Dimensional Outlier Detection

Jiawei Yang¹, Sylwan Rahardja¹, and Susanto Rahardja¹

¹Affiliation not available

March 29, 2024

Abstract

Few outlier detectors have considered all industrial standards in time complexity, space complexity, accuracy, interpretability, and scalability, which is very challenging in big data applications. To address these challenges, we proposed a framework for dimensional outlier detection (FDOD) to detect outliers that are dimensionally separable. Based on FDOD, two detectors were discovered. Both outperformed all 22 baseline state-of-the-art detectors including 10 detectors published in recent three years with 18 real-world datasets. Compared to existing standards, one proposed detector had around 4% improvement evaluated by the area under the receiver operating characteristic (ROC AUC) and required only 42% computational time and 13% memory usage and another proposed detector was even better. The techniques suggested are suitable for deployment in mobile settings that require the utilization of lightweight models. This study also poses a question to the outlier detection domain that whether technically complex solutions are really needed for outlier detection in big data applications since normally complex solutions require more computational resources. Besides, we advocate for opening larger outlier detection datasets from the industry to support the development of outlier detection in big data. The implementation of the proposed methods can be found on www.OutlierNet.com for reproducibility.
Dimensional Outlier Detection

Jiawei Yang, member, IEEE, Sylwan Rahardja, student member, IEEE, Susanto Rahardja, Fellow, IEEE

Abstract—Few outlier detectors have considered all industrial standards in time complexity, space complexity, accuracy, interpretability, and scalability, which is very challenging in big data applications. To address these challenges, we proposed a framework for dimensional outlier detection (FDOD) to detect outliers that are dimensionally separable. Based on FDOD, two detectors were discovered. Both outperformed all 22 baseline state-of-the-art detectors including 10 detectors published in recent three years with 18 real-world datasets. Compared to existing standards, one proposed detector had around 4% improvement evaluated by the area under the receiver operating characteristic (ROC AUC) and required only 42% computational time and 13% memory usage and another proposed detector was even better. The techniques suggested are suitable for deployment in mobile settings that require the utilization of lightweight models. This study also poses a question to the outlier detection domain that whether technically complex solutions are really needed for outlier detection in big data applications since normally complex solutions require more computational resources. Besides, we advocate for opening larger outlier detection datasets from the industry to support the development of outlier detection in big data.

The implementation of the proposed methods can be found on www.OutlierNet.com for reproducibility.

Index Terms—outlier detection, anomaly detection, FDOD, DDM, KOBE, dimensional outlier

I. INTRODUCTION

OUTLIERS are observations that deviate significantly from other observations, raising suspicions that they were produced by different mechanisms [1]. With more than two centuries of history in statistics, outlier detection began its modern history from astronomical observations. Over time, outlier detection has acquired increasingly complex techniques, with impressive results and modern-day applications. Outlier detection flourishes in computer science due to its relevance to applications such as network intrusions, terrorism detection, fake news detection, error detection, fraud detection, noise removal, and detection of rare events [2]. Recently, outlier detection has been in the spotlight due to applications in monitoring industrial system [3], and mobile computing services [4–6], because outliers cannot be neglected in big data applications [7].

In its infancy, outlier detection research was largely binary, focused on determining whether an object is an outlier. Recently, significant advancements have redefined outlier detection research, with most detectors attributing an object with a score as a surrogate of the probability of it being an outlier [8]. Many outlier-score-based detectors have been proposed, which will be introduced in Section II. However, few detectors satisfy the industrial standard in large-scale data applications while optimizing time complexity, space complexity, accuracy, interpretability, reproducibility, and scalability. To answer the challenges, we propose a framework for dimensional outlier detection (FDOD), based on which different outlier detectors can be designed. Based on FDOD, we propose two detectors called distance to dimensional median (DDM) and \( k\)-th dimensional-neighbor’s distance (KOBE). DDM and KOBE have the advantages of low time and space complexity, high accuracy, simple interpretability, easy reproducibility, and good scalability.

In this paper, our contributions are as follows: (1) We proposed FDOD, which is a novel framework to design scalable and interpretable detectors to detect dimensional outliers. Two main contributions of FDOD include the iterative projection process controlled by a parameter \( t \), resulting in an average improvement of approximately +5% AUC, and the score modification function, which yields an average increase of around +2% AUC. (2) Based on FDOD, we proposed a detector called DDM for detecting outliers far away from the center, a detector called KOBE, and its variant detectors for detecting outliers in the tails or between clusters. Both detectors are scalable and interpretable and outperform all baselines in accuracy with less computational time and memory usage, hence they are more suitable for large-scale data than existing techniques. (3) The additional contribution of FDOD lies in its potential to be integrated with existing detectors to enhance their performance. For instance, it can enhance a widely used baseline isolation forest (iForest) detector [9] by approximately +6% AUC on average. (4) We conducted extensive experiments with 22 detectors including 10 detectors published in the past three years on 18 real-world datasets.

The subsequent sections of this paper are structured as follows: Section II provides a comprehensive review of the current outlier detection techniques. Section III outlines the novel methods proposed in this study. Section IV presents the results obtained from the conducted experiments. Finally, in Section V, the overall conclusions drawn from the research are presented.

II. LITERATURE REVIEW

1) Statistics-based detectors: Statistics-based methods [10] were the earliest techniques utilized to detect outliers. They could be categorized into single-dimension-based rules for univariate outlier detection or multiple-dimension-based models for multivariate outlier detection. The single-dimension-based rules produced binary labels classifying objects into
outliers and normalities. For example, in the 1860s, Chauvenet proposed a rule to label objects beyond the lower or upper 1/(4n) points of the normal distribution as outliers, where n was a parameter [11]. The 3 sigma rule assumed that objects not located within 3 standard deviations of the mean were outliers [12]. It was noteworthy that these rules were also adopted to thresholding outlier scores to label outliers and normalities [13].

