD and two newer similar methods (EAOD and GAN-VAE) on the high-dimensional unbalanced dataset Mnist using AUC as the evaluation metric.

Compared with component detectors, EROD's two evaluation metrics are superior to other algorithms on Arrhythmia, Mnist, and Musk: on Arrhythmia, AUC and Precision scores improved by 1.2 and 1.7 percentage points compared to the second-best algorithm; on Mnist, improvements of 1.3 and 2.7 percentage points; on Musk, improvements of 0.9 and 1.6 percentage points. However, on Speech, EROD's metrics are second-highest because most component detectors perform poorly on this dataset, reducing EROD's ability to balance generalization error, though EROD still outperforms most component detectors.

Compared with other ensemble learning algorithms, EROD's metrics are superior on Arrhythmia, Mnist, and Speech: on Arrhythmia, improvements of 1.2 and 3.4 percentage points; on Mnist, improvements of 8.2 and 44 percentage points; on Speech, improvements of 11.2 and 33.3 percentage points. On Musk, EROD's Precision score is slightly lower than IForest, but its AUC score is superior to all others, with a 1.2 percentage point improvement over the second-best algorithm. This is because different metrics have different statistical emphases, causing EROD's AUC and Precision scores to be high and low, respectively.

Compared with newer similar methods EAOD and GAN-VAE on the high-

dimensional unbalanced dataset Mnist, EROD's AUC scores improved by 1.02% and 0.46%, respectively, demonstrating EROD's advancement in solving such problems.

In Tables 4 and 5, all algorithms show relatively low scores on the Speech dataset. As shown in Figure 14, the Speech dataset's 2-D visualization (red diamonds represent outliers, others represent normal points) reveals that outliers and normal points are highly mixed, with outliers hidden inside normal points and not positioned in the tail of the dimensional distribution. This makes them highly similar to normal points in dimensional distribution, preventing outlier detection algorithms from achieving optimal performance. Only when outliers are clearly exposed in the tail can they be accurately captured and identified.

In summary, comparative experiments with various outlier detection algorithms on multiple high-dimensional datasets verify the effectiveness and feasibility of the EROD algorithm.

---

4 Conclusion

This paper proposes a novel outlier detection framework—EROD. The algorithm integrates random projection for dimensionality reduction of high-dimensional data while enhancing data diversity. By integrating multiple heterogeneous outlier detectors, algorithmic robustness is improved. The heterogeneous ensemble model then trains on multiple dimensionality-reduced datasets, and the trained models are combined twice to effectively reduce generalization error and improve detection performance. We also theoretically analyze parameter sensitivity and discuss the basis for hyperparameter selection when integrating component detectors. Experiments on UCI datasets using AUC and Precision as evaluation metrics demonstrate that EROD has advantages in handling anomalies in high-dimensional unbalanced data compared with traditional outlier detection algorithms and ensemble learning-based outlier detection algorithms. Future work will further investigate the impact of different dimensionality reduction methods and detectors on EROD experimentally, and theoretically analyze the relationship between EROD's generalization error critical point and its component detectors' generalization error critical points.

References

[1] Boukerche A, Zheng L, Alfandi O. Outlier detection: Methods, models, and classification [J]. ACM Computing Surveys, 2020, 53(3): 1-37.
[2] Najafi M, He L, Philip S Y. Outlier-Robust Multi-Aspect Streaming Tensor Completion and Factorization [C]// Proc of the 28th International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann Press, 2019: 3187-3194.

[3] Walfish S. A review of statistical outlier methods [J]. Pharmaceutical Technology, 2006, 30(11): 82.
[4] Li Z, Zhao Y, Botta N, et al. COPOD: copula-based outlier detection [C]// Proc of the 20th International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2020: 1118-1123.
[5] Aggarwal C C. Outlier analysis [M]. 2nd ed. Berlin: Springer Press, 2017: 237-263.
[6] Breunig M M, Kriegel H P, Ng R T, et al. LOF: identifying density-based local outliers [C]// Proc of SIGMOD. New York: ACM Press, 2000: 93-104.
[7] Tang J, Chen Z, Fu A W C, et al. Enhancing effectiveness of outlier detections for low density patterns [C]// Proc of PAKDD. Berlin: Springer Press, 2002: 535-548.
[8] Kriegel H P, Schubert M, Zimek A. Angle-based outlier detection in high-dimensional data [C]// Proc of the 14th ACM Knowledge Discovery and Data Mining. New York: ACM Press, 2008: 444-452.
[9] Chen W, Wang Z, Zhong Y, et al. ADSIM: Network Anomaly Detection via Similarity-aware Heterogeneous Ensemble Learning [C]// Proc of the 17th IFIP/IEEE International Symposium on Integrated Network Management. Piscataway, NJ: IEEE Press, 2021: 608-612.
[10] Lazarevic A, Kumar V. Feature bagging for outlier detection [C]// Proc of the 11th ACM Knowledge Discovery and Data Mining. New York: ACM Press, 2005: 157-166.
[11] Pevny T. Loda: Lightweight on-line detector of anomalies [J]. Machine Learning, 2016, 102(2): 275-304.
[12] Liu F T, Ting K M, Zhou Z H. Isolation Forest [C]// Proc of the 8th International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2008: 413-422.
[13] Pang G, Cao L, Chen L, et al. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection [C]// Proc of the 24th ACM Knowledge Discovery and Data Mining. New York: ACM Press, 2018: 2041-2050.
[14] Cohen M B, Jayram T S, Nelson J. Simple analyses of the sparse Johnson-Lindenstrauss transform [C]// Proc of the 1st Symposium on Simplicity in Algorithms. Philadelphia, PA: SIAM Press, 2018: 1-9.
[15] Jin R, Kolda T G, Ward R. Faster Johnson–Lindenstrauss transforms via kronecker products [J]. Information and Inference: A Journal of the IMA, 2021, 10(4): 1533-1562.
[16] Venkatasubramanian S, Wang Q. The Johnson-Lindenstrauss transform: an empirical study [C]// Proc of the 13th Workshop on Algorithm Engineering and Experiments. Philadelphia, PA: SIAM Press, 2011: 164-173.
[17] Pasillas-Diaz J R, Ratte S. An unsupervised approach for combining scores of outlier detection techniques, based on similarity measures [J]. Electronic Notes in Theoretical Computer Science, 2016, 329: 61-77.
[18] Aggarwal C C, Sathe S. Outlier ensembles: An introduction [M]. Berlin: Springer Press, 2017: 35-73.
[19] Du Xusheng, Yu Jiong, Chen Jiaying, et al. An outlier detection algorithm

based on neighborhood system density difference measurement [J]. Application Research of Computers, 2020, 37(07): 1969-1973.
[20] Du Xusheng, Yu Jiong, Ye Lele, et al. Outlier detection algorithm based on graph random walk [J]. Journal of Computer Applications, 2020, 40(05): 1322-1328.
[21] Aggarwal C C, Sathe S. Theoretical foundations and algorithms for outlier ensembles [J]. ACM SIGKDD Explorations Newsletter, 2015, 17(1): 24-47.
[22] Guo Yiyang, Yu Jiong, Du Xusheng, et al. Outlier detection algorithm based on autoencoder and ensemble learning [J/OL]. Journal of Computer Applications. [2022-03-25]. http://kns.cnki.net/kcms/detail/51.1307.tp.20210929.1142.006.html.
[23] Jin Lina, Yu Jiong, Du Xusheng, et al. Generative adversarial network and variational auto-encoder based outlier detection [J]. Application Research of Computers, 2022, 39(03): 774-779.

Note: Figure translations are in progress. See original paper for figures.