The multiple-dimension-based models produced outlier scores and these models could be classified into two groups, parametric or non-parametric. Parametric models assumed normality data objects followed parametric distribution while outliers did not [14], such as Gaussian mixture models, hidden Markov models, and extreme value theory. Non-parametric models did not have any parametric distribution assumption such as histogram-based methods (HBOS) [15] and kernel density estimation.

Models that were based on robust statistics also played an important role in outlier detection such as the minimum covariance determinant (MCD) [16] and M-estimators [17].

The single-dimension-based rules did not produce outlier scores, while the multiple-dimension-based models and robust-statistic-based methods were typically costly models, limiting their usage in large-scale data applications.

2) Clustering-based detectors: Outlier detection and clustering were two interconnected tasks. When performing clustering, the points with insufficient numbers of objects to form clusters would be labeled as outliers [18]. Alternatively, another concept was to consider the objects far away from their nearest cluster centroids as outliers [19]. Clustering-based methods [20] were optimized for clustering tasks but not for outlier detection tasks, hence their accuracy in detecting outliers depended on the reliability of the clustering methods used.

3) Neighborhood-based detectors: Neighborhood-based detectors or proximity-based detectors [21–23] were popular due to their simplicity. There were mainly two types of neighborhood-based detectors. The first type aimed to detect distance-based outliers, defined as objects with a significant distance from their nearest neighbors [24–26]. An example of such a detector utilized the distance between an object and its kth nearest neighbor as an outlier score (KNN) [25]. The second type worked by detecting density-based outliers, defined as objects having significant density deviation from their neighbors [27–29]. An example of such a detector was the local outlier factor (LOF) which used the density ratio between an object and its k nearest neighbors as an outlier score.

Neighborhood-based detectors relied on identifying the nearest neighbors, which depended on the distance or similarity function. These detectors had high memory usage requirements and faced the curse of dimensionality in high dimension space. These detectors required adjustment of the parameter k to achieve optimal performance as discussed by Yang et al. [30].

4) Learning-based detectors: Learning-based detectors [31, 32] are normally comprised of the training stage and prediction stage. The models learned in the training stage would be used to predict an object’s possibility of being an outlier in the prediction stage. A typical example was the one-class support vector machine, which separated all data points from their origin in the feature space using a hypersphere, and labeled objects lying outside the hypersphere as outliers [33]. Another example was neural-networks-based detectors using generative adversarial networks [34], auto-encoder [35, 36], deep one-class models [37, 38], using contrastive learning [39], and self-supervised models [40, 41]. Neural network-based detectors typically calculate outlier scores by utilizing the loss function that was employed during the training of the neural networks. Training learning-based detectors were time-consuming and neural-networks-based models lacked good interpretability as illustrated by Li et al. [42].

5) Ensemble-based detectors: Ensembles were techniques to combine multiple weak detectors to obtain a more robust detector [2, 21]. Based on this concept, robust detectors can be designed using randomness-based methods such as random tree [9], randomly sampling objects, [43] and randomly sampling features [44]. The base detectors defined with one randomness setting were weak detectors, but the detector obtained by combining different base detectors defined with different randomness settings could be robust. Instead of using a randomness-based strategy, Li et al. [42] proposed a detector called empirical-cumulative-distribution-based outlier detection (ECOD), which aimed to detect outliers in the upper and lower tails of the distribution by aggregating the estimated tail probabilities of each dimension. Ensemble-based detectors [43, 45] depended on the performance of base detectors and were usually less interpretable [42].

6) Subspace-based detectors: Outliers may show different properties in different subspaces, as the role of objects can vary across different subsets of features. Hence, outliers may exhibit normal properties within the full feature space but display significant deviations within a specific subspace. Subspace-based detectors are designed to detect such outliers. Keller et al. introduced a data processing technique for identifying highly contrasting subspaces for all data objects [46] and individual objects [47] to enhance interpretability and accuracy compared to considering all dimensions. Similarly, Muller et al. [48] proposed a method to detect outliers that are not detectable in the full feature space by integrating multiple views into a measure of outlierness. More recently, random-projection-based methods [49, 50] have been utilized to select relevant subspaces for outlier detection.

III. PROPOSED METHOD

This section introduces the problem statement and the proposed FDOD. Based on FDOD, two detectors DDM and KOBE were developed. Related concepts and limitations of DDM and KOBE are also discussed. The FDOD was designed to satisfy several criteria concurrently: time complexity, space complexity, accuracy, interpretability, and scalability.

A. Problem statement

Based on the separability, outliers also could be grouped into dimensional outlier that were separable in at least one
dimension and jointly-dimensional outliers that were separable in at least two joint dimensions. Dimensional outliers fell into two categories based on their locations: outliers in tails and outliers between clusters. In Fig. 1, objects A, C, and D were dimensional outliers, while object B was a jointly-dimensional outlier. Object A and object D were outliers in tails, while object C was an outlier between clusters.

Our proposed FDOD framework focused on detecting dimensional outliers. The proposed method does not aim to detect the jointly-dimensional outliers. The DDM aimed to detect outliers mainly in tails and KOBE aimed to detect outliers both in tails and between clusters. Nevertheless, as illustrated in Fig. 1, it is evident that DDM shows a certain capability in identifying jointly-dimensional outliers, namely the object B.

Fig. 1: The illustration of outlier scores and detected outliers of the ECOD [42] and proposed method DDM (iteration t=1) and KOBE+ (k=2). Objects A, C, and D were dimensionally separable outliers; Object A and D were outliers in the tails, while object C was an outlier between clusters. Object B was a jointly-dimensionally separable outlier.

### B. Framework for dimensional outlier detection (FDOD)

The FDOD contained three components: dimensional score generation, dimensional score modification, and dimensional score aggregation. The projection function \( f(\cdot) \), modification function \( h(\cdot) \), and aggregation function \( g(\cdot) \) were defined in these three components. This article provided examples of defining function \( f(\cdot) \), \( g(\cdot) \) and \( h(\cdot) \).

For a dataset \( X \in \mathbb{R}^{n \times m} \), where \( n \) is the number of rows (dataset size) and \( m \) is the number of columns (dimension size), \( X_{i,:} \) and \( X_{:,j} \) denote the \( i^{th} \) row and \( j^{th} \) column, respectively, while \( X_{i,j} \) denotes the value in the \( i^{th} \) row and \( j^{th} \) column. \( f(\cdot) \) is defined in the dimensional score generation component as Eq. 1.

\[
Y_{i,j} = f(X_{i,j}),
\]

where \( f(\cdot) \) projects each value \( X_{i,j} \) to dimensional score \( Y_{i,j} \) to represent the probability of the object \( X_{i,j} \) being an outlier in \( j^{th} \) dimension. In the next section, two \( f(\cdot) \) functions will be introduced.

After the dimensional outlier score \( Y_{i,j} \) for each \( X_{i,j} \) was produced, there might be local variance or bias in the local region such as \( X_{i,j} \)’s neighborhood, the effect of which should be reduced before combining the dimensional outlier scores in the dimensional score aggregation component. There were two ways to mitigate the local variance or bias for outlier score smoothing. The most deviant scores could be removed; the deviant scores could be averaged [51, 52]. The strategy of summing up scores with \( k \) nearest dimensional neighbors for smoothing scores to improve the consistency between object similarity and score similarity was adopted by the dimensional score modification component. Logically, the closest dimensional neighbors should be considered. However, this resulted in the drawback of additional computational time. Hence, this study opted to define a local region \( \psi_{i,j} \) of \( X_{i,j} \) using Eq. 2.

\[
\begin{align*}
V_{a,j} &= X_{i,j}, \\
L_{i,j,p} &= V_{a-p,j} \in V; p \in \mathbb{N}^+, 1 \leq p < n, \\
R_{i,j,p} &= V_{a+p,j} \in V; p \in \mathbb{N}^+, 1 \leq p < n, \\
\psi_{i,j} &= \{ L_{i,j,p} \}_{p=a-k}^{a-1} \cup \{ V_{a,j} \} \cup \{ R_{i,j,p} \}_{p=a+k}^{a+k+1},
\end{align*}
\]

where \( V_{a,j} \) is the ascendingly sorted \( X_{i,j} \), \( a \) is the ranking of \( X_{i,j} \), \( L_{i,j,p} \) refers to the \( p^{th} \) value smaller than \( X_{i,j} \) in \( j^{th} \) dimension, \( R_{i,j,p} \) refers to the \( p^{th} \) value larger than \( X_{i,j} \) in \( j^{th} \) dimension, and \( k \in \mathbb{N}^+ \) is neighborhood size. \( L \) and \( R \) are called the left and right nearest dimensional neighbors, respectively. The subscripts \( i \) and \( j \) of \( \psi_{i,j} \) are not for indexing but used as a marker to show its correspondence to \( X_{i,j} \). The \( h(\cdot) \) function can be defined as per Eq. 3.

\[
Y_{i,j} = h(Y_{i,j}, k) = \sum_{l} Y_{i,j} Y_{i,j} \in \psi_{i,j},
\]

where \( Y_{i,j} \) is the object in the local region \( \psi_{i,j} \).

Once the dimensional scores \( Y \) were obtained, outlier scores of each \( X_{i,:} \) could be calculated by aggregating the dimensional scores \( Y_{i,j} \) using \( g(\cdot) \) function. Two methods could be used to define \( g(\cdot) \). The first method was to employ existing outlier detectors as \( g(\cdot) \) to model \( Y \) and produce outlier scores for \( X_{i,:} \). Then, the outlier score of \( X_{i,:} \) was used as the outlier score of \( X_{i,:} \). For instance, when employing the existing IForest detector [9] as the function \( g(\cdot) \) within the proposed FDOD, the \( Y \) generated by FDOD will serve as the input for IForest. The fusion of IForest and the input \( Y \) generated by Eq. 5 and Eq. 6 is denoted as DI and KI, correspondingly. The advantage of this approach was that the dependency between dimensions could be considered if detectors such as isolation forest (IForest) [9] or LOF were used. The second method was to combine each value \( Y_{i,j} \) in \( Y_{i,:} \) linearly using the sum, max, averaging or max-sum [2]. For simplicity, we used summation due to reduced time complexity and less need for adjustment of parameters, which summed up each value \( Y_{i,j} \) in \( Y_{i,:} \) as the outlier score of \( X_{i,:} \) as shown with Eq. 4.

\[
o_i = g(Y_{i,:}) = \sum_{j} Y_{i,j}, o \subset \mathbb{R}^n.
\]

The limitation of this approach was that it did not consider the dependency between dimensions. However, the focus in this paper was to detect dimensional outliers which assumed outliers were separable in at least one dimension.

With these three components defined above, we defined frameworks for dimensional outlier detection (FDOD) as shown in Algorithm 1. Unlike IForest, FDOD did not rely
Algorithm 1: \(FDOD(X, t, k) \rightarrow Y, o\)

**Input:** Dataset \(X \in \mathbb{R}^{n \times m}\); Iteration \(t \in \mathbb{N}\); Neighborhood size \(k \in \mathbb{N}^+\)

**Output:** Outlier scores \(o \in \mathbb{R}^n\)

1. \(Y \leftarrow X\);
2. for \(j = 1\) to \(m\) do
   3. while \(t \geq 1\) do
      4. \(Y_{i,j} \leftarrow f(Y_{i,j}); \quad \text{/* Score generation */}\)
      5. \(t \leftarrow t - 1;\)
   6. end
   7. \(Y_{i,j} \leftarrow h(Y_{i,j}, k); \quad \text{/* Score modification (optional) */}\)
end
8. \(o \leftarrow g(Y); \quad \text{/* Score aggregation */}\)

on any randomness, hence the results of FDOD were reproducible. In the projection step, each dimension was projected independently and in parallel, hence FDOD was scalable. Additionally, the projection function \(f(\cdot)\) could also be applied to the dimensional scores \(Y\) multiple times, which is controlled by the parameter \(t\) as shown in Algorithm 1. The projection step was to generate data to facilitate outlier detection, hence this effect was supposed to be enhanced by iteratively applying the projection step. The process of iteratively adjusting dimensional outlier scores can be viewed as a sequential ensemble strategy, which was less explored in the literature. The sequential ensemble follows a principle that successive execution of outlier detectors can provide fresh insights into detectors or portions of the data [2]. Different detectors could be designed by defining or combining the function \(f(\cdot), h(\cdot),\) and \(g(\cdot)\). The first and third components, namely dimensional score generation and dimensional score aggregation, were mandatory and the second component, namely dimensional score modification, was optional. Users can design simple or complex detectors using FDOD according to their demands. Based on FDOD, we introduced two detectors in the following subsections.

**C. Distance to the dimensional median (DDM)**

This subsection introduces a detector called the distance to the dimensional median (DDM) for detecting outliers in tails based on FDOD. DDM contained two components with function \(f(\cdot)\) using Eq. 5 and \(g(\cdot)\) using Eq. 4.

\[
Y_{i,j} = f(X_{i,:}) = |X_{i,j} - m_j|, \quad (5)
\]

where \(m_j\) is the median value of \(X_{.,j}\). Eq. 5 is approximately derived for the purpose of detecting outliers distributed in the tails. The projection step using Eq. 4 can be repeated and controlled by the parameter iteration \(t\). Hence, DDM has one parameter \(t\).

**D. \(k^{th}\) dimensional-neighbor’s distance (KOBES)**

This subsection introduces a detector called the \(k^{th}\) dimensional-neighbor’s distance (KOBES) and its three variants called KOBES, KOBES\(^+\) and KOBES\(^*\) for detecting dimensional outliers. All four models were based on FDOD by defining the \(k^{th}\) dimensional-neighbor \(L_{i,j,k}\) and \(R_{i,j,k}\) using Eq. 2. KOBES contained two components with function \(f(\cdot)\) using Eq. 6 and \(g(\cdot)\) using Eq. 4. KOBES\(^+\) used the same functions as KOBES but also with function \(h(\cdot)\) using Eq. 3.

\[
Y_{i,j} = f(X_{i,:}) = \begin{cases} 
|X_{i,j} - L_{i,j,k}|, & \text{if } a \leq k, \\
|R_{i,j,k} - L_{i,j,k}|, & \text{other}
\end{cases} \quad (6)
\]

\[
Y_{i,j} = f(X_{i,:}, k) = \begin{cases} 
\sum_{p=1}^{k} |X_{i,j} - L_{i,j,p}|, & \text{if } a \leq k, \\
\sum_{p=1}^{k} |R_{i,j,p} - X_{i,j}|, & \text{if } a > n - k,
\end{cases} \quad (7)
\]

where \(|\cdot|\) denote absolute value; \(L_{i,j,k}\) is the \(k^{th}\) value smaller than \(X_{i,j}\) and \(R_{i,j,k}\) is the \(k^{th}\) value bigger than \(X_{i,j}\). Both Eq. 6 and Eq. 7 are approximately designed with the aim to identify outliers distributed in the tails and between clusters. To reduce the number of parameters, KOBES-based variants set iteration \(t = 1\) by default.

![Fig. 2: An example of how KOBES computed the outlier score \(o_{ij}\) of the object \(X\): a) the original feature space; b) calculated the dimensional score \(Y_{i,2}\) for the dimension \(X\) using Eq. 6 with setting \(k = 2\); c) calculated the dimensional score \(X_{i,j}\) for the dimension \(Y\) using Eq. 6 with setting \(k = 2\); d) calculated the outlier score of \(o_{ij}\) by summing up those two dimensional scores. The numbers 3 and 4 mentioned were simply used as examples and could be replaced with other values.](image-url)

Fig. 2 illustrated an example of how KOBES computed the outlier score \(o_{ij}\) of the object \(X_{ij}\) as shown in Fig. 2a in 2-dimensional space. KOBES first calculated the dimensional score \(Y_{i,j} = X_{ij}\) for the dimension \(X\) as shown in Fig. 2b, and \(Y_{i,j} = Y_{ij}\) for the dimension \(Y\) as shown in Fig. 2c using Eq.
6 with setting $k = 2$. Then, KOBE summed up these two dimensional scores as the outlier score of $a_i$ of the object $X_i$; as shown in Fig. 2d.

E. Discussion

Contribution in FDOD: FDOD provided two main contributions. First, the projection function $f(\cdot)$ can be repeated to enhance the features of outliers. The usage of repetition had not been described in the existing literature. Second, FDOD introduced the $b(\cdot)$ to modify the dimension scores to reduce the local variance.

Related concepts: The concept of extrapolating the outlier score from the distance to the $k^{th}$ neighbor was adapted from the KNN detector [25]. KOBE-based variants had two key differences from KNN. First, KNN utilized the $k^{th}$ neighbor defined in full dimension space, while KOBE used the $k^{th}$ dimensional neighbor defined in a single dimension. Second, KOBE$^+$ combined the scores with dimensional neighbors, while KNN did not have this function. The concept of processing each dimension separately was also used in ECOD [42] and dimensionally distributed density estimation (DDDE) [53]. The details of ECOD had been introduced in the literature review. The DDDE evaluated the dimensional density for clustering, while KOBE computed dimensional scores for outlier detection. The concept of combining scores with neighbors was used in neighborhood averaging (NA) [51, 52]. NA defined neighbors in full-dimension space, while KOBE defined dimensional neighbors in a single dimension.

Complexity: The time and space complexity of FDOD depended on the functions $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$. DDM had $O(tm n)$ for both time and space complexity; KOBE, KOBES, KOBE$^+$, and KOBES$^+$ had $O(m n \log(n))$, $O(m(n \log(n) + kn))$, $O(m(n \log(n) + kn))$, and $O(m(n \log(n) + kn))$ time complexity, respectively, where $n$ is the dataset size, $m$ is the dimension size, $t$ is the iteration of applying $f(\cdot)$, and $k$ is the neighborhood size. All KOBE-based methods required $O(mn)$ space complexity. When FDOD is used in conjunction with existing detectors, the overall complexity is contingent upon both the complexity of FDOD and the employed detector. For instance, DI and KI exhibit time complexities of $\max(O(tm n), O(TN \log(N)))$ and $\max(O(m n \log(n)), O(TN \log(N)))$, respectively. The term $O(TN \log(N))$ represents the complexity of IForest, where $T$ denotes the number of trees and $N$ signifies the number of objects sampled from datasets for the training of each tree.

Limitation: DDM was designed to detect outliers in tails, which required outliers to be located in the tails of at least one dimension. KOBE was designed to detect dimensional outliers, which required outliers that were separable in at least one dimension. To detect outliers defined with joint dimensions, other detectors or FDOD jointly used with other detectors as $g(\cdot)$ were recommended (see section III-B). However, DDM and KOBE were designed for industrial usage and proved to be suitable for most real-world situations considering time complexity, space complexity, accuracy, interpretability, and scalability. This was verified by the experiments in section IV. When setting iteration $t = 1$ for FDOD, DDM had no any parameter to tune and KOBE relied on a parameter neighborhood size $k$. The main limitation was the lack of an established method to determine the optimum value of $k$. Some related work could be found in [30].

IV. EXPERIMENTAL RESULTS

Since the FDOD as a framework of producing detectors cannot be evaluated directly via experiments, we evaluate the detectors generated using the FDOD. In this section, firstly, we show the performance difference between the detectors generated using the FDOD and introduce selection rules for these detectors. Secondly, we show the advance in accuracy, complexity, and interpretability analysis by comparing the proposed methods with the baselines or competitive baselines. After that, we show the potential of jointly using existing detectors with the proposed FDOD. Finally, we discuss the limitation and future work.

A. Experimental settings

<table>
<thead>
<tr>
<th>TABLE I: Dataset information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>Cover</td>
</tr>
<tr>
<td>Saitame</td>
</tr>
<tr>
<td>Shuttle</td>
</tr>
<tr>
<td>WBC</td>
</tr>
<tr>
<td>Thyroid</td>
</tr>
<tr>
<td>WDBC</td>
</tr>
<tr>
<td>Musk</td>
</tr>
<tr>
<td>Lymphography</td>
</tr>
<tr>
<td>Glass</td>
</tr>
<tr>
<td>Wine</td>
</tr>
<tr>
<td>Stamps</td>
</tr>
<tr>
<td>Pageblocks</td>
</tr>
<tr>
<td>InternetAds</td>
</tr>
<tr>
<td>Cardiogicography</td>
</tr>
<tr>
<td>WPBC</td>
</tr>
<tr>
<td>Breastw</td>
</tr>
<tr>
<td>Arrhythmia</td>
</tr>
<tr>
<td>Parkinson</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II: Baseline information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
</tr>
<tr>
<td>DSV [37]</td>
</tr>
<tr>
<td>RCA [36]</td>
</tr>
<tr>
<td>NTL [54]</td>
</tr>
<tr>
<td>ICL [39]</td>
</tr>
<tr>
<td>SLAD [40]</td>
</tr>
<tr>
<td>DIF [45]</td>
</tr>
<tr>
<td>GAAL [34]</td>
</tr>
<tr>
<td>VAE [35]</td>
</tr>
<tr>
<td>LUN [32]</td>
</tr>
<tr>
<td>NSS [22]</td>
</tr>
<tr>
<td>NRE [21]</td>
</tr>
<tr>
<td>MOD [55]</td>
</tr>
<tr>
<td>LOF [27]</td>
</tr>
<tr>
<td>ODN [28]</td>
</tr>
<tr>
<td>SVM [33]</td>
</tr>
<tr>
<td>PCA [56]</td>
</tr>
<tr>
<td>ECOD [42]</td>
</tr>
</tbody>
</table>

The acronyms NDL and DL denote non-deep learning and deep learning, correspondingly.

The experiment was performed on 16 public real-world datasets. The information was summarized in Table I [57, 58].
The identification of diseases such as Cardiotocography and Parkinson's disease are crucial healthcare applications that are closely linked to mobile computing scenarios. 22 outlier detectors were used as baselines, including the method called Deep Support Vector data description (DSV) [37], the method using Robust Collaborative Autoencoders (RCA) [36], the method using Neural Transformation Learning (NTL) [54], the method using Internal Contrastive Learning (ICL) [39], the Scale Learning-based deep Anomaly Detection method (SLAD) [40], the method called Deep Isolation Forest (DIF) [45], Mean-shift Outlier Detection plus (MOD+) [55], LOF [27], Outlier Detection using Indegree of Nodes (ODIN) [28], the reverse unreachability (NC, as defined in [26]), KNN [24], Neighborhood Representative plus (NR+) [21], the Group Similarity System based on unsupervised outlier detection (GSS) [22], a graph-neural-networks-based method (LUNAR, as defined in [32]), a generative-adversarial-active-learning-based method (SO-GAAL, as defined in [34]), a variation-autoencoder-based (VAE) method [35], MCD [25], an isolation-forest-based (IForest) method [9], a support-vector-machines-based (SVM) method [33], a principal-component-analysis-based (PCA) method [56], HBOS [15], and ECOD [42]. More information about these baselines are summarized in Table I. Of these 22 outlier detectors, 9 detectors were based on deep learning and 10 detectors were published within 3 recent years. LOF was used as the base detector for NR+ and GSS. To evaluate the efficacy of each method, the area under the receiver operating characteristic (ROC AUC) was used.

B. Illustration

Two cases in Fig. 1 and Fig. 3 were used to depict the difference between the proposed detectors and the ECOD. As illustrated in Fig. 1, both ECOD and DDM attempted to detect outliers in tails, represented by objects A and D, but their efforts to identify the outlier in closer proximity to the center, D, were unsuccessful. This indicated that the skew of data distribution affected the performance. In contrast, KOBE+ managed to detect all outliers in tails and between clusters as expected. However, DDM was the only detector able to detect object B, proving that DDM had some ability to detect jointly-dimensional outliers.

A comparison between ECOD and proposed models was illustrated in Fig. 3. The results again verified that the proposed models with AUCs of more than 0.931 were superior to ECOD with an AUC of 0.909. The best detector was KOBE+ with an AUC of 1.00. The outlier score findings in the left column indicate that KOBE+ and KOBES+ exhibit more consistency between object similarity and score similarity compared to KOBE and KOBES, respectively. This observation suggests that the score modification function \( h(\cdot) \) has the capability to smooth the scores.

C. Comparison between DDM and KOBE-based variants

The AUC results for the proposed methods under different parameter settings are shown in Table III. The \( \text{var} \) means the best AUC was reported for each dataset after tuning the parameter iteration varying \( t \) from 0 to 5 and the parameter iteration \( k \) varying from 3 to 100. The \( 7n \) means to set \( k = [7 \cdot \log(n)] \) following the settings by Yang et al. [30], where \( n \) is the dataset size. The rows labeled as \( \text{AVG} \) were the average AUC values per column. Due to the limitation of the table cell, KS, KS+, and KS+ were used to denote KOBES, KOBE+, and KOBES+. Several observations can be made.

Firstly, in the \( \text{AVG} \) row of the first 8 columns, the performance of KOBE+ and KOBES+ surpassed that of KOBE and KOBES with AUC improvement from +1% to +2%, demonstrating the efficacy of the score modification function \( h(\cdot) \). KOBE exhibited superior performance to DDM, as KOBE could identify outliers distributed in tails and between clusters, whereas DDM could only detect outliers in tails. KI was better than KOBE, which was reasonable since IForest had considered the effect of joint dimensions compared to KOBE. Surprisingly, DI did not enhance DDM and even exhibited a performance decrease of over -10% AUC, particularly on the wine and Parkinson datasets. However, DI did show an improvement in performance over DDM by more than +4% AUC in other datasets such as WDBC, PageBlock, and InternetAds. The performance disparity between DI and DDM was contingent on the number of jointly-dimensional outliers in the dataset.

Secondly, from the 9th to 13th columns, all KOBE-based methods demonstrated comparable performance around 87% AUC, with KOBE emerging as the top performer with 87.10% AUC. KI surpassed all KOBE-based methods, underscoring the significance of the parameter \( t \) for KOBE-based techniques.

Thirdly, in the final 7 columns, DI exhibited superior performance to DDM, indicating that the parameter \( t \) had a more pronounced impact on DDM than on DI. Among the KOBE-based methods, KOBE performed the best, while KI displayed the weakest performance, highlighting their differing sensitivities to the parameters \( k \). KOBE also emerged as the top performer overall with 83.76% AUC, showcasing its relatively lower sensitivity to parameters.

Fourthly, the performance of the proposed methods varied across datasets, suggesting the need for tailored selection based on specific applications. Given KOBE’s ability to detect outliers in tails and between clusters, as well as its superior performance under fixed parameters, KOBE was recommended for utilization.

Fifthly, upon parameter \( t \) tuning, DDM’s AUC was improved from 82.59% to 87.20%, DI’s AUC from 83.70% to 85.01%, and KOBE-based methods saw enhancements of +1% to +3% AUC, elevating them from around 87% to around 89% AUC. Adjusting parameter \( k \) led to a substantial improvement in KI’s AUC by approximately +12%, from 76.58% to 88.46% AUC, and other KOBE-based methods saw enhancements of around +4% AUC, elevating them from around 83% to 87% AUC. This verified the efficacy of the iterative projection process controlled by the parameter \( t \).

In conclusion, KOBE, with an average AUC of 83.76%, performed the best with default parameter settings. Further enhancements of around +4% AUC could be achieved by tuning parameter \( k \), around +5% AUC by tuning both parameters \( k \) and \( t \).
The parameter k ranged from 3 to 100 and the best AUC results were reported for k = 15. The parameter k impacts both the number of detectors per column and the standard deviation of AUC values per column. In the absence of feature selection, the proposed methods exhibit superior performance compared to the baseline methods. Notably, the most effective baseline, DIF, achieved 89.56% AUC by the proposed KOBE method. Additionally, the KOBE method with default parameters exhibits superior performance compared to the baseline methods. Furthermore, the KOBE method with default parameters could improve DIF by +0.6% to reach 83.76% AUC, showcasing advancements over the baseline. In comparison to the competitive baseline ECOD, DDM and KOBE demonstrate enhancements in AUC of approximately +4% and +5%, respectively.
the baseline methods. Furthermore, in the Best standard deviation, indicating greater stability compared to STD proposed FDOD approach with existing detectors. Analysis of baseline IForest from 83.39% to 85.01% and 89.04% AUC, Fig. 1, while ECOD cannot. Both DI and KI enhanced the identify outliers between clusters, and DDM can identify some the decades.

highlighted the inherent challenges of outlier detection over greater efficacy compared to those designed for detecting any or unspecified types of outliers. This can be supported by both the proposed detectors and the competitive baseline detectors KNN designed for distance-based outliers and ECOD defined for outliers in the tails.

In summary, the detectors generated through the FDOD consistently exhibit superior performance compared to the baseline detectors, underscoring the efficacy of the FDOD. In the selection of these detectors, it is crucial to weigh the balance between detection performance and computational consumption. Based on the results obtained under default settings, KOBE emerged as the top performer and is therefore suggested as the default choice. Nevertheless, if a swifter alternative to KOBE is required, DDM is advised, as its performance under default parameters remained competitive with the baseline detectors.

### E. Score aggregation methods

As outlined in Section III-B, the score aggregation function $g(\cdot)$ offers various alternatives. Our study evaluated the effectiveness of different score aggregation techniques, including minimum (min), maximum (max), median, and sum (sum) of the dimensional outlier scores. The optimal outcomes are detailed in Table V following the adjustment of parameter iteration between 0 to 5 and parameter iteration $k$ between 3 and 100.

According to the findings presented in the AVG row, the summation approach demonstrated superior performance compared to other score aggregation techniques across various detectors. This could be attributed to the impact of each dimensional score on the overall degree of being an outlier, suggesting that all dimensional scores should be taken into account rather than only a subset of them. However, it is noted that the most effective score aggregation methods varied based on the specific application and detector utilized, considering the computational complexity, it is advisable to adopt summation as the default choice.

### F. Parameter analysis

The impact of the parameters $t$ and $k$ for KOBE and the parameter $t$ for DDM on various datasets was demonstrated in Fig. 4 and Fig. 5. Here, $t = 0$ represents the outcomes of the respectively. This is justified as KOBE has the capability to identify outliers between clusters, and DDM can identify some jointly-dimensional outliers, such as object B illustrated in Fig. 1, while ECOD cannot. Both DI and KI enhanced the baseline IForest from 83.39% to 85.01% and 89.04% AUC, respectively. This underscores the efficacy of integrating the proposed FDOD approach with existing detectors. Analysis of the $STD$ row revealed that the proposed KI exhibited the lowest standard deviation, indicating greater stability compared to the baseline methods. Furthermore, in the Best row, KOBE+ emerged as the top-performing outlier-detector. It is important to acknowledge that each detector has its own limitations, and their performance should be evaluated within the specific context. This was consistent with the conclusions drawn by Li et al. [42], Aggarwal et al. [2], and Yang et al. [52] that no single detector has demonstrated consistent superiority over other detectors across various applications. The observations highlighted the inherent challenges of outlier detection over the decades.

In datasets of larger size, such as Cover, InternetAds, and Arrhythmia, the proposed KI achieved a higher AUC of 96.22% compared to the top-performing baseline DIF with an AUC of 95.68% on the Cover dataset, as illustrated in Table IV. Similarly, the proposed KOBE, with a larger $k$ value, demonstrated an AUC of 97.98% as depicted in Fig 5, surpassing the best baseline KNN with an AUC of 78.11% on the InternetAds dataset, as shown in Table IV. While the proposed approach did not outperform the leading ECOD model with a 75.76% AUC on the Arrhythmia dataset, the proposed DI model achieved a competitive AUC of 75.57% relative to ECOD. Additionally, the proposed DDM model, with an AUC of 74.67%, exhibited slightly inferior performance to ECOD; however, DDM necessitates significantly fewer computational resources than ECOD, as detailed in Section III-G.

However, even a simple method like DDM with tuned parameters can outperform baselines. This poses the question of whether technically complex solutions are necessary for outlier detection in big data applications in the outlier detection domain as complex solutions are typically computationally costly. This also raises the question as to whether detectors designed to identify a particular type of outliers might exhibit greater efficacy compared to those designed for detecting any
The first category showed that the results of $t > 1$ were consistently better than those with $t = 1$, as seen in Shuttle, Breastw, and Arrhythmia datasets. The second category demonstrated that the results of $t > 1$ were consistently better than those with $t = 1$, as observed in Pageblocks and Parkinson datasets.

The third category indicated that the results of $t > 1$ were better than those with $t = 1$ within a specific range of parameter $k$, such as Cover, Glass, and InternetAds datasets. Some datasets like Satimage, Thyroid, and Musk were less sensitive to the parameter $k$, while datasets such as Shuttle, Glass, and InternetAds were more sensitive to the parameter $k$. It is noteworthy that the results for InternetAds using the proposed methods in Tables III, IV, and V were obtained with appropriate values of $k$ within a specific range of $t > 1$ and $t > 0$. The results fell into three categories. Based on the average results with default settings, KOBE achieved 97.98% AUC instead of 64.28% as shown in Table III and outperformed the baselines as shown in Table IV. Therefore, it is recommended to use $k = \lceil 7 \times \log(n) \rceil$ and $t = 1$ as the default settings.

In conclusion, optimal results can be achieved with appropriate values of $t$ and $k$, which are dependent on the application. Based on the average results with default settings of $k = \lceil 7 \times \log(n) \rceil$ and $t = 1$ in Table III, KOBE still outperformed the baselines as shown in Table IV. Therefore, it is recommended to use $k = \lceil 7 \times \log(n) \rceil$ and $t = 1$ as the default settings.

original datasets. Analysis of Fig. 4 revealed that when comparing results between $t = 1$ and $t > 1$, there was an average improvement of +1.11% in AUC, with varying improvements per dataset. The maximum enhancement was observed in the Parkinson dataset with a +8.84% increase in AUC. Notably, the performance of Breastw and Arrhythmia datasets slightly decreased by less than -1% AUC. Surprisingly, the optimal performance for Wine and Parkinson datasets was achieved when $t = 0$. This indicates that achieving optimal performance requires parameter tuning, and once optimal performance is attained, larger or smaller values of $t$ may degrade performance.

In Fig. 5, except for WBC and Breastw datasets, results with $t > 0$ were superior to those with $t = 0$. When comparing between $t = 1$ and $t > 1$, the results fell into three categories. The first category showed that the results of $t = 1$ were consistently better than those with $t > 1$, as seen in Shuttle, Breastw, and Arrhythmia datasets. The second category demonstrated that the results of $t > 1$ were consistently better than those with $t = 1$, as observed in Pageblocks and Parkinson datasets.

The third category indicated that the results of $t > 1$ were better than those with $t = 1$ within a specific range of parameter $k$, such as Cover, Glass, and InternetAds datasets. Some datasets like Satimage, Thyroid, and Musk were less sensitive to the parameter $k$, while datasets such as Shuttle, Glass, and InternetAds were more sensitive to the parameter $k$. It is noteworthy that the results for InternetAds using the proposed methods in Tables III, IV, and V were obtained with appropriate values of $k$ within a specific range of $t > 1$ and $t > 0$. The results fell into three categories. Based on the average results with default settings, KOBE achieved 97.98% AUC instead of 64.28% as shown in Table III and outperformed the baselines as shown in Table IV. Therefore, it is recommended to use $k = \lceil 7 \times \log(n) \rceil$ and $t = 1$ as the default settings.
G. Complexity analysis

We selected competitive baselines to compare the space and time complexity. ECOD was the best in accuracy and the most effective in memory usage while HBOS was the fastest [42]. Since the LOF, IForest and KNN were frequently used detectors in the outlier detection domain, a comparison was made to these detectors. The running time and memory usage peak were shown in Fig 6 tested with 2-dimensional datasets ranging the dataset size from $10^2$ to $10^7$. HBOS, DDM and KOBE were superior regardless of running time or memory usage peak.

To study the effect of dimensions, the running time and the dataset size from $10^2$ to $10^6$ and dimension $d$ from $10^1$ to $10^4$ were summarized in Table VI. DDM and KOBE were more effective than ECOD and HBOS in terms of memory usage. In terms of the effectiveness of running time, DDM was the fastest, followed by HBOS, KOBE and lastly ECOD. KOBE required only 42% running time of ECOD while DDM required only 0.06% running time of HBOS when $d=100$ and $n=1,000,000$.

H. Interpretation analysis

![Fig. 6: Running time (Left) and memory usage peak (Right) for competitive detectors. KOBE needed only 13% memory usage peak of ECOD.](image)

The dimensional scores computed by $f(·)$ of FDOD were helpful to find the dimensions that the outliers significantly deviated. For example, two outliers were randomly selected from the Musk dataset: row no. 177 and 300. Fig. 7 illustrated the normalized dimensional outlier scores computed by DDM and KOBE on these two rows. The scores exceeding the 99th percentile of each dimension were circled and the corresponding dimensions contributed to the outlierness. The 99th percentile was a reference, and not limited to it, the median absolute deviation (MAD) [13] could be used as a reference. As illustrated in Fig. 7, the circled dimensions lead to the judgment of being outliers.

I. Joint Usage with other detectors

FDOD supported the usage of existing detectors as the function $g(·)$ to compute outlier scores based on the projected dataset $Y$. ECOD and IForest detectors were tested. Their results based on the projected dataset $Y$ by Eq. 6 and Eq. 5 were shown in Fig. 8. $K = 0$ or $t = 0$ denoted the original results of ECOD and IForest. We can see both ECOD and IForest could be significantly improved with +0.04 ~ +0.08 AUC when jointly used with FDOD. This proved that FDOD was robust and complementary to the existing detectors.

![Fig. 8: Applying existing outlier detector ECOD and IForest as the function $g(·)$ to the projected dataset $Y$ by Eq. 6 (Left) and Eq. 5 (Right) to calculate outlier scores.](image)

J. Limitation and future work

**Limitation of this work:** The evaluation of the proposed methods is hindered by the limited size and dimensionality of the benchmark datasets. Therefore, it is necessary to obtain larger outlier detection datasets from industry in order to adequately assess and advance the field of outlier detection. Thus, we advocate for the creation of additional larger outlier detection datasets from the industry to facilitate the advancement of outlier detection methodologies.

**Future work:** The evaluation of the effectiveness of the suggested approaches in handling extensive datasets and their applicability in real-world scenarios, including mobile environments, will be a subject of future investigation.

V. Conclusion

We proposed a framework called FDOD for dimensional outlier detection, which had good interpretability and scalability. Based on this framework, we proposed DDM and KOBE detectors, which outperformed state-of-the-art detectors as determined by ROC AUC, yet requiring less computational time and memory usage. In summary, the proposed methods were more suitable for large-scale data applications in the industry than the existing techniques. Moreover, this study also poses a question to the outlier detection domain that *whether technically complex solutions are really needed for outlier detection in big data applications.* At last, we advocate for opening larger outlier detection datasets from the industry to support the development of outlier detection in big data.
REFERENCES


[51] S. R. Yang, Jiawei and P. Fränti, “Smoothing outlier scores is all you need to improve outlier detectors,” IEEE Transactions on Knowledge and Data Engineering, 2023.


